BRL Seminar

(2023. 03. 13. Mon)

Self-Supervised Learning with Time Series Data (3)

통합과정 6학기 이승한

Papers

1. Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency (NeurIPS 2022)
2. CoST : Contrastive Learning of Disentangled Seasonal-Trend Representations for TS forecasting (ICLR 2022)

Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency

Xiang Zhang[†] [*]
Harvard University
xiang_zhang@hms.harvard.edu

Ziyuan Zhao[*]
Harvard University
ziyuanzhao@college.harvard.edu

Theodoros Tsiligkaridis
MIT Lincoln Laboratory
ttsili@ll.mit.edu

Marinka Zitnik
Harvard University
marinka@hms.harvard.edu

Abstract

Pre-training on time series poses a unique challenge due to the potential mismatch between pre-training and target domains, such as shifts in temporal dynamics, fast-evolving trends, and long-range and short-cyclic effects, which can lead to poor downstream performance. While domain adaptation methods can mitigate these shifts, most methods need examples directly from the target domain, making them suboptimal for pre-training. To address this challenge, methods need to accommodate target domains with different temporal dynamics and be capable of doing so without seeing any target examples during pre-training. Relative to other modalities, in time series, we expect that time-based and frequency-based representations of the same example are located close together in the time-frequency space. To this end, we posit that time-frequency consistency (TF-C) — embedding a time-based neighborhood of an example close to its frequency-based neighborhood — is desirable for pre-training. Motivated by TF-C, we define a decomposable pre-training model, where the self-supervised signal is provided by

https://arxiv.org/pdf/2206.08496.pdf

Published as a conference paper at ICLR 2022

CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting

Gerald Woo[1 2], **Chenghao Liu**[1] [*], **Doyen Sahoo**[1], **Akshat Kumar**[2] **& Steven Hoi**[1]
[1]Salesforce Research Asia, [2]Singapore Management University
{gwoo,chenghao.liu,dsahoo,shoi}@salesforce.com, akshatkumar@smu.edu.sg

Abstract

Deep learning has been actively studied for time series forecasting, and the mainstream paradigm is based on the end-to-end training of neural network architectures, ranging from classical LSTM/RNNs to more recent TCNs and Transformers. Motivated by the recent success of representation learning in computer vision and natural language processing, we argue that a more promising paradigm for time series forecasting, is to first learn disentangled feature representations, followed by a simple regression fine-tuning step – we justify such a paradigm from a causal perspective. Following this principle, we propose a new time series representation learning framework for long sequence time series forecasting named CoST, which applies contrastive learning methods to learn disentangled seasonal-trend representations. CoST comprises both time domain and frequency domain contrastive losses to learn discriminative trend and seasonal representations, respectively. Extensive experiments on real-world datasets show that CoST consistently outperforms the state-of-the-art methods by a considerable margin, achieving a 21.3% improvement in MSE on multivariate benchmarks. It is also robust to various choices of backbone encoders, as well as downstream regressors. Code is available at https://github.com/salesforce/CoST.

https://arxiv.org/pdf/2202.01575.pdf

Papers

1. Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency (NeurIPS 2022)
2. CoST : Contrastive Learning of Disentangled Seasonal-Trend Representations for TS forecasting (ICLR 2022)

Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency

Xiang Zhang[†] * Harvard University xiang_zhang@hms.harvard.edu

Ziyuan Zhao* Harvard University ziyuanzhao@college.harvard.edu

Theodoros Tsiligkaridis MIT Lincoln Laboratory ttsili@ll.mit.edu

Marinka Zitnik Harvard University marinka@hms.harvard.edu

Abstract

Pre-training on time series poses a unique challenge due to the potential mismatch between pre-training and target domains, such as shifts in temporal dynamics, fast-evolving trends, and long-range and short-cyclic effects, which can lead to poor downstream performance. While domain adaptation methods can mitigate these shifts, most methods need examples directly from the target domain, making them suboptimal for pre-training. To address this challenge, methods need to accommodate target domains with different temporal dynamics and be capable of doing so without seeing any target examples during pre-training. Relative to other modalities, in time series, we expect that time-based and frequency-based representations of the same example are located close together in the time-frequency space. To this end, we posit that time-frequency consistency (TF-C) — embedding a time-based neighborhood of an example close to its frequency-based neighborhood — is desirable for pre-training. Motivated by TF-C, we define a decomposable pre-training model, where the self-supervised signal is provided by

https://arxiv.org/pdf/2206.08496.pdf

Published as a conference paper at ICLR 2022

CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting

Gerald Woo[1 2], **Chenghao Liu**[1] *, **Doyen Sahoo**[1], **Akshat Kumar**[2] & **Steven Hoi**[1]
[1]Salesforce Research Asia, [2]Singapore Management University
{gwoo,chenghao.liu,dsahoo,shoi}@salesforce.com, akshatkumar@smu.edu.sg

Abstract

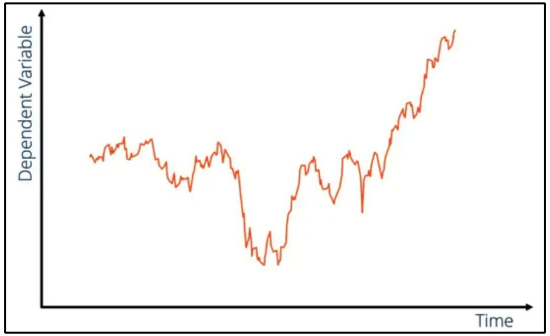
Deep learning has been actively studied for time series forecasting, and the mainstream paradigm is based on the end-to-end training of neural network architectures, ranging from classical LSTM/RNNs to more recent TCNs and Transformers. Motivated by the recent success of representation learning in computer vision and natural language processing, we argue that a more promising paradigm for time series forecasting, is to first learn disentangled feature representations, followed by a simple regression fine-tuning step – we justify such a paradigm from a causal perspective. Following this principle, we propose a new time series representation learning framework for long sequence time series forecasting named CoST, which applies contrastive learning methods to learn disentangled seasonal-trend representations. CoST comprises both time domain and frequency domain contrastive losses to learn discriminative trend and seasonal representations, respectively. Extensive experiments on real-world datasets show that CoST consistently outperforms the state-of-the-art methods by a considerable margin, achieving a 21.3% improvement in MSE on multivariate benchmarks. It is also robust to various choices of backbone encoders, as well as downstream regressors. Code is available at https://github.com/salesforce/CoST.

https://arxiv.org/pdf/2202.01575.pdf

Contents

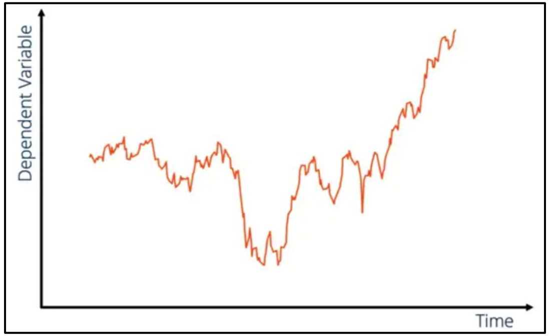
1. Time Series Data in TIME & FREQUENCY Domain

1. **TIME Domain Analysis**:
- provides an **intuitive understanding** of the data's characteristics
    - ex) when **changes** occurred & **magnitude** of values
- can identify trends, cycles, and seasonality

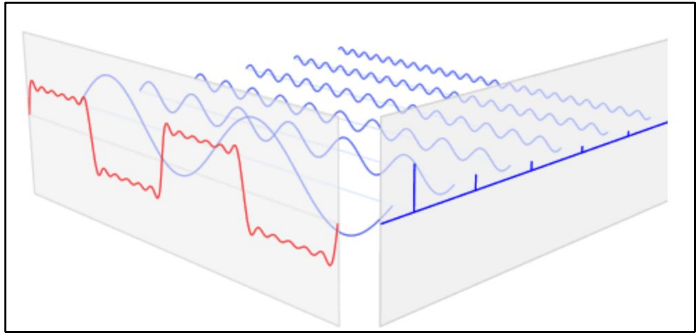


1. Time Series Data in TIME & FREQUENCY Domain

1. **TIME Domain Analysis**:
- provides an **intuitive understanding** of the data's characteristics
  - ex) when **changes** occurred & **magnitude** of values
- can identify trends, cycles, and seasonality



2. **FREQUENCY Domain Analysis**:
- useful for identifying the **periodicity of data**
- can be used to analyze the **frequency distribution** of data
- +) can filter noise from data, resulting in refined data

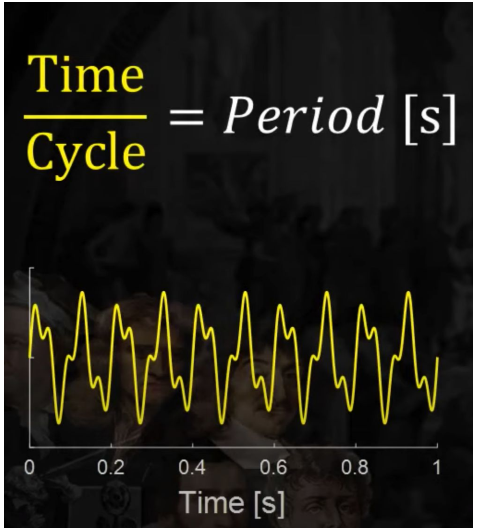


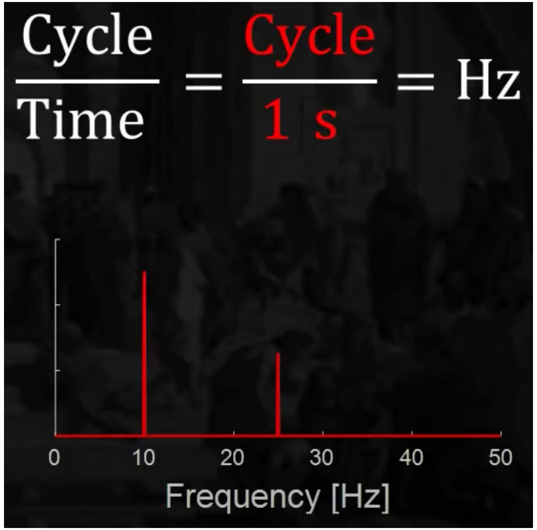
➡ Better to use BOTH!

1. Time Series Data in TIME & FREQUENCY Domain

Different Perspective of TS data

- (1) Time Domain
- (2) Frequency Domain

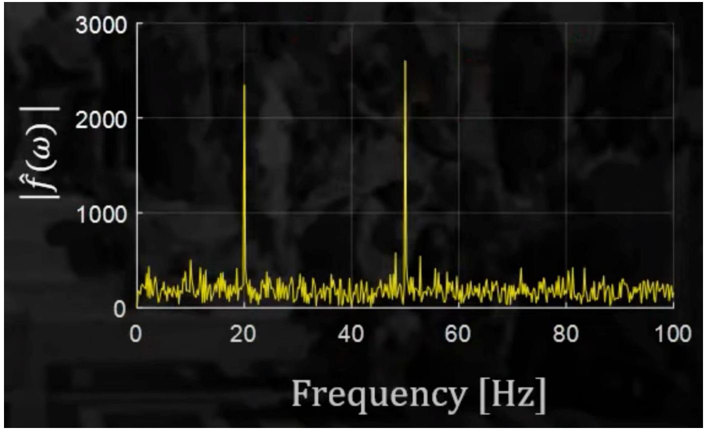




1. Time Series Data in TIME & FREQUENCY Domain

Different Perspective of TS data

- (1) Time Domain
- (2) Frequency Domain

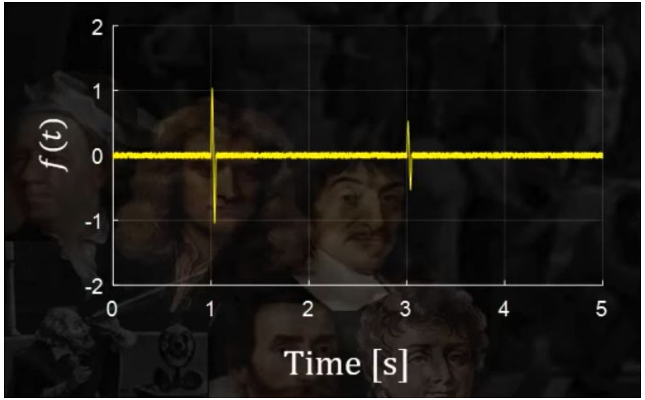


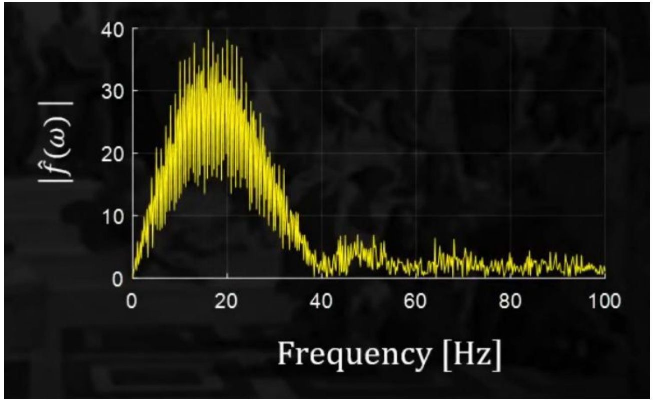
analyzing in FREQUENCY domain : useful in **periodical functions**

1. Time Series Data in TIME & FREQUENCY Domain

Different Perspective of TS data

- (1) Time Domain
- (2) Frequency Domain



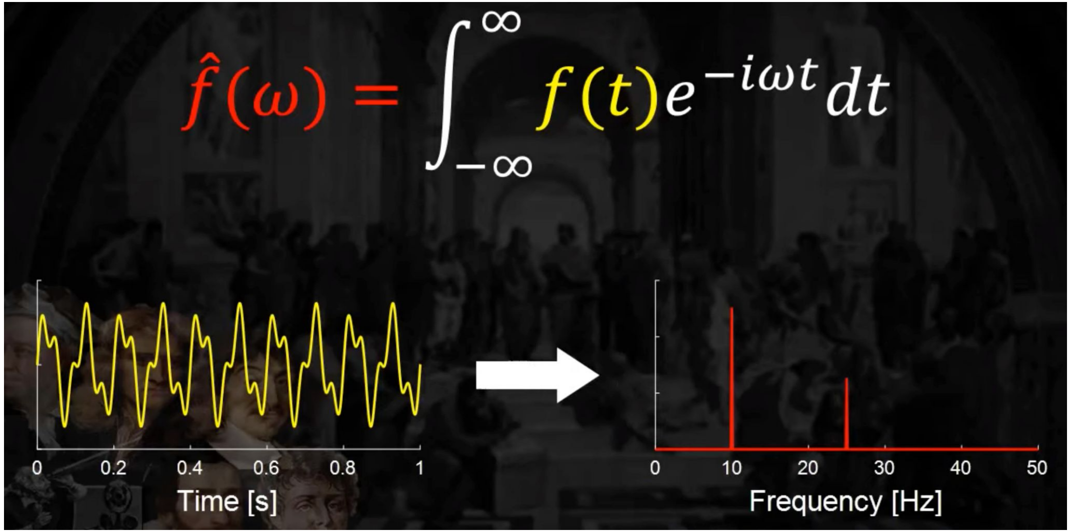


analyzing in FREQUENCY domain : useful in **periodical functions**

https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Fourier Transform : TIME domain -> FREQUENCY domain



https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- b) Euler's formula
- c) Fourier Transform

https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- **a) Fourier Series**
- b) Euler's formula
- c) Fourier Transform

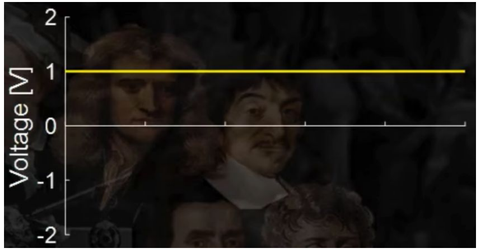
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

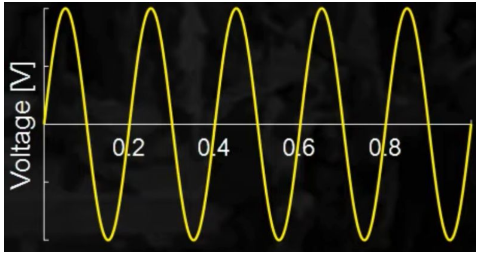
Preliminaries

- **a) Fourier Series**
- b) Euler’s formula
- c) Fourier Transform

$$\hat{f}(x) = \underbrace{\frac{a_0}{2}}_{\text{DC}} + \underbrace{\sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi n x}{T}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi n x}{T}\right)}_{\text{AC}}$$



DC



AC

https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- **a) Fourier Series**
- b) Euler's formula
- c) Fourier Transform

Example)

How can we express this function, using Fourier Series?

https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

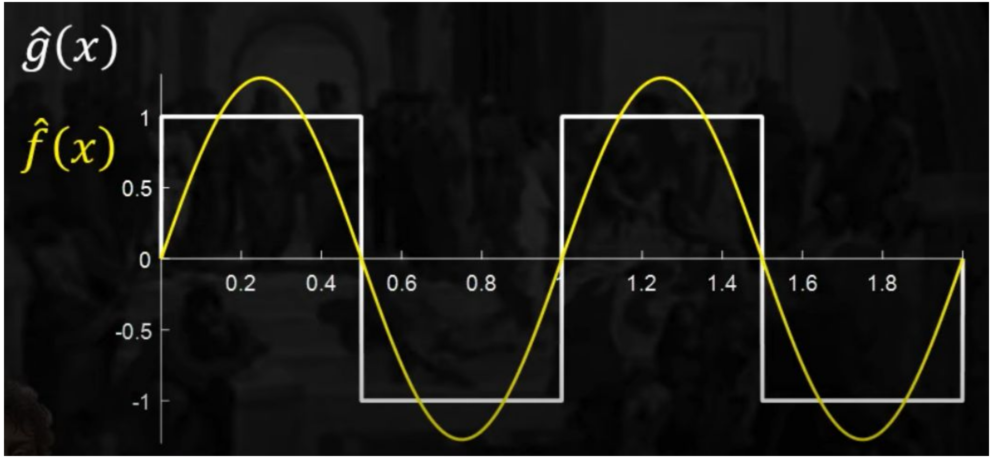
Preliminaries

- **a) Fourier Series**
- b) Euler's formula
- c) Fourier Transform

using **1** sine function

$$\hat{f}(x) = \sum_{n=1,3,5..}^{N=1} \frac{4}{\pi n} \sin\left(\frac{2\pi n x}{1}\right) = \frac{4}{\pi 1} \sin\left(\frac{2\pi 1 x}{1}\right)$$

Example)



https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

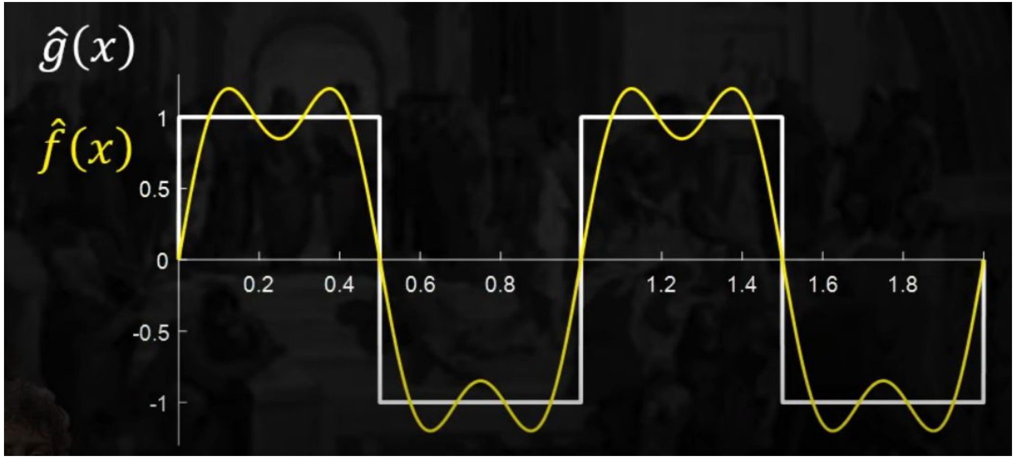
Preliminaries

- **a) Fourier Series**
- b) Euler's formula
- c) Fourier Transform

using **5** sine function

$$\hat{f}(x) = \sum_{n=1,3,5..}^{N=5} \frac{4}{\pi n}\sin\left(\frac{2\pi n x}{1}\right) = \frac{4}{\pi 1}\sin\left(\frac{2\pi 1 x}{1}\right) + \frac{4}{\pi 3}\sin\left(\frac{2\pi 3 x}{1}\right)$$

