DCdetector:
Dual Attention Contrastive Representation Learning for Time Series Anomaly Detection

---

Division of Artificial Intelligence Engineering, SMWU
정채리 (SNSec Lab.)
26.01.23

1 Intorduction

Time Series Anomaly Detection
Overview

Introduction

>>> **Time series anomaly detection**

Challenges of time series anomaly detection

- it is still being determined what the anomalies will be like
- anomalies are usually rare, so it takes work to get labels
- models should consider temporal, multidimensional, and non-stationary features for time series data

Various time series anomaly detection methods

- supervised and semi-supervised → it is hard to label the data
- unsupervised
  - reconstruction based → it is challenging to learn a well-reconstructed model for normal data
- contrastive representative learning
  - need to be explored in the time-series anomaly detection area

Introduction

>>> Overview

Key idea

- normal time series points **have strong correlations with other points (the anomalies do not)**
- Learning **consistent representations for anomalies from different views** will be hard
- if normal and abnormal points' representations are distinguishable, we can **detect anomalies without reconstruction model**

DCdetecor

- **D**ual attention **C**ontrastive representation learning anomaly **detector**
- contrastive structure with **two branches** and a dual attention module, and two branches **share weights**
  - **Patch-wise representation**
  - **In-patch representation**
- model is **trained based on the similarity of two branches**
- utilizes **patching-based attention** networks, **multi-scale** design, **channel independence** design

2 Related Work