Example)



https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

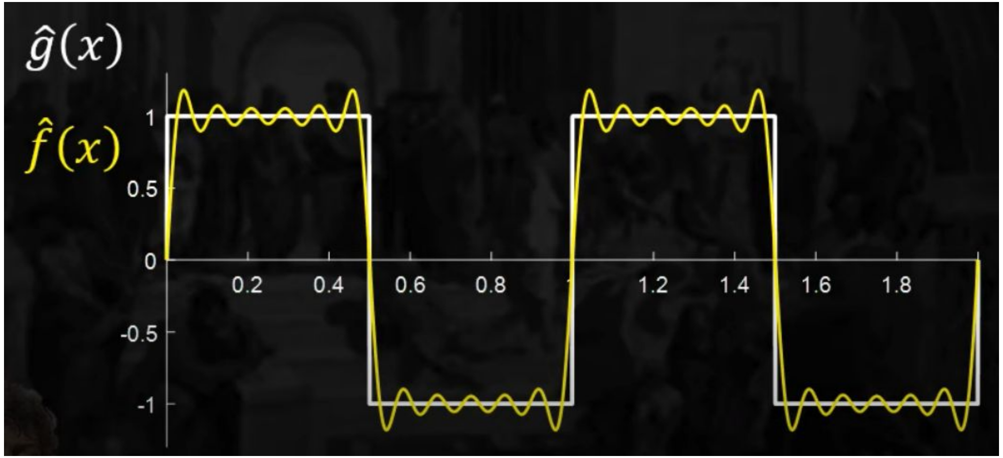
Preliminaries

- **a) Fourier Series**
- b) Euler's formula
- c) Fourier Transform

using **11** sine function

$$\hat{f}(x) = \sum_{n=1,3,5..}^{N=11} \frac{4}{\pi n}\sin\left(\frac{2\pi n x}{1}\right) = \frac{4}{\pi 1}\sin\left(\frac{2\pi 1 x}{1}\right) + \frac{4}{\pi 3}\sin\left(\frac{2\pi 3 x}{1}\right) + \frac{4}{\pi 5}\sin\left(\frac{2\pi 5 x}{1}\right) \ldots + \frac{4}{\pi 7}\sin\left(\frac{2\pi 7 x}{1}\right) + \frac{4}{\pi 9}\sin\left(\frac{2\pi 9 x}{1}\right) + \frac{4}{\pi 11}\sin\left(\frac{2\pi 11 x}{1}\right)$$

Example)



https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

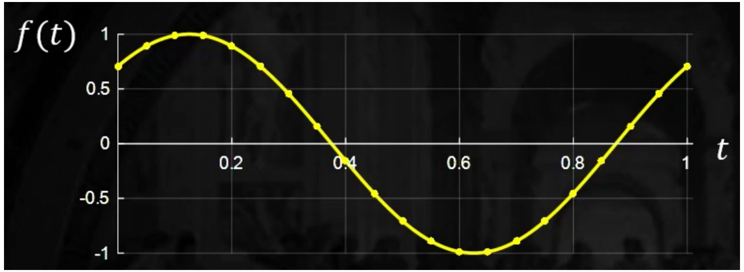
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

$$e^{+i\omega t} = \cos(\omega t) + i\sin(\omega t)$$



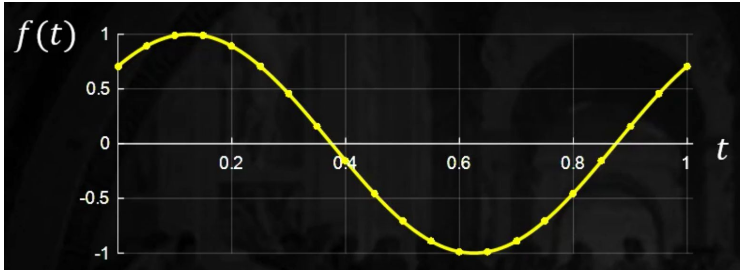
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

$$e^{+i\omega t} = \cos(\omega t) + i\sin(\omega t)$$



$$f(t) = 1 * \sin(2\pi * (t + 0.125))$$

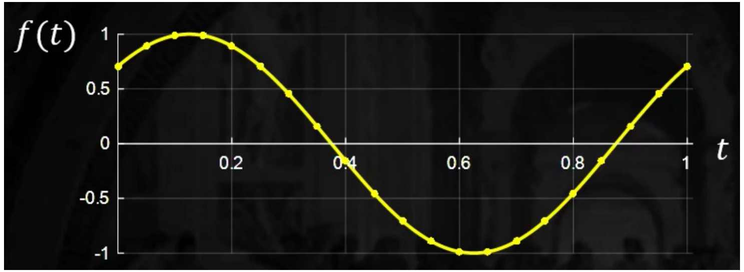
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

$$e^{+i\omega t} = \cos(\omega t) + i\sin(\omega t)$$



$$f(t) = 1 * \sin(2\pi * (t + 0.125))$$

$$f(t) = 1 * \cos(2\pi * (t - 0.125))$$

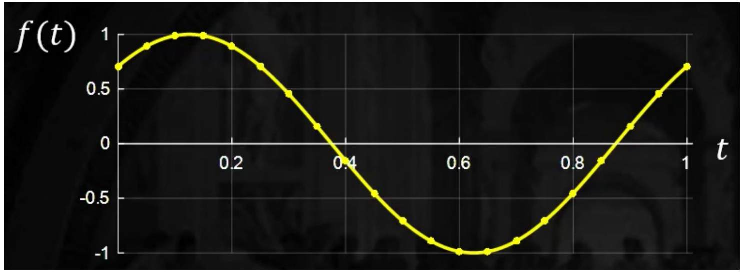
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

$$e^{+i\omega t} = \cos(\omega t) + i\sin(\omega t)$$



$$f(t) = 1 * \sin(2\pi * (t + 0.125))$$
$$f(t) = 1 * \cos(2\pi * (t - 0.125))$$
$$f(t) = \frac{\sqrt{2}}{2}\sin(2\pi * t) + \frac{\sqrt{2}}{2}\cos(2\pi * t)$$

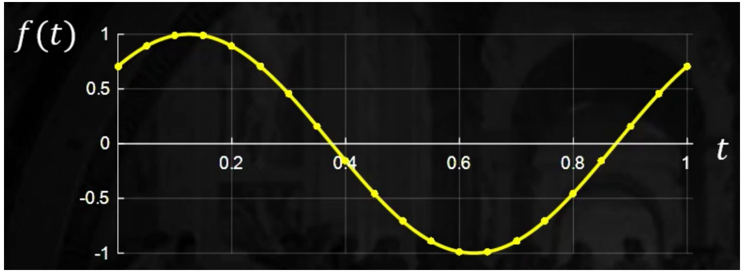
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

$$e^{+i\omega t} = \cos(\omega t) + i\sin(\omega t)$$



$$f(t) = 1 * \sin(2\pi * (t + 0.125))$$
$$f(t) = 1 * \cos(2\pi * (t - 0.125))$$
$$f(t) = \frac{\sqrt{2}}{2}\sin(2\pi * t) + \frac{\sqrt{2}}{2}\cos(2\pi * t)$$
$$f(t) = \mathrm{Re}\{1 * e^{i2\pi * (t-0.125)}\}$$

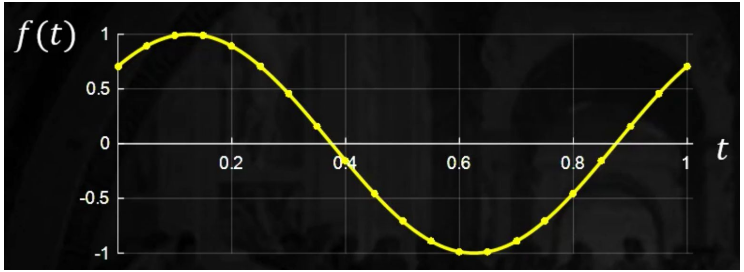
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

$$e^{+i\omega t} = \cos(\omega t) + i\sin(\omega t)$$



$$f(t) = 1 * \sin(2\pi * (t + 0.125))$$

$$f(t) = 1 * \cos(2\pi * (t - 0.125))$$

$$f(t) = \frac{\sqrt{2}}{2}\sin(2\pi * t) + \frac{\sqrt{2}}{2}\cos(2\pi * t)$$

$$f(t) = \mathrm{Re}\{1 * e^{i2\pi*(t-0.125)}\}$$

$$f(t) = A * e^{i2\pi f*(t-\varphi)}$$

amplitude frequency phase

https://www.youtube.com/watch?v=60cgbKX0fmE

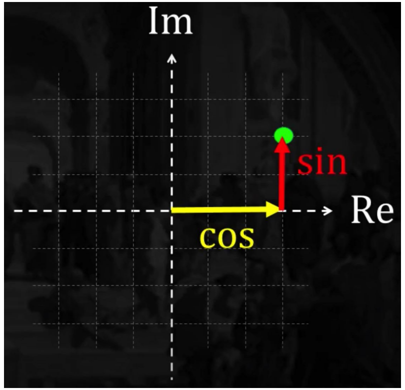
1. Time Series Data in TIME & FREQUENCY Domain

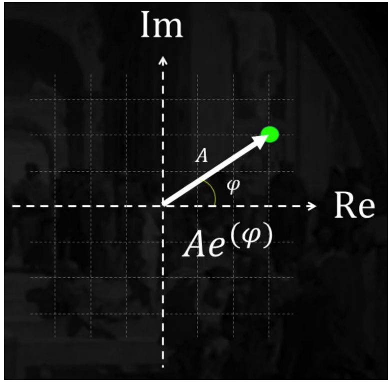
Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)\cos(\omega t)\,dt + i\int_{-\infty}^{\infty} f(t)\sin(\omega t)\,dt$$

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{+i\omega t}\,dt$$





https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- **b) Euler's formula**
- c) Fourier Transform

Inner Product : Relationship between two functions

$$\langle \hat{f}, \hat{g} \rangle = \int f(t)g(t)\, dt$$

Relationships between **sin & cos** => **ORTHOGONAL**

$$\langle \hat{f}, \hat{g} \rangle = \int \sin(t)\cos(t)\, dt = 0$$

Thus, able to express any periodical function, using sin & cos!

https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- b) Euler's formula
- **c) Fourier Transform**

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt$$

↓

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) [\cos(\omega t) + i\sin(\omega t)]\, dt$$

↓

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) \cos(\omega t)\, dt + i \int_{-\infty}^{\infty} f(t) \sin(\omega t) dt$$

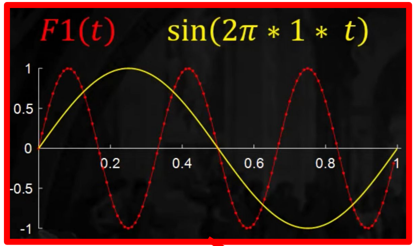
https://www.youtube.com/watch?v=60cgbKX0fmE

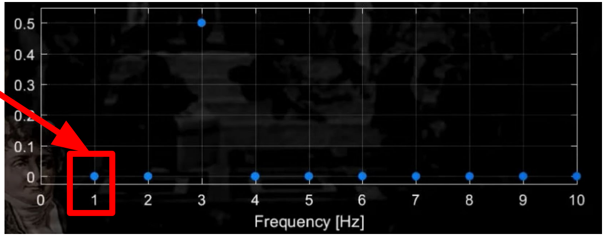
1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- b) Euler's formula
- **c) Fourier Transform**

calculate using inner product : $\langle \hat{f}, \hat{g} \rangle = \int f(t)g(t)\,dt$





. . .

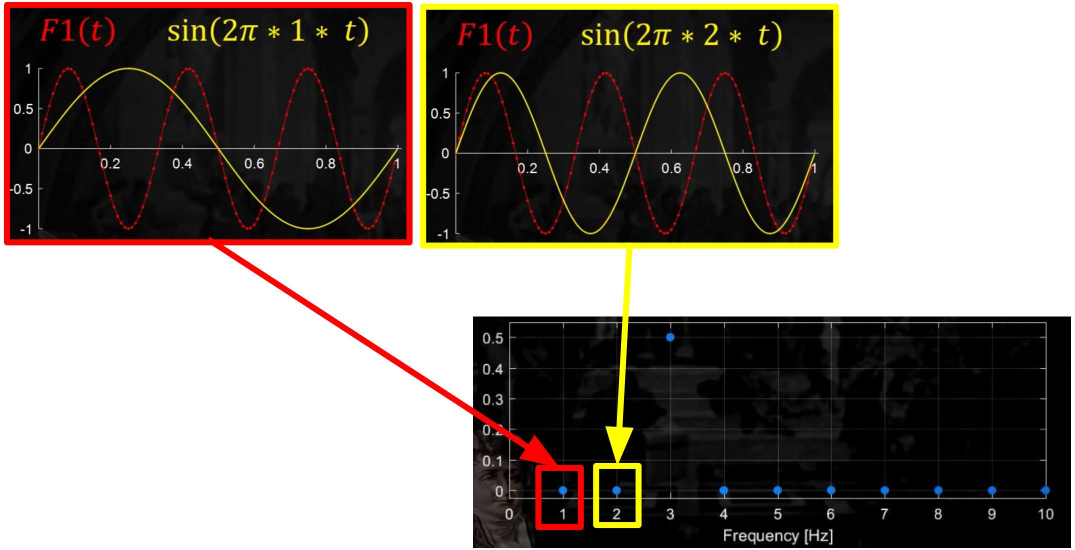
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- b) Euler's formula
- **c) Fourier Transform**

calculate using inner product : $\langle \hat{f}, \hat{g} \rangle = \int f(t)g(t)\, dt$



...

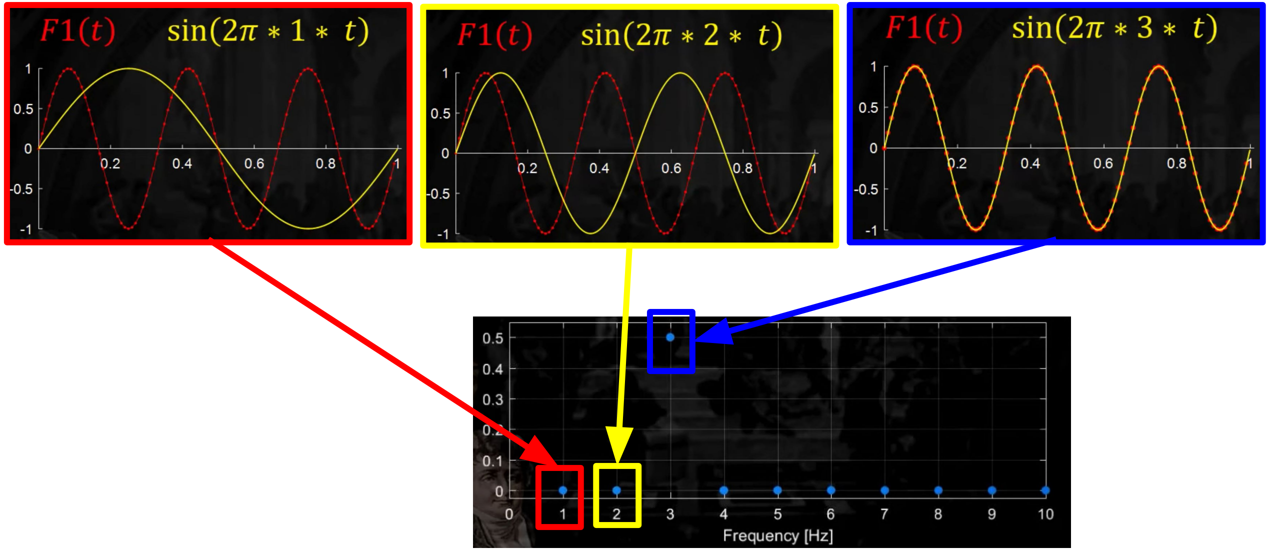
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- b) Euler's formula
- **c) Fourier Transform**

calculate using inner product : $\langle \hat{f}, \hat{g} \rangle = \int f(t)g(t)\,dt$



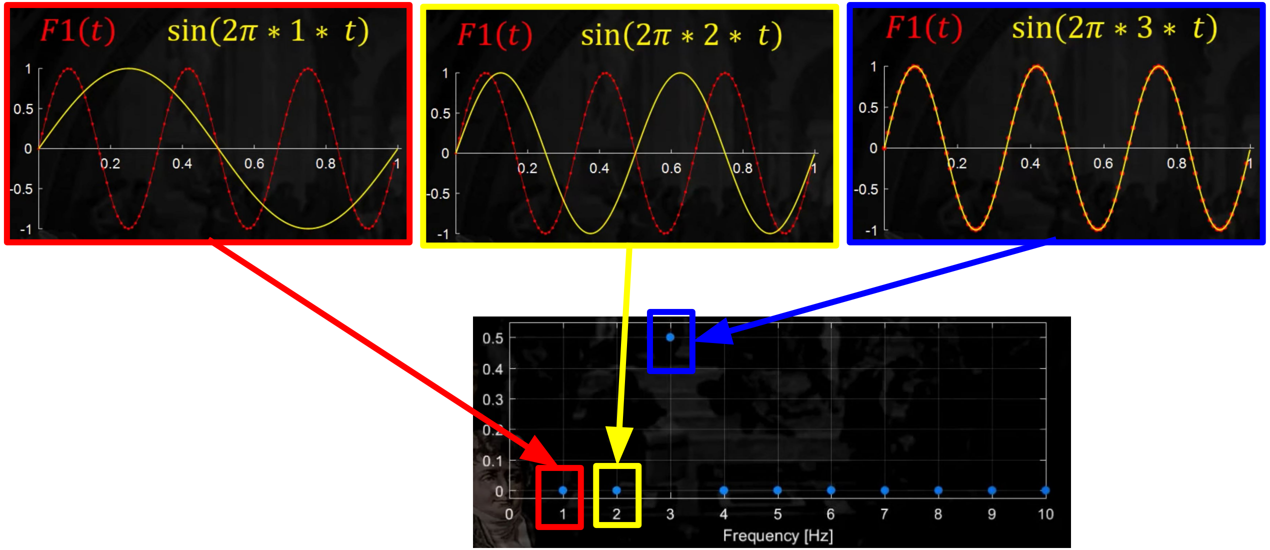
https://www.youtube.com/watch?v=60cgbKX0fmE

1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

- a) Fourier Series
- b) Euler's formula
- **c) Fourier Transform**

calculate using inner product : $\langle \hat{f}, \hat{g} \rangle = \int f(t)g(t)\,dt$



...

https://www.youtube.com/watch?v=60cgbKX0fmE

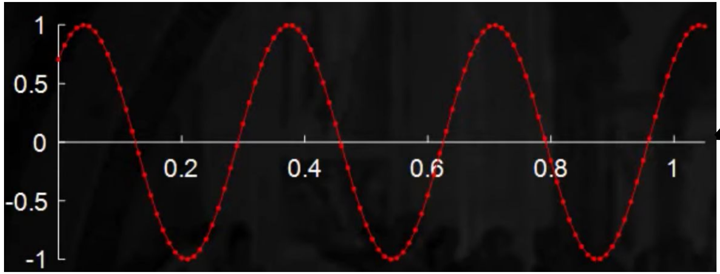
1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

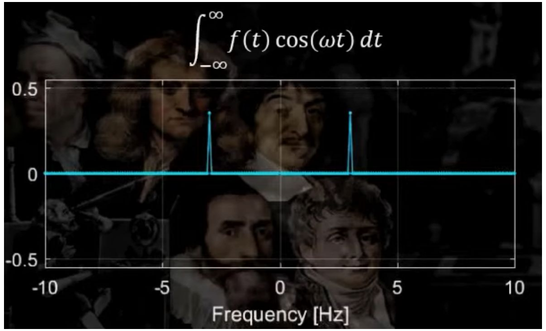
- a) Fourier Series
- b) Euler's formula
- **c) Fourier Transform**

Example :

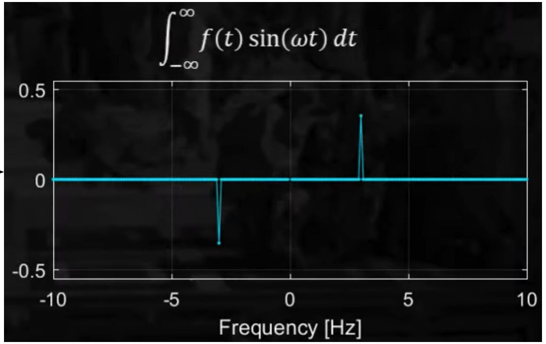
$$f(t) = \frac{\sqrt{2}}{2}\cos(2\pi\, 3 * t) + \frac{\sqrt{2}}{2}\sin(2\pi\, 3 * t)$$



cos



sin



https://www.youtube.com/watch?v=60cgbKX0fmE

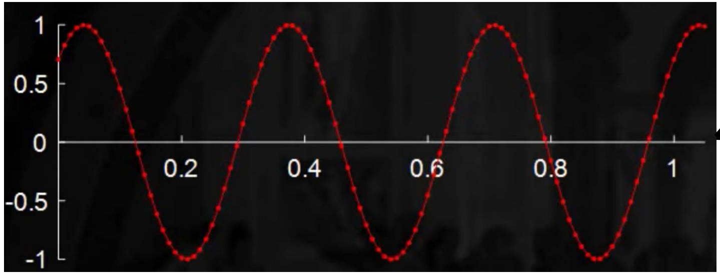
1. Time Series Data in TIME & FREQUENCY Domain

Preliminaries

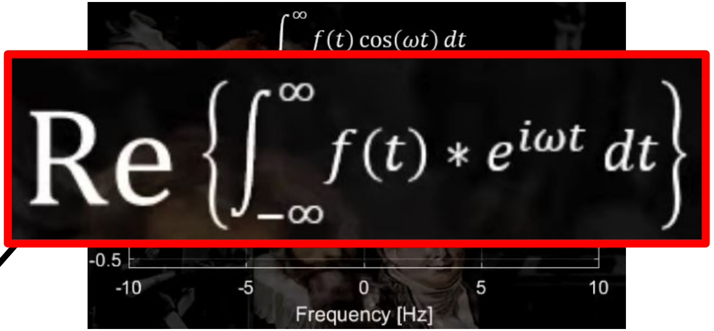
- a) Fourier Series
- b) Euler's formula
- **c) Fourier Transform**

Example :

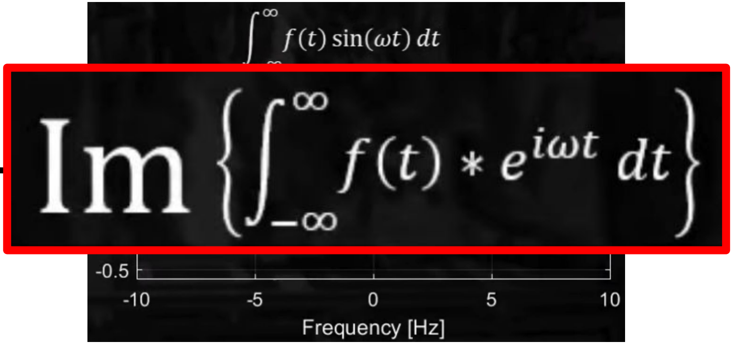
$$f(t) = \frac{\sqrt{2}}{2}\cos(2\pi\, 3 * t) + \frac{\sqrt{2}}{2}\sin(2\pi\, 3 * t)$$



cos



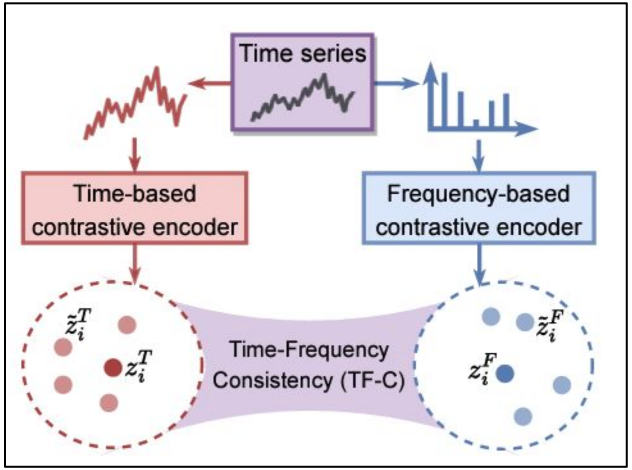
sin



https://www.youtube.com/watch?v=60cgbKX0fmE

2. Abstract

Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency



Expect that **TIME-based** and **FREQUENCY-based** representations of the same data

to be located **close together** in the **time frequency space**

2. Abstract

Contributions

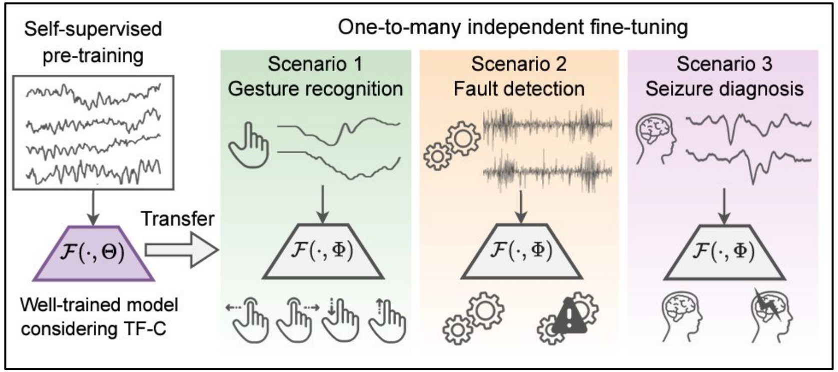
- adopts **contrastive learning in**
  - **in TIME space**
  - **in FREQUENCY space**
  - **in TIME & Frequency space**
- propose a set of **novel augmentations**
  - based on the characteristic of **frequency spectrum**
  - first work to implement augmentation in frequency domain
- evaluate the new method on eight datasets

3. Time-Frequency Consistency (TF-C)

Problem Formulation

a) Notation

- pre-training dataset : $\mathcal{D}^{\text{pret}} = \left\{ \boldsymbol{x}_i^{\text{pret}} \mid i = 1, \ldots, N \right\}$ .... (unlabeled)
  - $\boldsymbol{x}_i^{\text{pret}}$: K^{pret} channels & L^{pret} time-stamps

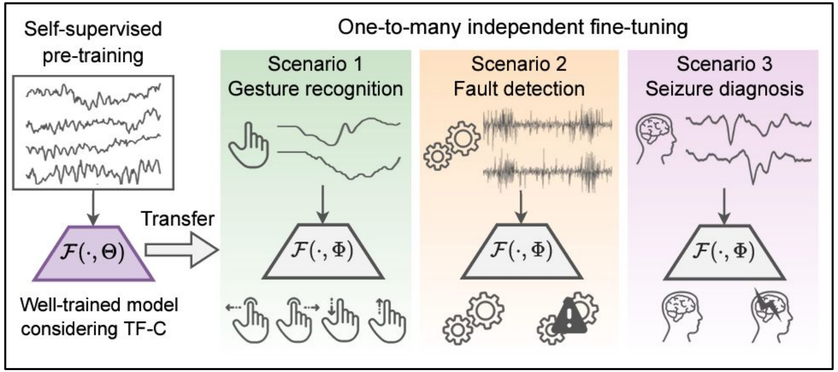


3. Time-Frequency Consistency (TF-C)

Problem Formulation

a) Notation

- pre-training dataset : $\mathcal{D}^{\text{pret}} = \left\{ \boldsymbol{x}_i^{\text{pret}} \mid i = 1, \ldots, N \right\}$ .... (unlabeled)
  - $\boldsymbol{x}_i^{\text{pret}}$: K^{pret} channels & L^{pret} time-stamps
- fine-tuning dataset : $\mathcal{D}^{\text{tune}} = \{(\boldsymbol{x}_i^{\text{tune}}, y_i) \mid i = 1, \ldots, M\}$ .... (labeled)
  - class label : $y_i \in \{1, \ldots, C\}$
  - $(M \ll N)$.

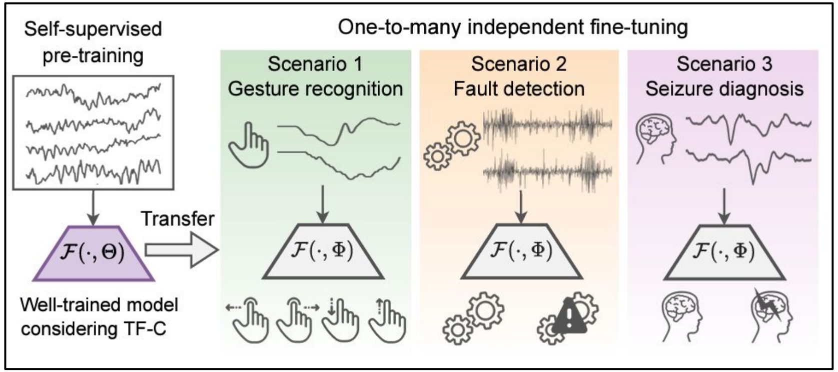


3. Time-Frequency Consistency (TF-C)

Problem Formulation

a) Notation

- pre-training dataset : $\mathcal{D}^{\text{pret}} = \left\{ \boldsymbol{x}_i^{\text{pret}} \mid i = 1, \ldots, N \right\}$ .... (unlabeled)
  - $\boldsymbol{x}_i^{\text{pret}}$: K^{pret} channels & L^{pret} time-stamps
- fine-tuning dataset : $\mathcal{D}^{\text{tune}} = \{ (\boldsymbol{x}_i^{\text{tune}}, y_i) \mid i = 1, \ldots, M \}$ .... (labeled)
  - class label : $y_i \in \{1, \ldots, C\}$
  - $(M \ll N)$.
- Input time series : $\boldsymbol{x}_i^{\text{T}} \equiv \boldsymbol{x}_i$
- Frequency spectrum : $\boldsymbol{x}_i^{\text{F}}$



3. Time-Frequency Consistency (TF-C)

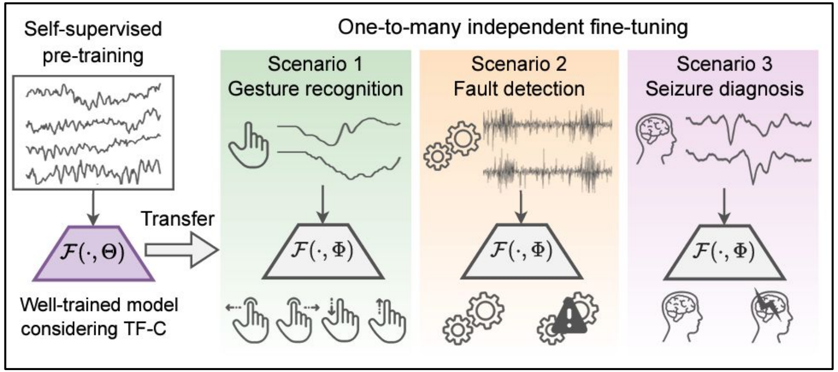
Problem Formulation

b) Problem

= Self-Supervised Contrastive Pretraining for TS

Goal : use $\mathcal{D}^{\text{pret}}$ to pre-train $\mathcal{F}$

$\rightarrow$ generate a generalizable representation $\boldsymbol{z}_i^{\text{tune}} = \mathcal{F}(\boldsymbol{x}_i^{\text{tune}})$



- $\mathcal{F}$ is pre-trained on $\mathcal{D}^{\text{pret}}$ & Θ are fine-tuned using $\mathcal{D}^{\text{tune}}$
  - $\mathcal{F}(\cdot, \Theta)$ to $\mathcal{F}(\cdot, \Phi)$ using dataset $\mathcal{D}^{\text{tune}}$
- NOT a domain adaptation !!

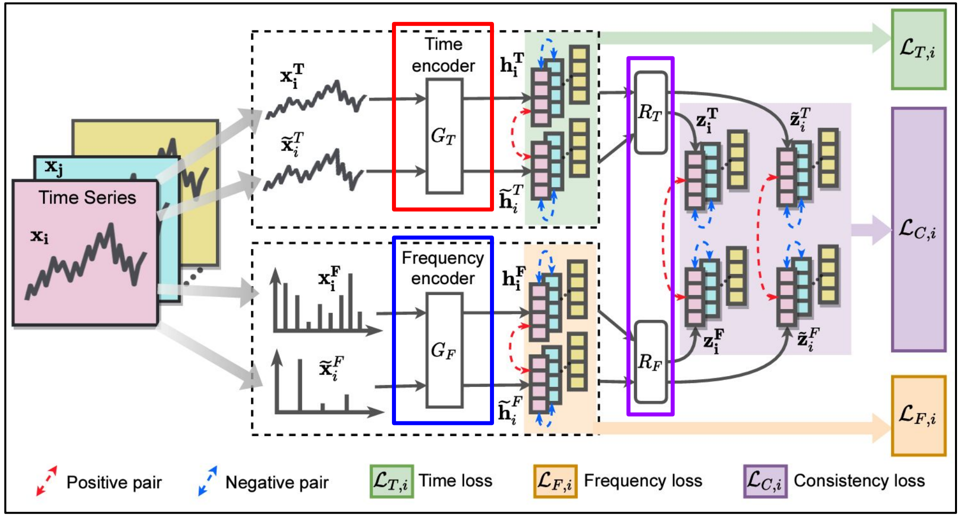
3. Time-Frequency Consistency (TF-C)

Model Architecture

$\mathcal{F}$: 4 components

- (1) time encoder : G_{T}
- (2) frequency encoder : G_{F}
- (3) two cross-space projectors : (map to time-frequency space)
  - (3-1) for time domain : R_{T}
  - (3-2) for frequency domain : R_{F}

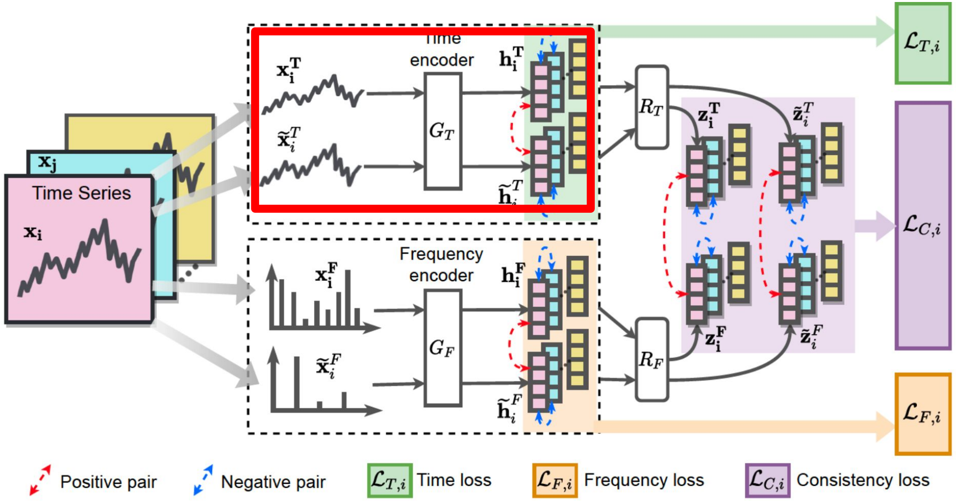
→ 4 components embed $\boldsymbol{x}_i$ to the latent time-frequency space



3. Time-Frequency Consistency (TF-C)

Model Architecture

a) Time Based Contrastive Encoder



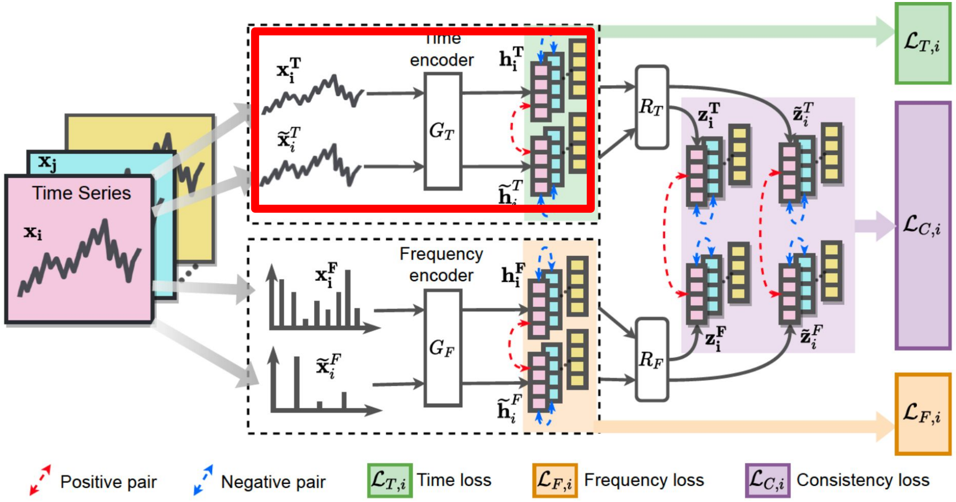
3. Time-Frequency Consistency (TF-C)

Model Architecture

a) Time Based Contrastive Encoder

Data Augmentation

- input : $\boldsymbol{x}_i$
- Augmentation : $\mathcal{B}^{\mathrm{T}} : \boldsymbol{x}_i^{\mathrm{T}} \rightarrow \mathcal{X}_i^{\mathrm{T}}$
- output : (set) $\mathcal{X}_i^{\mathrm{T}}$ .... $\tilde{\boldsymbol{x}}_i^{\mathrm{T}} \in \mathcal{X}_i^{\mathrm{T}}$
  - augmented based on temporal characteristics



3. Time-Frequency Consistency (TF-C)

Model Architecture

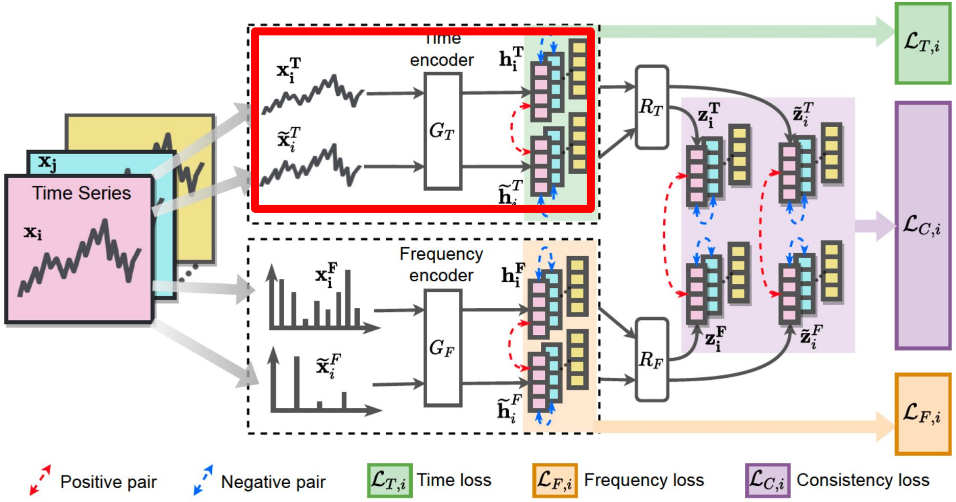
a) Time Based Contrastive Encoder

Data Augmentation

- input : $\boldsymbol{x}_i$
- Augmentation : $\mathcal{B}^{\mathrm{T}} : \boldsymbol{x}_i^{\mathrm{T}} \rightarrow \mathcal{X}_i^{\mathrm{T}}$
- output : (set) $\mathcal{X}_i^{\mathrm{T}}$ .... $\tilde{\boldsymbol{x}}_i^{\mathrm{T}} \in \mathcal{X}_i^{\mathrm{T}}$
  - augmented based on temporal characteristics

Time-based augmentation bank

- ex) jittering, scaling, time-shifts, and neighborhood segments ....
- use diverse augmentations
  - make more robust time-based embeddings!



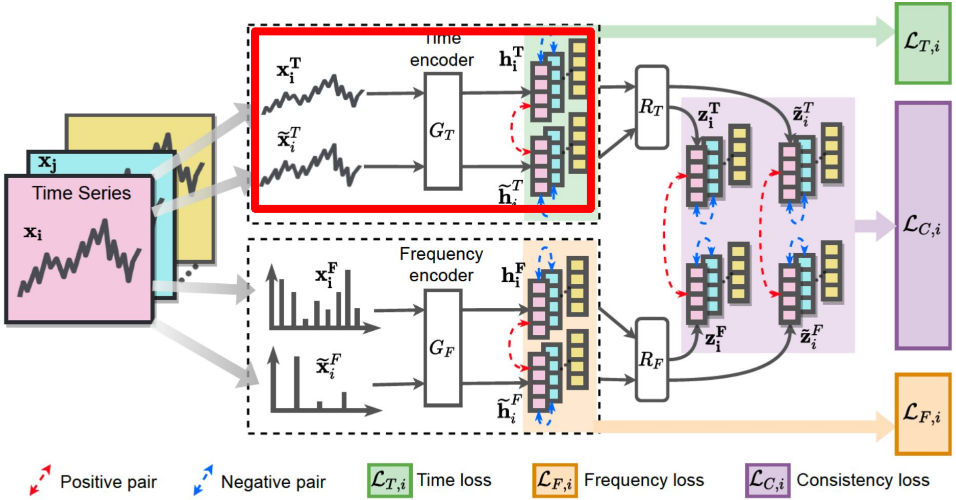
3. Time-Frequency Consistency (TF-C)

Model Architecture

a) Time Based Contrastive Encoder

Procedure

- step 1) randomly select an augmented sample $\tilde{\boldsymbol{x}}_i^{\mathrm{T}} \in \mathcal{X}_i^{\mathrm{T}}$



3. Time-Frequency Consistency (TF-C)

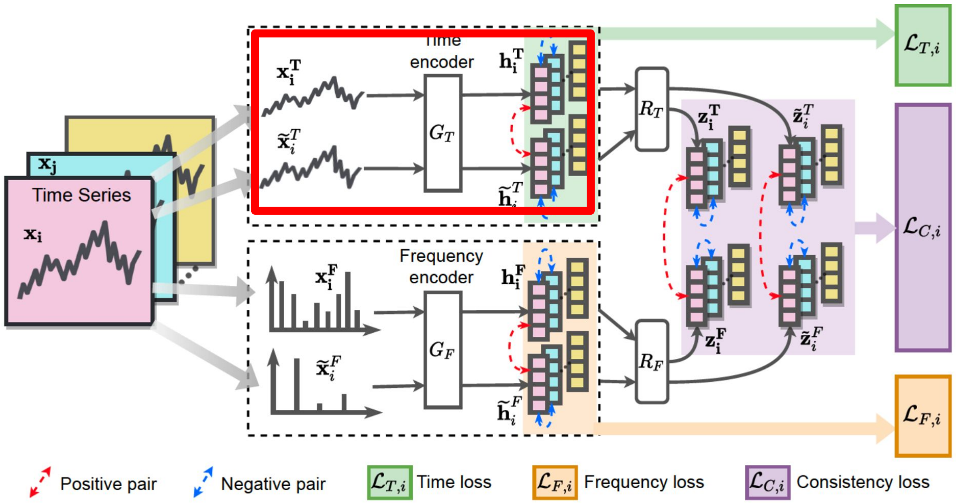
Model Architecture

a) Time Based Contrastive Encoder

Procedure

- step 1) randomly select an augmented sample $\widetilde{\boldsymbol{x}}_i^{\mathrm{T}} \in \mathcal{X}_i^{\mathrm{T}}$
- step 2) feed into a contrastive time encoder G_{T}
  - $\boldsymbol{h}_i^{\mathrm{T}} = G_{\mathrm{T}}\left(\boldsymbol{x}_i^{\mathrm{T}}\right)$ & $\widetilde{\boldsymbol{h}}_i^{\mathrm{T}} = G_{\mathrm{T}}\left(\widetilde{\boldsymbol{x}}_i^{\mathrm{T}}\right)$
  - assume these two are close, if from same i

    (far, if different i)
  - pos & neg pairs :
    - pos pairs : $\left(\boldsymbol{x}_i^{\mathrm{T}}, \widetilde{\boldsymbol{x}}_i^{\mathrm{T}}\right)$
    - neg pairs : $\left(\boldsymbol{x}_i^{\mathrm{T}}, \boldsymbol{x}_j^{\mathrm{T}}\right)$ and $\left(\boldsymbol{x}_i^{\mathrm{T}}, \widetilde{\boldsymbol{x}}_j^{\mathrm{T}}\right)$



3. Time-Frequency Consistency (TF-C)

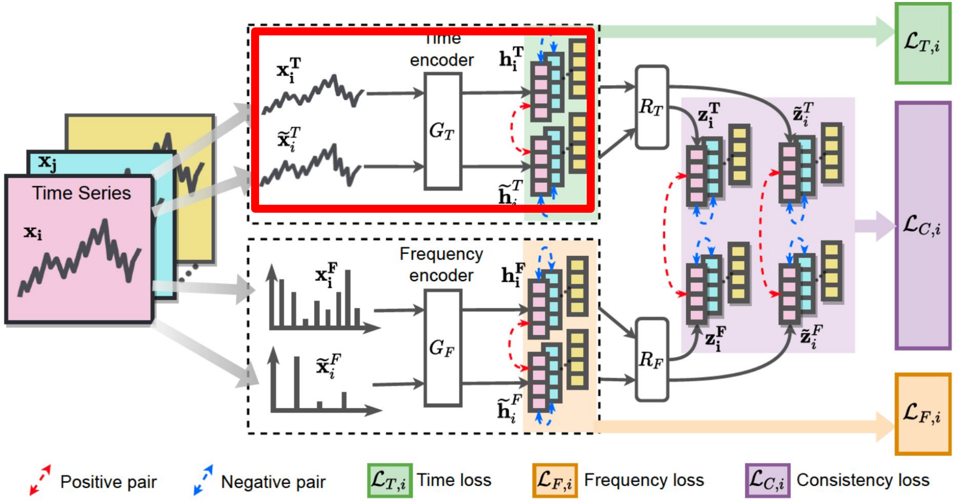
Model Architecture

a) Time Based Contrastive Encoder

Procedure

- step 1) randomly select an augmented sample $\tilde{\boldsymbol{x}}_i^{\mathrm{T}} \in \mathcal{X}_i^{\mathrm{T}}$
- step 2) feed into a contrastive time encoder G_{T}
  - $\boldsymbol{h}_i^{\mathrm{T}} = G_{\mathrm{T}}\left(\boldsymbol{x}_i^{\mathrm{T}}\right)$ & $\tilde{\boldsymbol{h}}_i^{\mathrm{T}} = G_{\mathrm{T}}\left(\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\right)$
  - assume these two are close, if from same i

    (far, if different i)
  - pos & neg pairs :
    - pos pairs : $\left(\boldsymbol{x}_i^{\mathrm{T}}, \tilde{\boldsymbol{x}}_i^{\mathrm{T}}\right)$
    - neg pairs : $\left(\boldsymbol{x}_i^{\mathrm{T}}, \boldsymbol{x}_j^{\mathrm{T}}\right)$ and $\left(\boldsymbol{x}_i^{\mathrm{T}}, \tilde{\boldsymbol{x}}_j^{\mathrm{T}}\right)$
- step 3) calculate contrastive time loss



3. Time-Frequency Consistency (TF-C)

Model Architecture

a) Time Based Contrastive Encoder

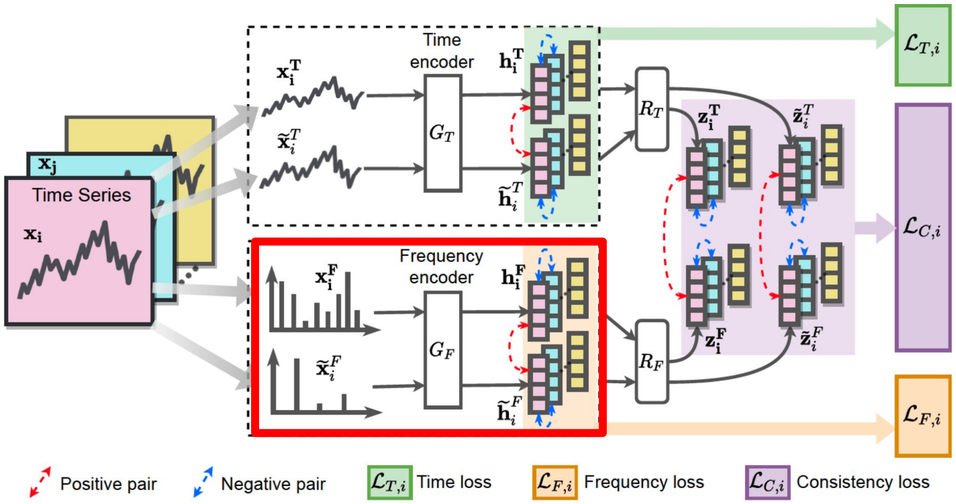
Contrastive Time Loss

- adopt the NT-Xent (the normalized temperature-scaled cross entropy loss)
- $$\mathcal{L}_{\mathrm{T},i} = d\left(\boldsymbol{h}_i^{\mathrm{T}}, \widetilde{\boldsymbol{h}}_i^{\mathrm{T}}, \mathcal{D}^{\mathrm{pret}}\right) = -\log \frac{\exp\left(\mathrm{sim}\left(\boldsymbol{h}_i^{\mathrm{T}}, \widetilde{\boldsymbol{h}}_i^{\mathrm{T}}\right)/\tau\right)}{\sum_{\boldsymbol{x}_j \in \mathcal{D}^{\mathrm{pret}}} \mathbb{1}_{i \neq j} \exp\left(\mathrm{sim}\left(\boldsymbol{h}_i^{\mathrm{T}}, G_{\mathrm{T}}(\boldsymbol{x}_j)\right)/\tau\right)}$$
  - where $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$
  - $\boldsymbol{x}_j \in \mathcal{D}^{\mathrm{pret}}$: different TS sample and its augmented sample

3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder



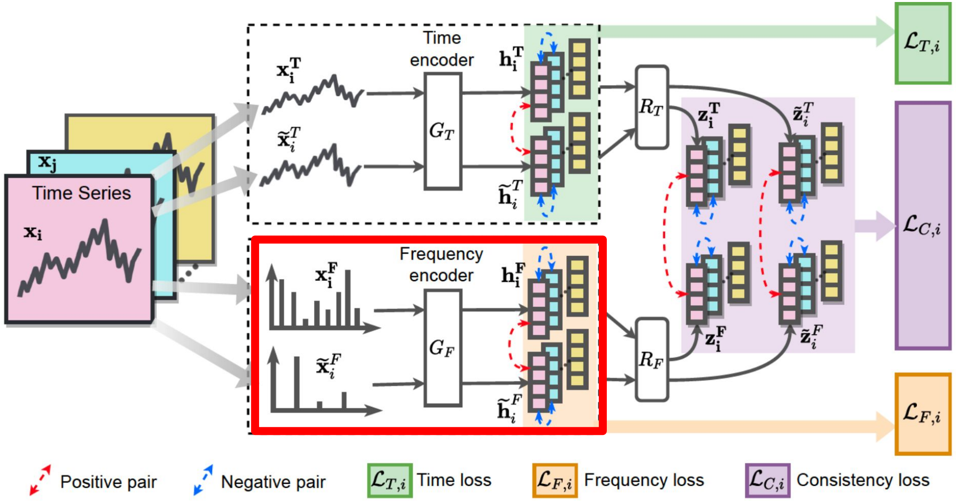
3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder

Frequency Transformation

- input : $\boldsymbol{x}_i$
- transformation : transform operator ($e.g.,$ Fourier Transformation)
- output : $\boldsymbol{x}_i^{\mathrm{F}}$



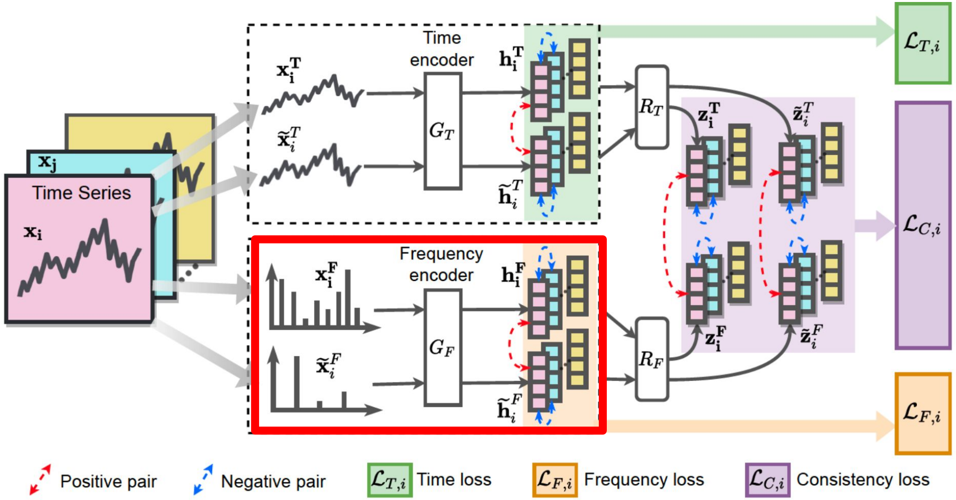
3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder

Frequency Transformation

- input : $\boldsymbol{x}_i$
- transformation : transform operator ($e.g.,$ Fourier Transformation)
- output : $\boldsymbol{x}_i^{\mathrm{F}}$



Frequency component

= base function (e.g., sinusoidal function for Fourier transformation) with the corresponding **frequency and amplitude**

3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder

Augmentation

- perturb $\boldsymbol{x}_i^{\mathrm{F}}$ based on characteristics of frequency spectra
  - perturb the frequency spectrum by adding/removing frequency components
- (small perturbation in freq spectrum $\rightarrow$ may cause large change in time domain)

3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder

Augmentation

- perturb $\boldsymbol{x}_i^{\mathrm{F}}$ based on characteristics of frequency spectra
  - perturb the frequency spectrum by adding/removing frequency components
- (small perturbation in freq spectrum $\rightarrow$ may cause large change in time domain)

Frequency-augmentation bank

- input : $\boldsymbol{x}_i$
- augmentation : $\mathcal{B}^{\mathrm{F}} : \boldsymbol{x}_i^{\mathrm{F}} \rightarrow \mathcal{X}_i^{\mathrm{F}}$
  - 2 methods : removing or adding
- output : (set) $\mathcal{X}_i^{\mathrm{F}}$ .... $| \mathcal{X}_i^{\mathrm{Y}} | = 2$

3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder

Small Budget E

use E in perturbation,

- where E : # of frequency components we manipulate

To removing frequency components ...

$\rightarrow$ randomly select E frequency components & set their amplitudes as 0

3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder

Small Budget E

use E in perturbation,

- where E : # of frequency components we manipulate

To removing frequency components ...

$\rightarrow$ randomly select E frequency components & set their amplitudes as 0

To add frequency components ...

$\rightarrow$ randomly choose E frequency components

- from the ones that have smaller amplitude than $\alpha \cdot A_m$
- increase their amplitude to $\alpha \cdot A_m$.
  - A_m : maximum amplitude
  - α : pre-defined coefficient (set 0.5)

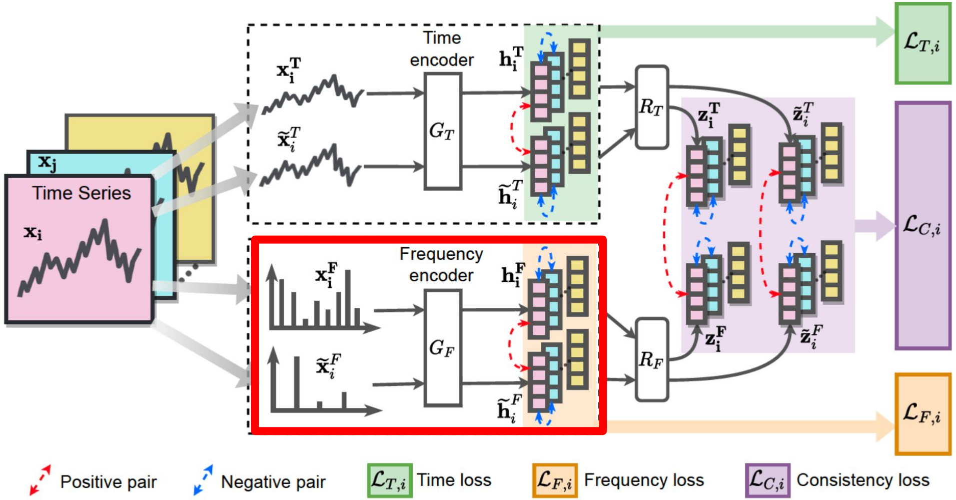
3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder

Procedure

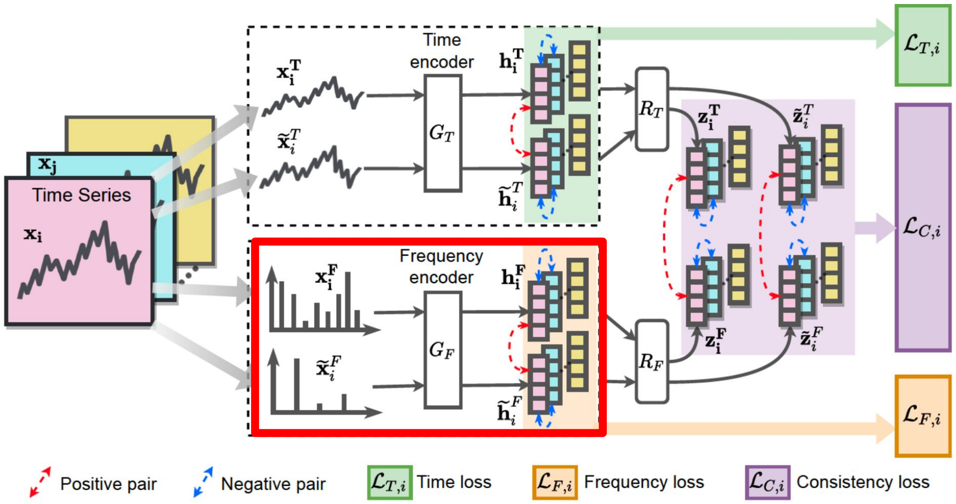
- step 1) $\boldsymbol{h}_i^{\mathrm{F}} = G_{\mathrm{F}}\left(\boldsymbol{x}_i^{\mathrm{F}}\right)$
- step 2) set pos & neg pairs :
  - pos pairs : $\left(\boldsymbol{x}_i^{\mathrm{F}}, \tilde{\boldsymbol{x}}_i^{\mathrm{F}}\right)$
  - neg pairs : $\left(\boldsymbol{x}_i^{\mathrm{F}}, \boldsymbol{x}_j^{\mathrm{F}}\right)$ and $\left(\boldsymbol{x}_i^{\mathrm{F}}, \tilde{\boldsymbol{x}}_j^{\mathrm{F}}\right)$
- step 3) calculate frequency-based contrastive loss



3. Time-Frequency Consistency (TF-C)

Model Architecture

b) Frequency Based Contrastive Encoder



Procedure

- step 1) $\boldsymbol{h}_i^{\mathrm{F}} = G_{\mathrm{F}}\left(\boldsymbol{x}_i^{\mathrm{F}}\right)$
- step 2) set pos & neg pairs :
  - pos pairs : $\left(\boldsymbol{x}_i^{\mathrm{F}}, \tilde{\boldsymbol{x}}_i^{\mathrm{F}}\right)$
  - neg pairs : $\left(\boldsymbol{x}_i^{\mathrm{F}}, \boldsymbol{x}_j^{\mathrm{F}}\right)$ and $\left(\boldsymbol{x}_i^{\mathrm{F}}, \tilde{\boldsymbol{x}}_j^{\mathrm{F}}\right)$
- step 3) calculate frequency-based contrastive loss

Contrastive frequency loss

- $\mathcal{L}_{\mathrm{F},i} = d\left(\boldsymbol{h}_i^{\mathrm{F}}, \widetilde{\boldsymbol{h}}_i^{\mathrm{F}}, \mathcal{D}^{\mathrm{pret}}\right) = -\log \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{h}_i^{\mathrm{F}}, \widetilde{\boldsymbol{h}}_i^{\mathrm{F}}\right)/\tau\right)}{\sum_{\boldsymbol{x}_j \in \mathcal{D}^{\mathrm{pret}}} \mathbb{1}_{i \neq j} \exp\left(\operatorname{sim}\left(\boldsymbol{h}_i^{\mathrm{F}}, G_{\mathrm{F}}(\boldsymbol{x}_j)\right)/\tau\right)}$

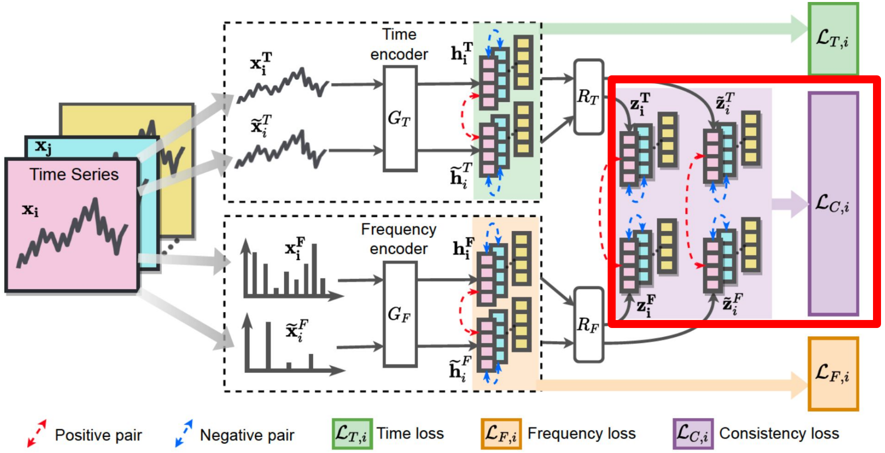
3. Time-Frequency Consistency (TF-C)

Time-Frequency Consistency

Consistency loss $\mathcal{L}_{C,i}$

- to urge the learned embeddings to satisfy TF-C

  $\rightarrow$ ***time-based & frequency-based embeddings : CLOSE !***



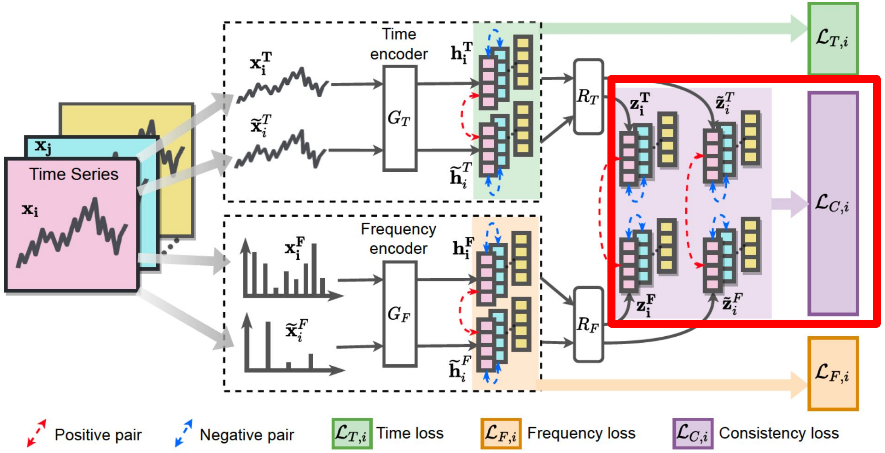
3. Time-Frequency Consistency (TF-C)

Time-Frequency Consistency

Consistency loss $\mathcal{L}_{\mathrm{C},i}$

- to urge the learned embeddings to satisfy TF-C

  → ***time-based & frequency-based embeddings : CLOSE !***
- $\boldsymbol{z}_i^{\mathrm{T}} = R_{\mathrm{T}}\left(\boldsymbol{h}_i^{\mathrm{T}}\right), \widetilde{\boldsymbol{z}}_i^{\mathrm{T}} = R_{\mathrm{T}}\left(\widetilde{\boldsymbol{h}}_i^{\mathrm{T}}\right).$
  - map $\boldsymbol{h}_i^{\mathrm{T}}$ from time space to a joint time-frequency space with R_{T}
- $\boldsymbol{z}_i^{\mathrm{F}} = R_{\mathrm{F}}\left(\boldsymbol{h}_i^{\mathrm{F}}\right), \widetilde{\boldsymbol{z}}_i^{\mathrm{F}} = R_{\mathrm{F}}\left(\widetilde{\boldsymbol{h}}_i^{\mathrm{F}}\right).$
  - map $\boldsymbol{h}_i^{\mathrm{F}}$ from frequency space to a joint time-frequency space with R_{F}



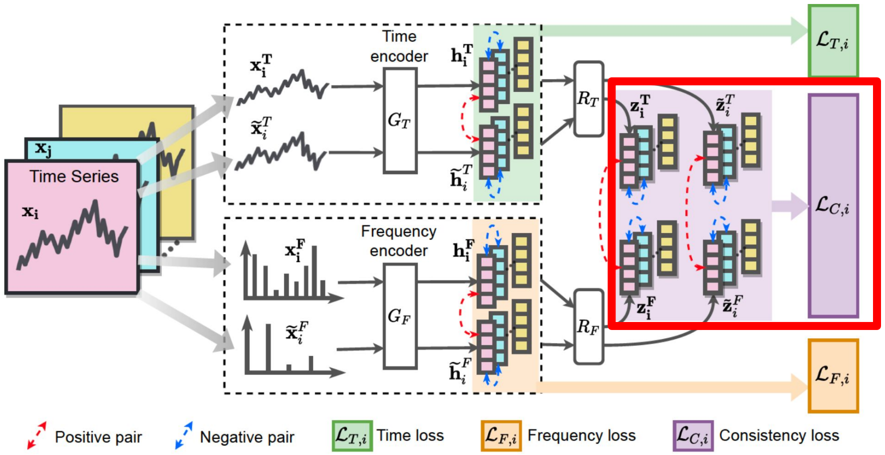
3. Time-Frequency Consistency (TF-C)

Time-Frequency Consistency

$$S_i^{\mathrm{TF}} = d\left(\boldsymbol{z}_i^{\mathrm{T}}, \boldsymbol{z}_i^{\mathrm{F}}, \mathcal{D}^{\mathrm{pret}}\right),$$

- distance between $\boldsymbol{z}_i^{\mathrm{T}}$ and $\boldsymbol{z}_i^{\mathrm{F}}$

  (define S_i^{TF}, $S_i^{\widetilde{T}F}$, and $S_i^{T\widetilde{F}}$ similarly)



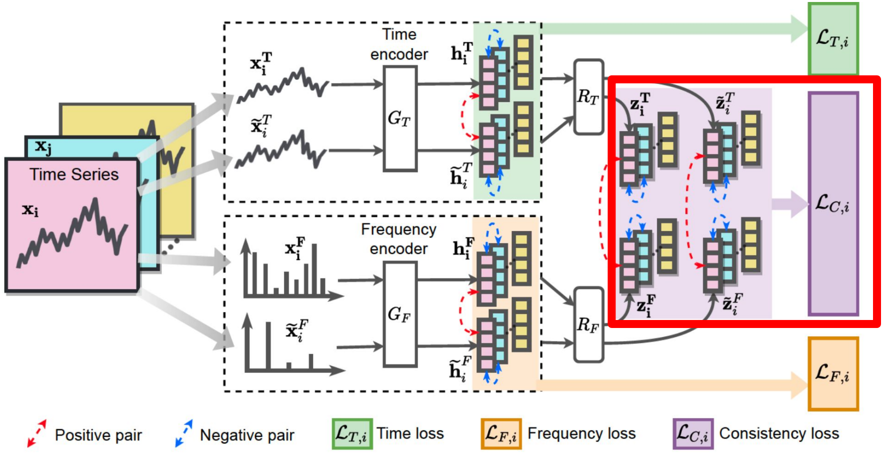
3. Time-Frequency Consistency (TF-C)

Time-Frequency Consistency

$$S_i^{\mathrm{TF}} = d\left(\boldsymbol{z}_i^{\mathrm{T}}, \boldsymbol{z}_i^{\mathrm{F}}, \mathcal{D}^{\mathrm{pret}}\right),$$

- distance between $\boldsymbol{z}_i^{\mathrm{T}}$ and $\boldsymbol{z}_i^{\mathrm{F}}$

(define $S_i^{\mathrm{T}\widetilde{\mathrm{F}}}$, $S_i^{\widetilde{T}F}$, and $S_i^{\widetilde{T}\widetilde{F}}$ similarly)



intuitively, $\boldsymbol{z}_i^{\mathrm{T}}$ should be closer to $\boldsymbol{z}_i^{\mathrm{F}}$ in comparison to $\tilde{\boldsymbol{z}}_i^{\mathrm{F}}$

→ encourage the proposed model to learn a $S_i^{\mathrm{TF}} < S_i^{\mathrm{T\widetilde{F}}}$

→ (inspired by the triplet loss) design $\left(S_i^{\mathrm{TF}} - S_i^{\mathrm{T\widetilde{F}}} + \delta\right)$ as a term of consistency loss $\mathcal{L}_{\mathrm{C},i}$

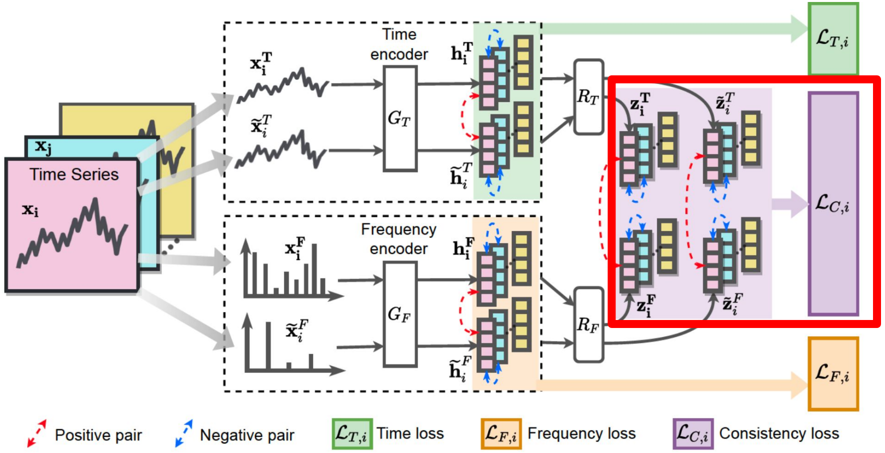
3. Time-Frequency Consistency (TF-C)

Time-Frequency Consistency

$$S_i^{\mathrm{TF}} = d\left(\boldsymbol{z}_i^{\mathrm{T}}, \boldsymbol{z}_i^{\mathrm{F}}, \mathcal{D}^{\mathrm{pret}}\right),$$

- distance between $\boldsymbol{z}_i^{\mathrm{T}}$ and $\boldsymbol{z}_i^{\mathrm{F}}$

(define $S_i^{\mathrm{T\tilde{F}}}$, $S_i^{\tilde{T}F}$, and $S_i^{\tilde{T}\tilde{F}}$ similarly)



intuitively, $\boldsymbol{z}_i^{\mathrm{T}}$ should be closer to $\boldsymbol{z}_i^{\mathrm{F}}$ in comparison to $\tilde{\boldsymbol{z}}_i^{\mathrm{F}}$

→ encourage the proposed model to learn a $S_i^{\mathrm{TF}} < S_i^{\mathrm{T\tilde{F}}}$

→ (inspired by the triplet loss) design $\left(S_i^{\mathrm{TF}} - S_i^{\mathrm{T\tilde{F}}} + \delta\right)$ as a term of consistency loss $\mathcal{L}_{\mathrm{C},i}$

don't consider the distance between $\boldsymbol{z}_i^{\mathrm{T}}$ and $\tilde{\boldsymbol{z}}_i^{\mathrm{T}}$ & distance between $\boldsymbol{z}_i^{\mathrm{F}}$ and $\tilde{\boldsymbol{z}}_i^{\mathrm{F}}$

(where the two embeddings are from the same domain)

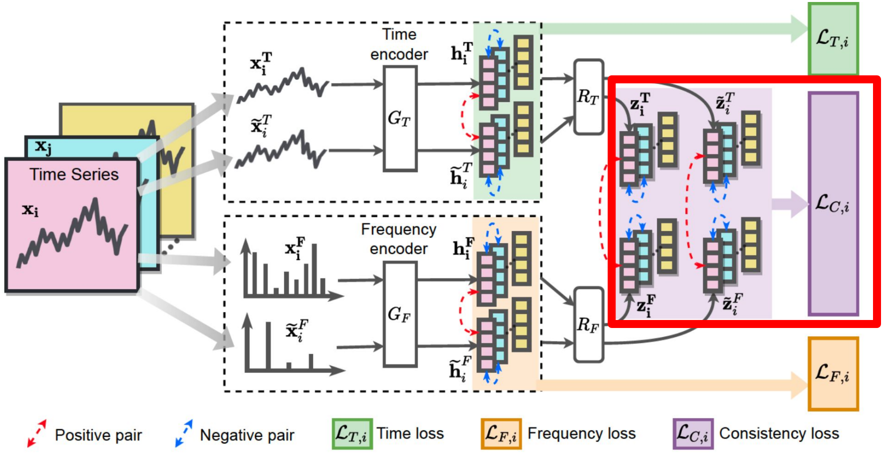
- information is already in $\mathcal{L}_{\mathrm{T},i}$ and $\mathcal{L}_{\mathrm{F},i}$

3. Time-Frequency Consistency (TF-C)

Time-Frequency Consistency

Consistency loss $\mathcal{L}_{\mathrm{C},i}$

$$\mathcal{L}_{\mathrm{C},i} = \sum_{S_{\mathrm{pair}}} \left(S_i^{\mathrm{TF}} - S_i^{\mathrm{pair}} + \delta \right), \quad S^{\mathrm{pair}} \in \left\{ S_i^{\widetilde{\mathrm{T}}\widetilde{\mathrm{F}}}, S_i^{\widetilde{\mathrm{T}}F}, S_i^{T\widetilde{\mathrm{F}}} \right\}$$

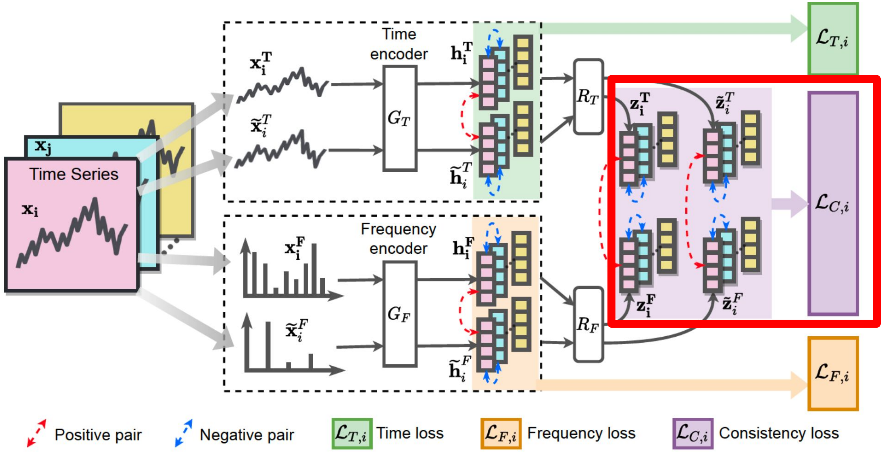


3. Time-Frequency Consistency (TF-C)

Time-Frequency Consistency

Consistency loss $\mathcal{L}_{C,i}$

$$\mathcal{L}_{C,i} = \sum_{S^{\text{pair}}} \left(S_i^{\text{TF}} - S_i^{\text{pair}} + \delta \right), \quad S^{\text{pair}} \in \left\{ S_i^{\widetilde{\text{T}}\widetilde{\text{F}}}, S_i^{\widetilde{\text{T}}F}, S_i^{T\widetilde{\text{F}}} \right\}$$



Instead of Triplet Loss …

why not directly use NT-Xent Loss?

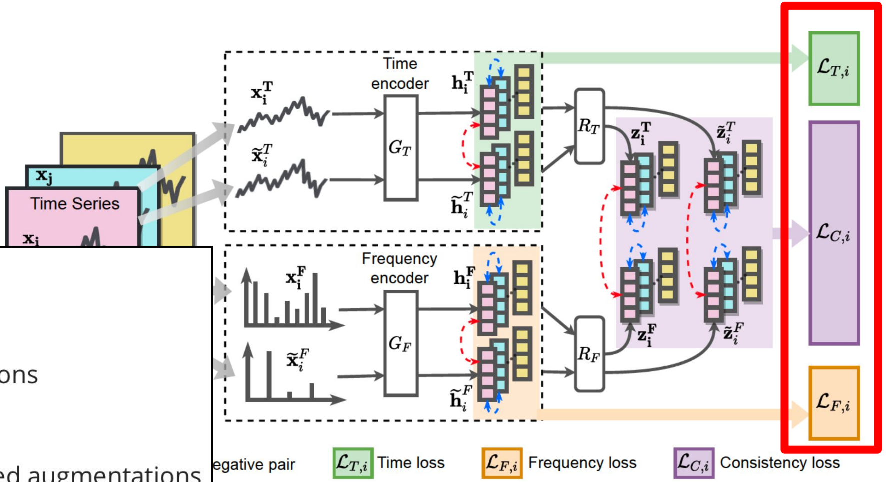
3. Time-Frequency Consistency (TF-C)

Total Loss Function

$$\mathcal{L}_{\text{TF-C},i} = \lambda(\mathcal{L}_{\text{T},i} + \mathcal{L}_{\text{F},i}) + (1-\lambda)\mathcal{L}_{\text{C},i}.$$

overall loss function : 3 terms

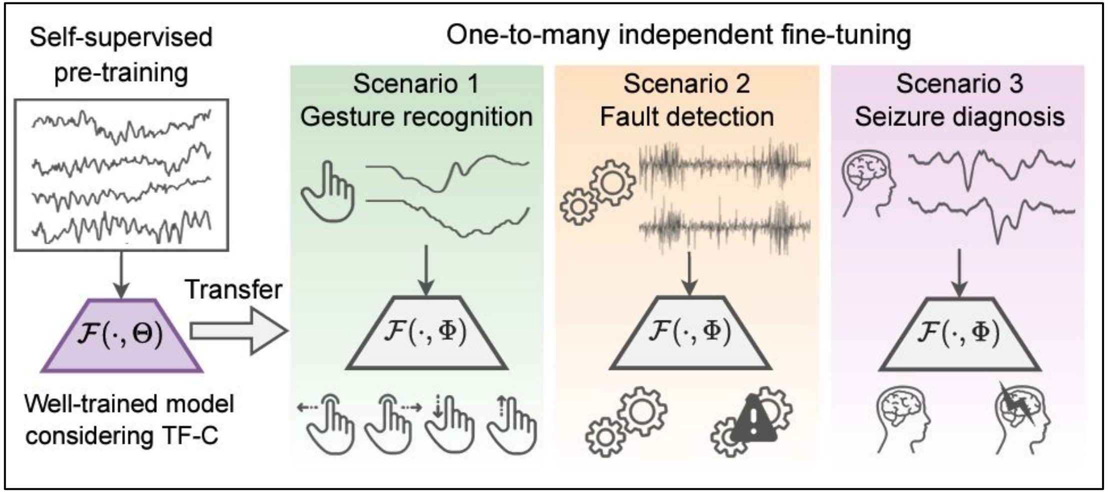
- (1) time-based contrastive loss $\mathcal{L}_{\text{T}}$
  - urges the model to learn embeddings invariant to temporal augmentations
- (2) frequency-based contrastive loss $\mathcal{L}_{\text{F}}$
  - promotes learning of embeddings invariant to frequency spectrum-based augmentations
- (3) consistency loss $\mathcal{L}_{\text{C}}$
  - guides the model to retain the consistency between time-based and frequency-based embeddings.



3. Time-Frequency Consistency (TF-C)

Final Embeddings

$$\boldsymbol{z}_i^{\text{tune}} = \mathcal{F}(\boldsymbol{x}_i^{\text{tune}}, \Phi) = [\boldsymbol{z}_i^{\text{tune,T}}; \boldsymbol{z}_i^{\text{tune,F}}]$$



4. Conclusion

- Contrastive Learning in Frequency domain
- Data Augmentation in Frequency domain
- ***Suggestions***
  - 1) Inverse Fourier Transform after augmentation in Frequency domain?

    (= working in Frequency domain **only in augmentation step**)
    - a) no need for three types of loss functions
    - b) only need one type of encoder
      - inherent Time & Frequency Loss
  - 2) Instead of Triplet loss in TF-C, why not use direct comparison (ex. NT-Xent) across different domains?

Papers

1. Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency (NeurIPS 2022)
2. CoST : Contrastive Learning of Disentangled Seasonal-Trend Representations for TS forecasting (ICLR 2022)

Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency

Xiang Zhang[†] [*]
Harvard University
xiang_zhang@hms.harvard.edu

Ziyuan Zhao[*]
Harvard University
ziyuanzhao@college.harvard.edu

Theodoros Tsiligkaridis
MIT Lincoln Laboratory
ttsili@ll.mit.edu

Marinka Zitnik
Harvard University
marinka@hms.harvard.edu

Abstract

Pre-training on time series poses a unique challenge due to the potential mismatch between pre-training and target domains, such as shifts in temporal dynamics, fast-evolving trends, and long-range and short-cyclic effects, which can lead to poor downstream performance. While domain adaptation methods can mitigate these shifts, most methods need examples directly from the target domain, making them suboptimal for pre-training. To address this challenge, methods need to accommodate target domains with different temporal dynamics and be capable of doing so without seeing any target examples during pre-training. Relative to other modalities, in time series, we expect that time-based and frequency-based representations of the same example are located close together in the time-frequency space. To this end, we posit that time-frequency consistency (TF-C) — embedding a time-based neighborhood of an example close to its frequency-based neighborhood — is desirable for pre-training. Motivated by TF-C, we define a decomposable pre-training model, where the self-supervised signal is provided by

https://arxiv.org/pdf/2206.08496.pdf

Published as a conference paper at ICLR 2022

CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting

Gerald Woo[1 2], **Chenghao Liu**[1 *], **Doyen Sahoo**[1], **Akshat Kumar**[2] **& Steven Hoi**[1]
[1]Salesforce Research Asia, [2]Singapore Management University
{gwoo,chenghao.liu,dsahoo,shoi}@salesforce.com, akshatkumar@smu.edu.sg

Abstract

Deep learning has been actively studied for time series forecasting, and the mainstream paradigm is based on the end-to-end training of neural network architectures, ranging from classical LSTM/RNNs to more recent TCNs and Transformers. Motivated by the recent success of representation learning in computer vision and natural language processing, we argue that a more promising paradigm for time series forecasting, is to first learn disentangled feature representations, followed by a simple regression fine-tuning step – we justify such a paradigm from a causal perspective. Following this principle, we propose a new time series representation learning framework for long sequence time series forecasting named CoST, which applies contrastive learning methods to learn disentangled seasonal-trend representations. CoST comprises both time domain and frequency domain contrastive losses to learn discriminative trend and seasonal representations, respectively. Extensive experiments on real-world datasets show that CoST consistently outperforms the state-of-the-art methods by a considerable margin, achieving a 21.3% improvement in MSE on multivariate benchmarks. It is also robust to various choices of backbone encoders, as well as downstream regressors. Code is available at https://github.com/salesforce/CoST.

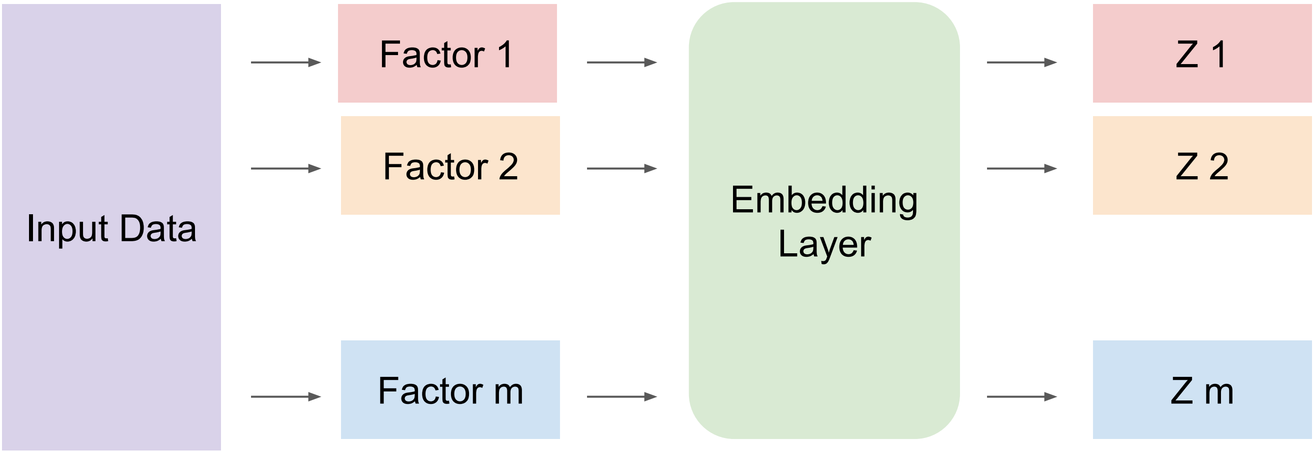
https://arxiv.org/pdf/2202.01575.pdf

Contents

1. Time Series Decomposition

Disentangled Representation Learning

- disentangle features (factors) from input data
- get representation for each factor



1. Time Series Decomposition

Disentangled Representation Learning

- TS in DL : learn (1) **feature representation** & (2) **prediction function** e2e

  => **prone to overfitting**!
- worsens ***when the representations are entangled***!

1. Time Series Decomposition

Disentangled Representation Learning

- TS in DL : learn (1) **feature representation** & (2) **prediction function** e2e

  => **prone to overfitting**!

- worsens ***when the representations are entangled***!

TS = composed of "seasonal module" + "non-linear trend"

- Problem : change in one module, affect other module!

  => how can we **disentangle TREND & SEASONALITY**?

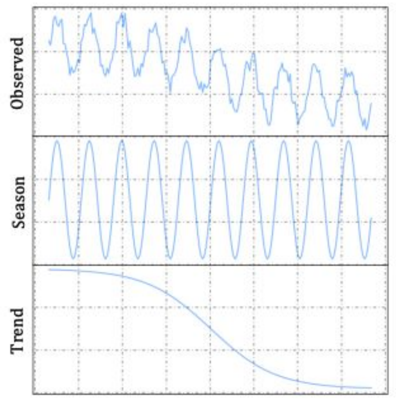


Figure 1: Time series composed of seasonal and trend components.

1. Time Series Decomposition

Decompose Time Series (TS) into..

- (1) Trend
- (2) Seasonality
- (3) Residual

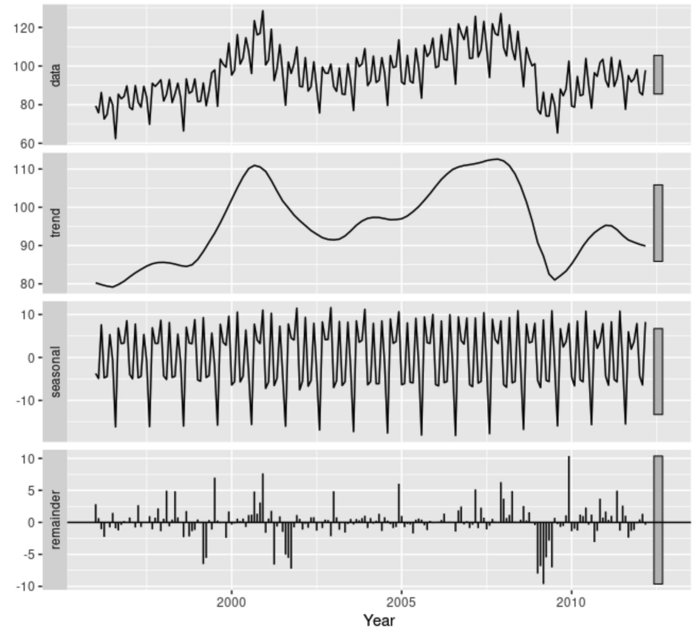
2 ways of TS decomposition

- (1) Additive

$$y_t = S_t + T_t + R_t$$

- (2) Multiplicative

$$y_t = S_t \times T_t \times R_t$$

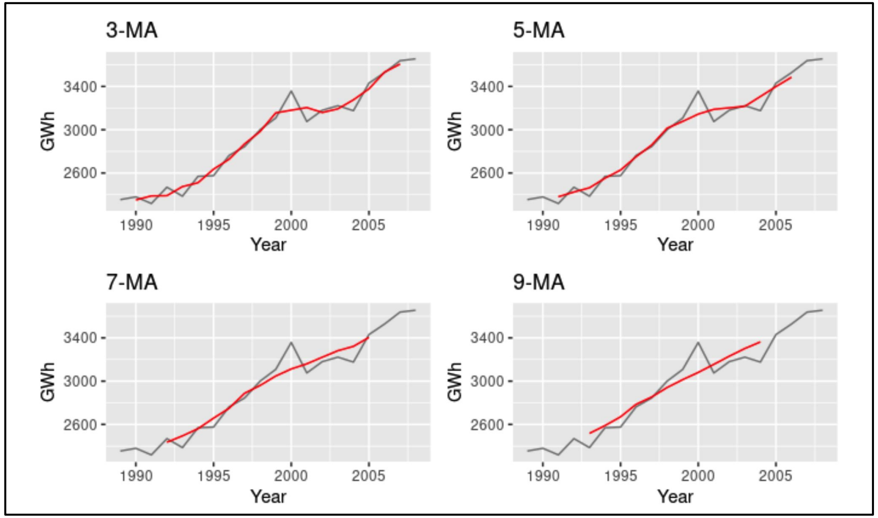


$$y_t = S_t \times T_t \times R_t \quad \text{is equivalent to} \quad \log y_t = \log S_t + \log T_t + \log R_t$$

1. Time Series Decomposition

Ways of extracting Trend from TS :

- ex) **Moving Average** (look back window of H)



2. Abstract

CoST = **Co**ntrastive Learning of Disentangled **S**easonal–**T**rend Representations for TS forecasting

- applies **contrastive learning** methods, to learn **disentangled seasonal-trend representations**
- comprises both
  - (1) **TIME domain** contrastive losses
  - (2) **FREQUENCY domain** contrastive losses
- use **structural time series model**
  - TS = trend + season + error variable

3. CoST

1. Seasonal-Trend Representations

a) Problem Formulation

propose **CL** framework to learn disentangled **seasonal** & **trend** representation for **LTSF** task

- LTSF task : Long Sequence Time-Series Forecasting task

3. CoST

1. Seasonal-Trend Representations

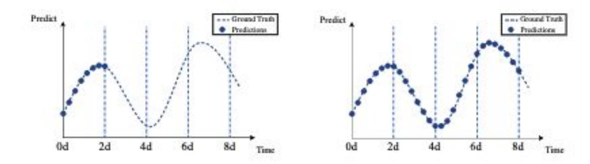
a) Problem Formulation

propose **CL** framework to learn disentangled **seasonal** & **trend** representation for **LTSF** task

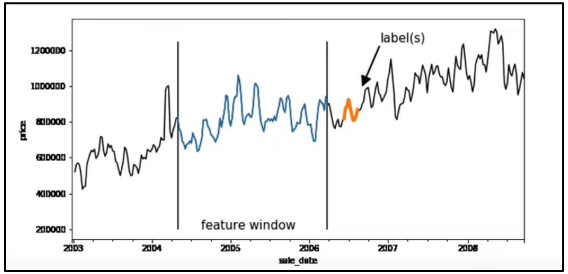
- LTSF task : Long Sequence Time-Series Forecasting task

Notation

- $(\boldsymbol{x}_1, \ldots \boldsymbol{x}_T) \in \mathbb{R}^{T \times m}$: MTS
- h : lookback window
- k : forecasting horizon
- $\hat{\boldsymbol{X}} = g(\boldsymbol{X})$: model
    - $\boldsymbol{X} \in \mathbb{R}^{h \times m}$: input
    - $\hat{\boldsymbol{X}} \in \mathbb{R}^{k \times m}$: output



(a) Short Sequence Forecasting. (b) Long Sequence Forecasting.



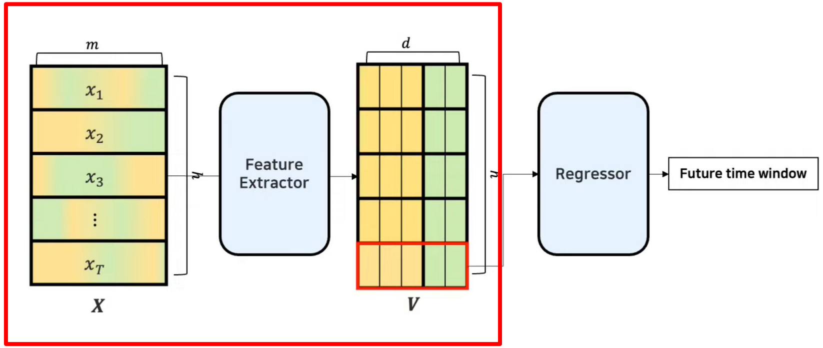
3. CoST

1. Seasonal-Trend Representations

a) Problem Formulation

Not an end-to-end model!

- instead, focus on **learning feature representations from observed data**
- aim to learn a **nonlinear feature embedding** function $\boldsymbol{V} = f(\boldsymbol{X})$,
  - where $\boldsymbol{X} \in \mathbb{R}^{h \times m}$ and $\boldsymbol{V} \in \mathbb{R}^{h \times d}$,
  - map per each timestamp



3. CoST

1. Seasonal-Trend Representations

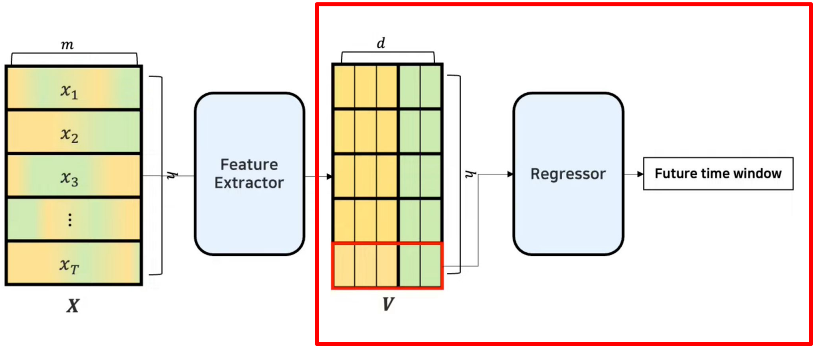
a) Problem Formulation

Not an end-to-end model!

- instead, focus on **learning feature representations from observed data**
- aim to learn a **nonlinear feature embedding** function $\boldsymbol{V} = f(\boldsymbol{X})$,
  - where $\boldsymbol{X} \in \mathbb{R}^{h \times m}$ and $\boldsymbol{V} \in \mathbb{R}^{h \times d}$,
  - map per each timestamp

Then, using the learned representations of the **final timestamp** $\boldsymbol{v}_h$

→ used as inputs for the **downstream regressor of the forecasting task.**



3. CoST

1. Seasonal-Trend Representations

b) Disentangled Seasonal-Trend Representation Learning & Causal Interpretatio

Introduce **structural priors** for TS

- use **Bayesian Structural Time Series Model**

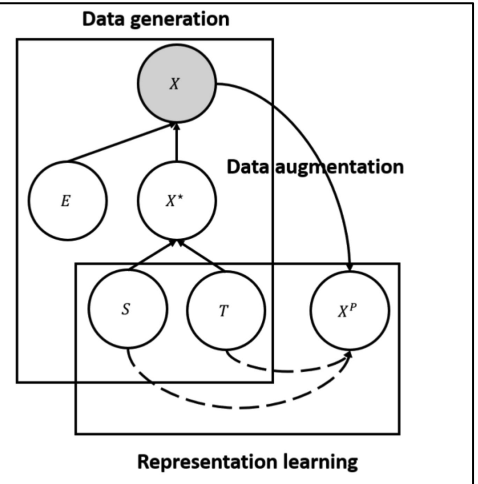


Figure 2: Causal graph of the generative process for time series data.

3. CoST

1. Seasonal-Trend Representations

b) Disentangled Seasonal-Trend Representation Learning & Causal Interpretatio

Introduce **structural priors** for TS

- use **Bayesian Structural Time Series Model**

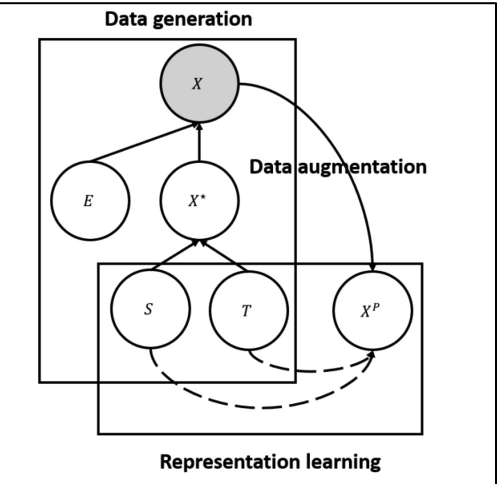


Figure 2: Causal graph of the generative process for time series data.

Assumption 1

- observed TS : X is generated from...
  - (1) E : error variable
  - (2) $X^\star$: error-free latent variable : generated from...
    - (2-1) T : trend variable
    - (2-2) S : seasonal variable
- Since E is not predictable...focus on $X^\star$

3. CoST

1. Seasonal-Trend Representations

b) Disentangled Seasonal-Trend Representation Learning & Causal Interpretatio

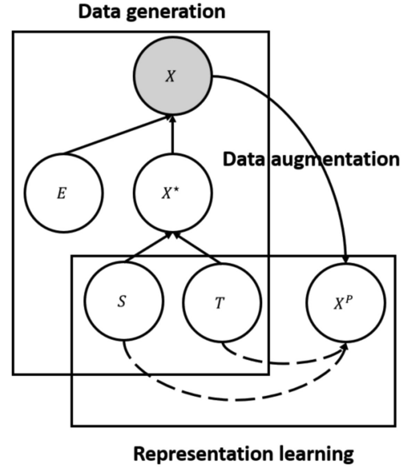


Figure 2: Causal graph of the generative process for time series data.

Introduce **structural priors** for TS

- use **Bayesian Structural Time Series Model**

Assumption 1

- observed TS : X is generated from...
  - (1) E : error variable
  - (2) $X^{\star}$: error-free latent variable : generated from...
    - (2-1) T : trend variable
    - (2-2) S : seasonal variable
- Since E is not predictable...focus on $X^{\star}$

Assumption 2 : Independent mechanism

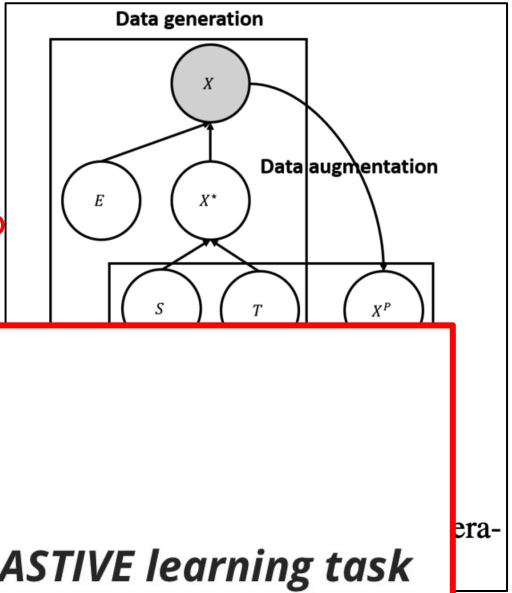
- season & trend do not interact with each other

$\rightarrow$ **disentangle S & T**

3. CoST

1. Seasonal-Trend Representations

b) Disentangled Seasonal-Trend Representation Learning & Causal Interpretatio



Introduce **structural priors** for TS

Learning representations for S & T

- allows us to find stable result
- since targets $X^\star$ are unknown.... ***construct a proxy CONTRASTIVE learning task***

- (1) E : error variable
- (2) $X^\star$: error-free latent variable : generated from...
  - (2-1) T : trend variable
  - (2-2) S : seasonal variable
- Since E is not predictable...focus on $X^\star$

Assumption 2 : Independent mechanism

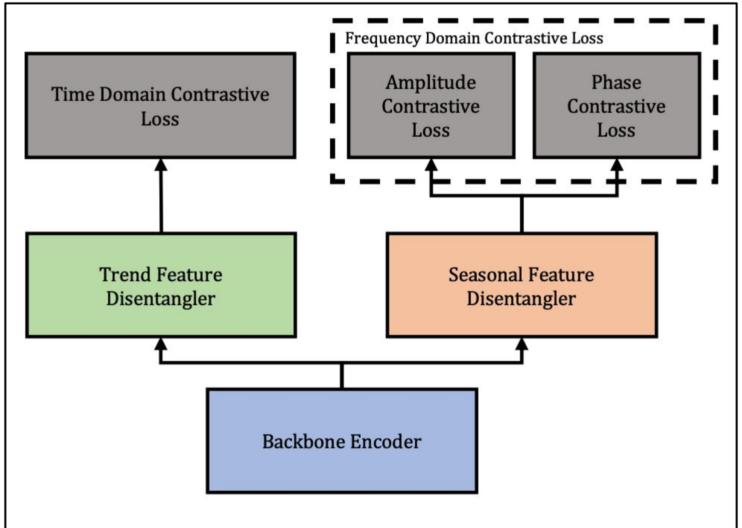
- season & trend do not interact with each other

  $\rightarrow$ **disentangle S & T**

3. CoST

1. Seasonal-Trend Representations

CoST framework



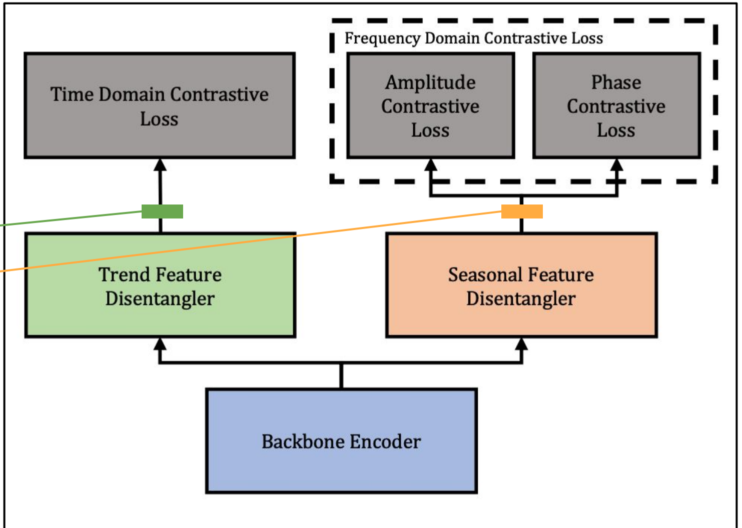
(a) Overall Framework

3. CoST

1. Seasonal-Trend Representations

CoST framework

- learn **disentangled** seasonal-trend reperesentation
- for each time step, have the **disentangled representations for S & T**
  - $\boldsymbol{V} = \left[\boldsymbol{V}^{(T)}; \boldsymbol{V}^{(S)}\right] \in \mathbb{R}^{h \times d}$.
    - trend : $\boldsymbol{V}^{(T)} \in \mathbb{R}^{h \times d_T}$
    - season : $\boldsymbol{V}^{(S)} \in \mathbb{R}^{h \times d_S}$



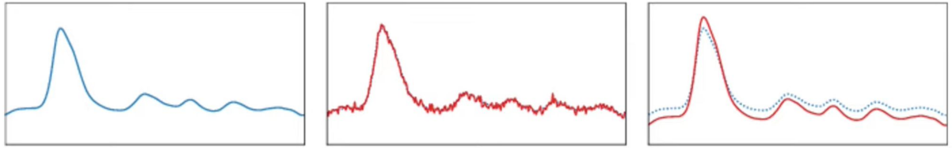
(a) Overall Framework

3. CoST

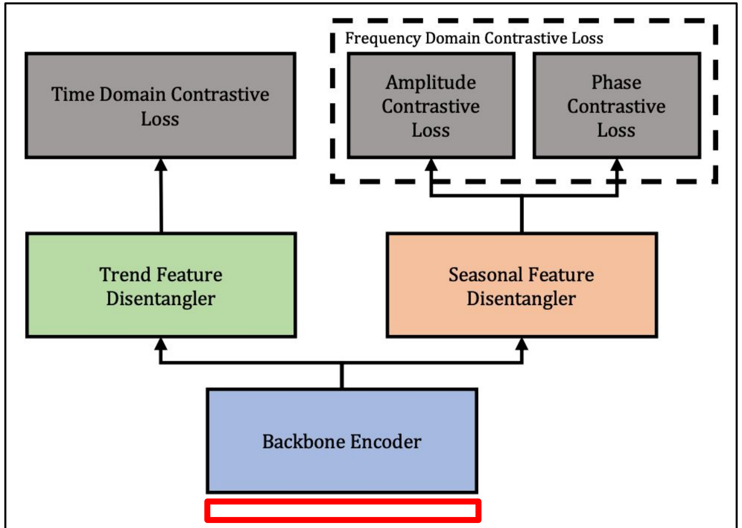
2. Seasonal-Trend Contrastive Learning Framework

step 0) Augmentation

- obtain robustness to error variables
- uses ..
    - 1) **scaling**
    - 2) **shifting**
    - 3) **jittering**



(a) Original (b) Jittering (c) Scaling



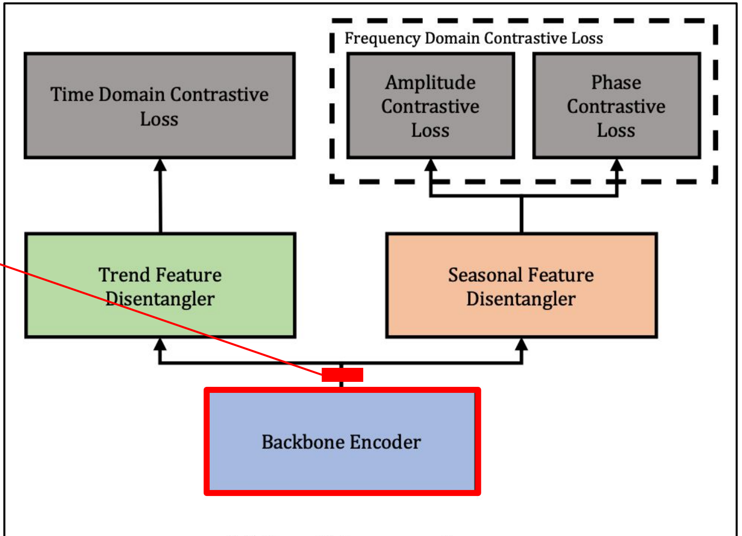
(a) Overall Framework

3. CoST

2. Seasonal-Trend Contrastive Learning Framework

step 1) Encoder

- encoder $f_b : \mathbb{R}^{h \times m} \rightarrow \mathbb{R}^{h \times d}$
- map into latent space (= intermediate representation)



(a) Overall Framework

3. CoST

2. Seasonal-Trend Contrastive Learning Framework

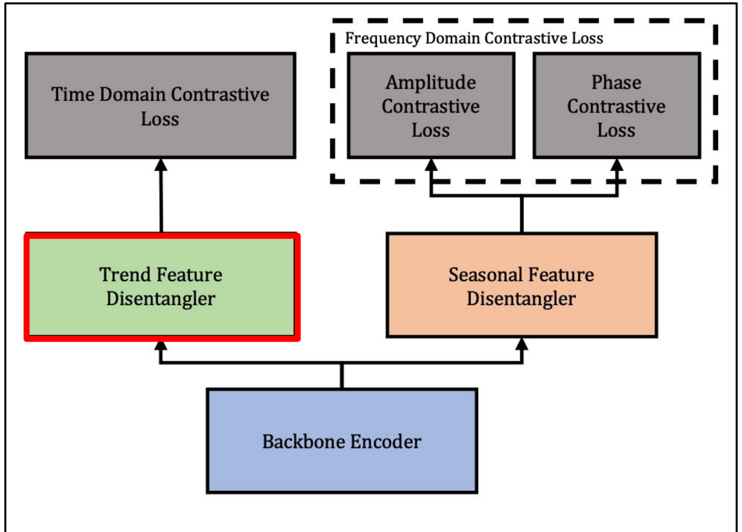
step 2) Trend & Seasonal Representation

(1) TFD (Trend Feature Disentagler) : $f_T : \mathbb{R}^{h \times d} \rightarrow \mathbb{R}^{h \times d_T}$

- extracts trend representation,
- via a **mixture of AR experts**
- learned via a **time domain constrastive loss** L_{time}

$$\mathcal{L} = \mathcal{L}_{\text{time}} + \frac{\alpha}{2}(\mathcal{L}_{\text{amp}} + \mathcal{L}_{\text{phase}})$$

- α : trade-off between T & S



(a) Overall Framework

3. CoST

2. Seasonal-Trend Contrastive Learning Framework

step 2) Trend & Seasonal Representation

(1) TFD (Trend Feature Disentagler) : $f_T : \mathbb{R}^{h \times d} \rightarrow \mathbb{R}^{h \times d_T}$

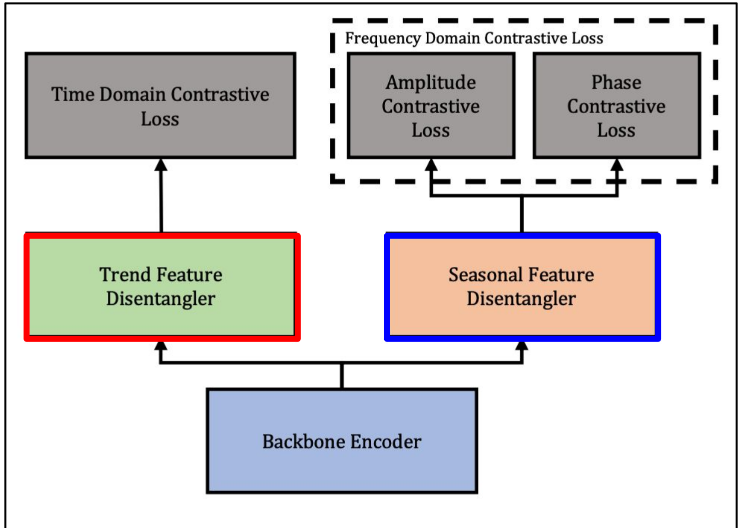
- extracts trend representation,
- via a **mixture of AR experts**
- learned via a **time domain constrastive loss** L_{time}

(2) SFD (Seasonal Feature Disentagler) : $f_S : \mathbb{R}^{h \times d} \rightarrow \mathbb{R}^{h \times d_S}$

- extracts seasonal representation,
- via a **learnable Fourier layer**
- learned via **frequency domain constrastive loss**, which consists of
  - a) L_{amp} : amplitude component
  - b) L_{phase} : phase component

$$\mathcal{L} = \mathcal{L}_{\text{time}} + \frac{\alpha}{2}(\mathcal{L}_{\text{amp}} + \mathcal{L}_{\text{phase}})$$

- α : trade-off between T & S



(a) Overall Framework

3. CoST

2. Seasonal-Trend Contrastive Learning Framework

step 3) Concatenate

Concatenate the outputs of **Trend and Seasonal Feature Disentaglers**,

to obtain final output representations

3. CoST

3. Trend & Seasonal Feature Representation

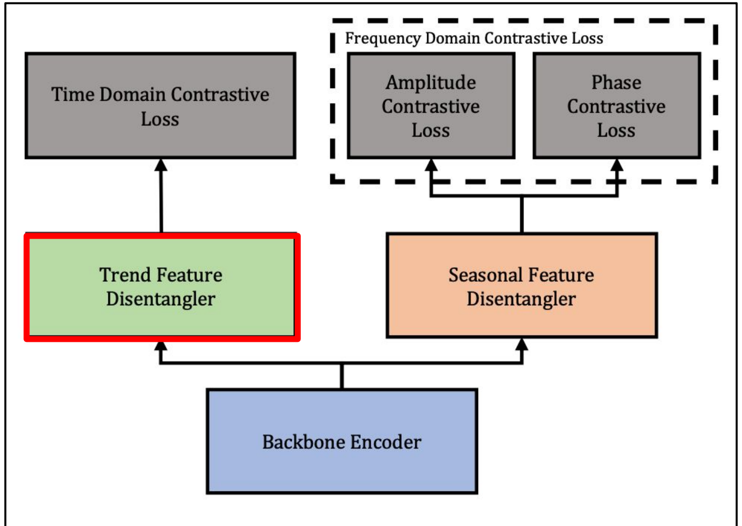
1) Trend Feature Representation

Autoregressive filtering

- able to capture **time-lagged causal relationships** from past observation
- problem : ***how to select lookback window?***

  → propose to use a **MIXUTRE of auto-regressive exports**

  (adaptively select the appropriate lookback window)



(a) Overall Framework

3. CoST

3. Trend & Seasonal Feature Representation

1) Trend Feature Representation

Autoregressive filtering

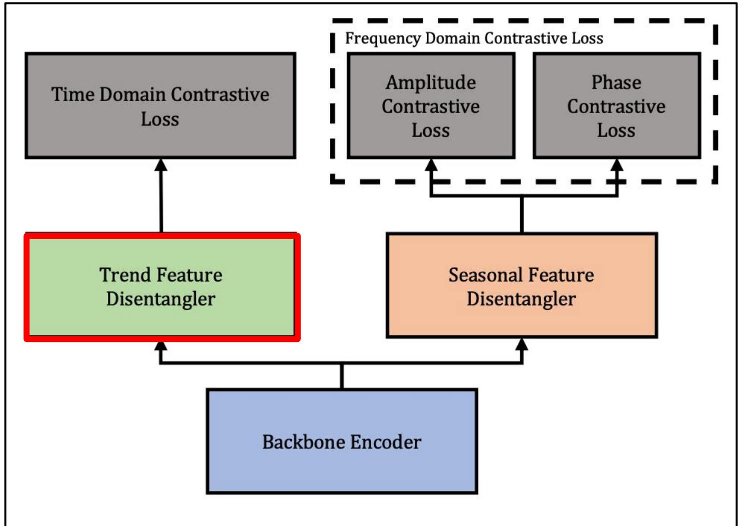
- able to capture **time-lagged causal relationships** from past observation
- problem : ***how to select lookback window?***

→ propose to use a **MIXUTRE of auto-regressive exports**

(adaptively select the appropriate lookback window)

MA of **window** size 2,4,8,16 …

= **Convolutional filter** with **kernel** size 2,4,8,16 …

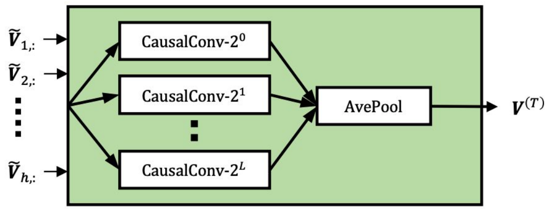


(a) Overall Framework

3. CoST

3. Trend & Seasonal Feature Representation

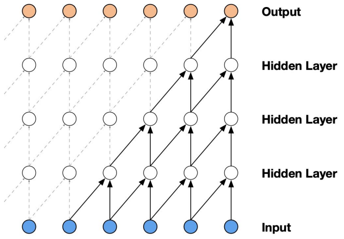
1) Trend Feature Representation

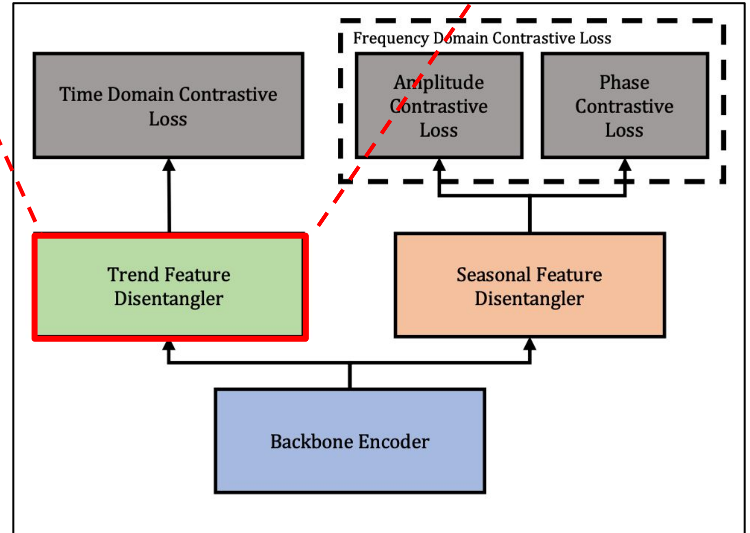


(b) Trend Feature Disentangler

Trend Feature Disentangler (TFD)

- mixture of $L+1$ autoregressive experts
- implemented as 1-d causal convolution
  - input channel : d
  - output channel : d_T
  - kernel size : 2^i
- each expert : $\tilde{V}^{(T,i)} = \text{CausalConv}\left(\tilde{V}, 2^i\right)$
- average-pooling operation :
  - $V^{(T)} = \text{AvePool}\left(\tilde{V}^{(T,0)}, \tilde{V}^{(T,1)}, \ldots, \tilde{V}^{(T,L)}\right) = \frac{1}{(L+1)} \sum_{i=0}^{L} \tilde{V}^{(T,i)}$





(a) Overall Framework

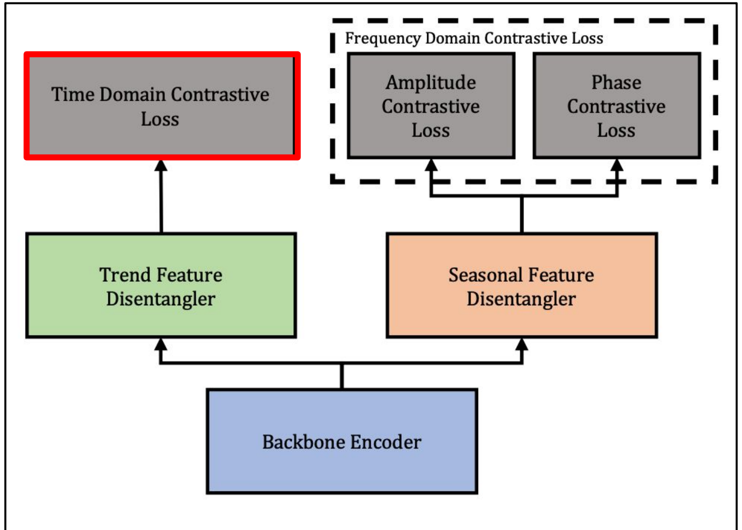
3. CoST

3. Trend & Seasonal Feature Representation

1) Trend Feature Representation

Time Domain Contrastive Loss

- employ contrastive loss in time domain
- Given N Samples & K negative samples...
  - $\mathcal{L}_{\text{time}} = \sum_{i=1}^{N} -\log \frac{\exp(\boldsymbol{q}_i \cdot \boldsymbol{k}_i / \tau)}{\exp(\boldsymbol{q}_i \cdot \boldsymbol{k}_i / \tau) + \sum_{j=1}^{K} \exp(\boldsymbol{q}_i \cdot \boldsymbol{k}_j / \tau)}$



(a) Overall Framework

3. CoST

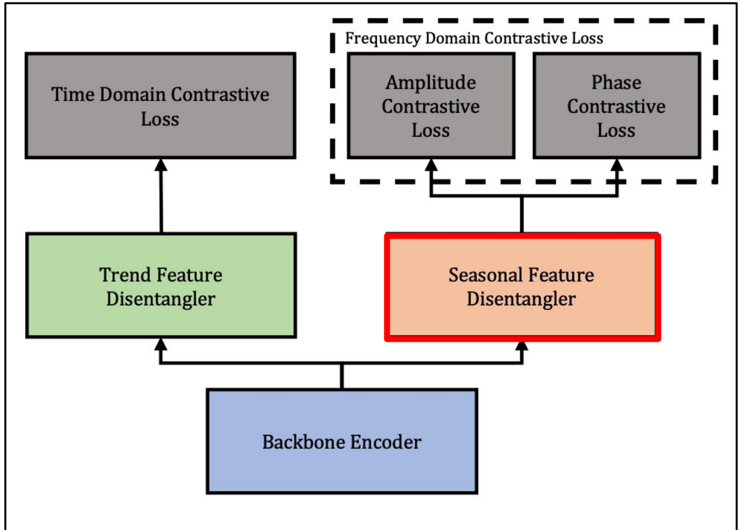
3. Trend & Seasonal Feature Representation

2) Seasonal Feature Representation

spectral analysis in frequency domain

2 issues

- (1) how to support **INTRA-frequency interactions**
- (2) what kind of learning signal is requred to learn representations, which are able to **discriminate between different seasonality patterns**



(a) Overall Framework

3. CoST

3. Trend & Seasonal Feature Representation

2) Seasonal Feature Representation

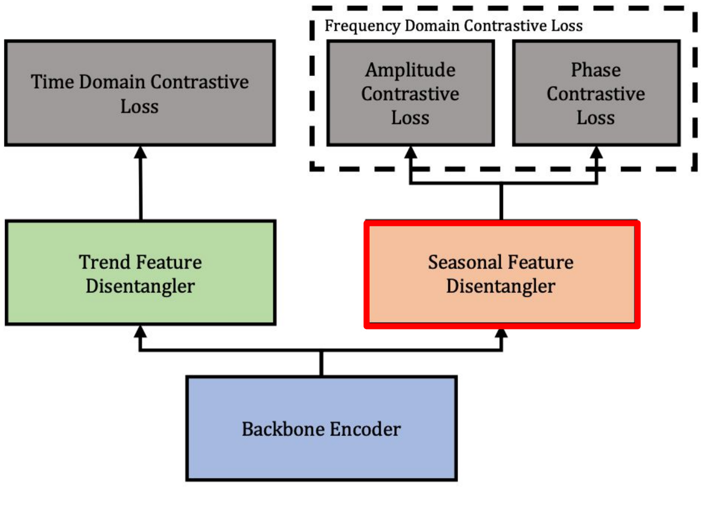
spectral analysis in frequency domain

2 issues

- (1) how to support **INTRA-frequency interactions**
- (2) what kind of learning signal is requred to learn representations, which are able to **discriminate between different seasonality patterns**

→ introduce **SFD**, which makes use of a **learnable Fourier Layer**

(SFD = Seasonal Feature Disentangler)



(a) Overall Framework

3. CoST

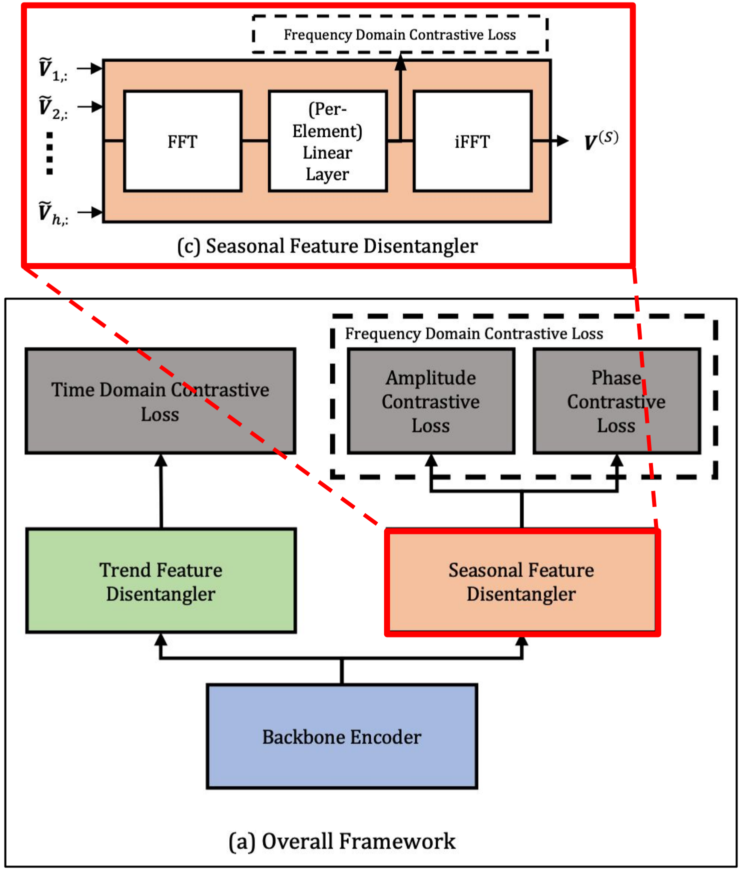
3. Trend & Seasonal Feature Representation

2) Seasonal Feature Representation

Seasonal Feature Disentangler (SFD)

composed of 2 parts

- (1) DFT (discrete Fourier Transform)
  - map **intermediate features** to **FREQUENCY** domain ($\mathcal{F}(\tilde{\boldsymbol{V}}) \in \mathbb{C}^{F \times d}$)
- (2) learnable Fourier layer
  - map in to $\boldsymbol{V}^{(S)} \in \mathbb{R}^{h \times d_S}$



3. CoST

3. Trend & Seasonal Feature Representation

2) Seasonal Feature Representation

Seasonal Feature Disentangler (SFD)

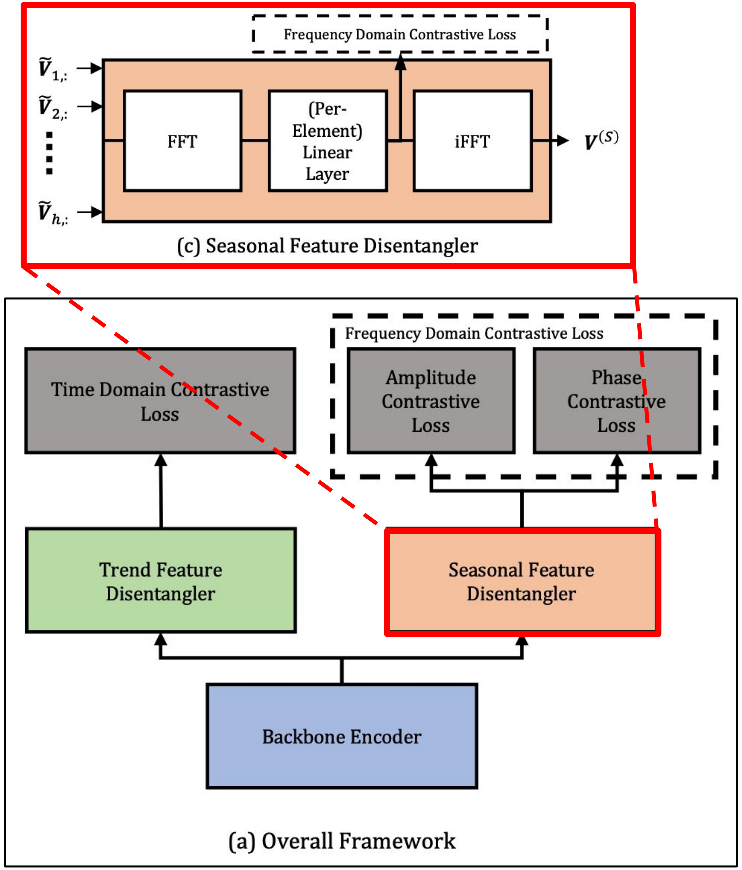
composed of 2 parts

- (1) DFT (discrete Fourier Transform)
  - map **intermediate features** to **FREQUENCY** domain ($\mathcal{F}(\tilde{\boldsymbol{V}}) \in \mathbb{C}^{F \times d}$)
- (2) learnable Fourier layer
  - map in to $\boldsymbol{V}^{(S)} \in \mathbb{R}^{h \times d_S}$

Model

iFFT (↑ $\mathcal{F}^{-1}$)   FFT (↑ $\mathcal{F}(\tilde{\boldsymbol{V}})$)

- $$V_{i,k}^{(S)} = \mathcal{F}^{-1}\left(\sum_{j=1}^{d} A_{i,j,k}\mathcal{F}(\tilde{\boldsymbol{V}})_{i,j} + B_{i,k}\right)$$



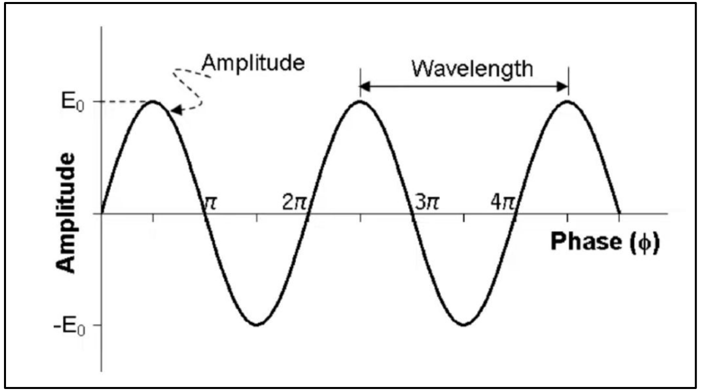
3. CoST

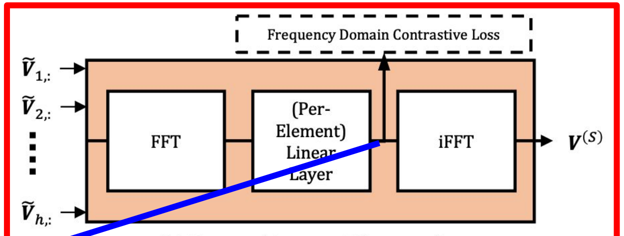
3. Trend & Seasonal Feature Representation

2) Seasonal Feature Representation

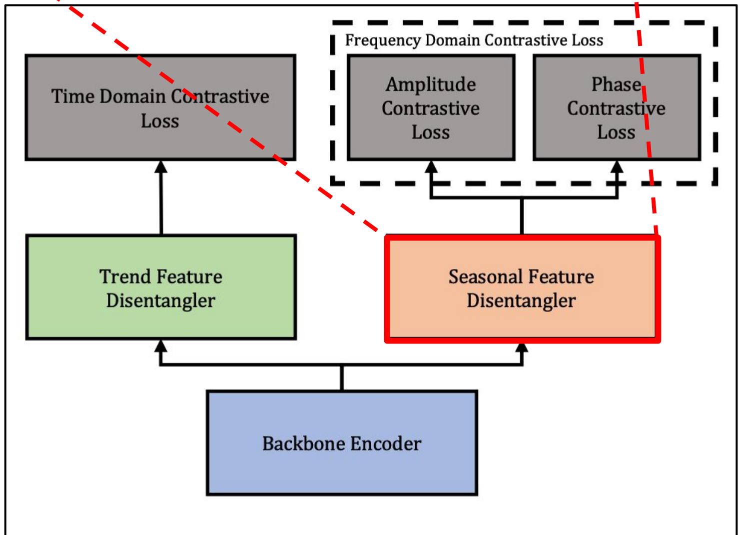
Representation for **AMPLITUDE** & **PHASE** of each frequency

$$|\boldsymbol{F}_{i,:}| \text{ and } \phi(\boldsymbol{F}_{i,:})$$





(c) Seasonal Feature Disentangler



(a) Overall Framework

3. CoST

3. Trend & Seasonal Feature Representation

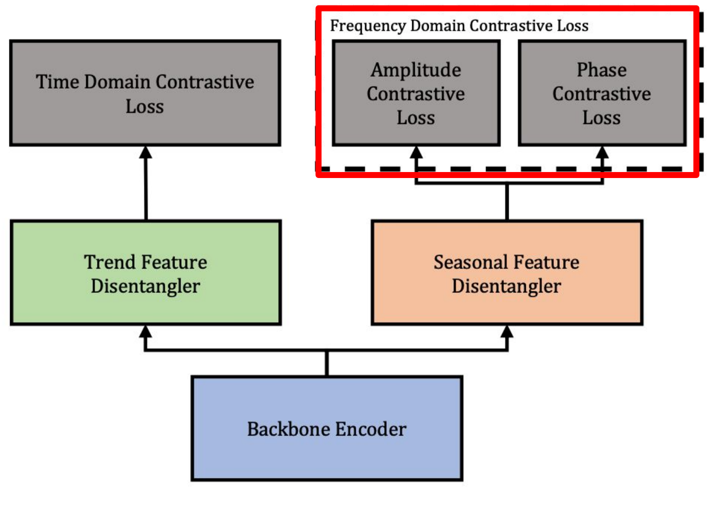
2) Seasonal Feature Representation

Frequency Domain Contrastive Loss

discriminate between **different periodic patterns**, given an frequency

- $\mathcal{L}_{\text{amp}} = \frac{1}{FN} \sum_{i=0}^{F} \sum_{j=1}^{N} -\log \frac{\exp\left(|\boldsymbol{F}_{i,:}^{(j)}| \cdot |\left(\boldsymbol{F}_{i,:}^{(j)}\right)'|\right)}{\exp\left(|\boldsymbol{F}_{i,:}^{(j)}| \cdot |\left(\boldsymbol{F}_{i,:}^{(j)}\right)'|\right) + \sum_{k \neq j}^{N} \exp\left(|\boldsymbol{F}_{i,:}^{(j)}| \cdot |\boldsymbol{F}_{i,:}^{(k)}|\right)}.$
- $\mathcal{L}_{\text{phase}} = \frac{1}{FN} \sum_{i=0}^{F} \sum_{j=1}^{N} -\log \frac{\exp\left(\phi\left(\boldsymbol{F}_{i,:}^{(j)}\right) \cdot \phi\left(\left(\boldsymbol{F}_{i,:}^{(j)}\right)'\right)\right)}{\exp\left(\phi\left(\boldsymbol{F}_{i,:}^{(j)}\right) \cdot \phi\left(\left(\boldsymbol{F}_{i,:}^{(j)}\right)'\right)\right) + \sum_{k \neq j}^{N} \exp\left(\phi\left(\boldsymbol{F}_{i,:}^{(j)}\right) \cdot \phi\left(\boldsymbol{F}_{i,:}^{(k)}\right)\right)}.$

where $\boldsymbol{F}_{i,:}^{(j)}$ is the j-th sample in a mini-batch, and $\left(\boldsymbol{F}_{i,:}^{(j)}\right)'$ is the augmented version of that sample.



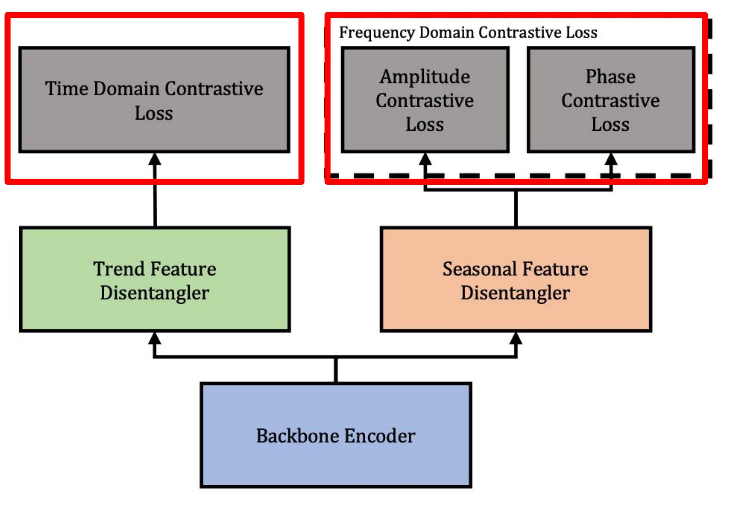
(a) Overall Framework

3. CoST

3. Trend & Seasonal Feature Representation

3) Overall Loss

$$\mathcal{L} = \mathcal{L}_{\text{time}} + \frac{\alpha}{2}(\mathcal{L}_{amp} + \mathcal{L}_{phase})$$



(a) Overall Framework

4. Conclusion

- **Time Series Decomposition** using DL module
    - different kernel sizes to obtain multiple trends
- Representation Learning in …
    - (1) **Time** Domain (for TREND)
    - (2) **Frequency** Domain (for SEASONALITY)
        - 2-1) Amplitude
        - 2-2) Phase
- Limitation : evaluation only on FORECASTING tasks

Thank You!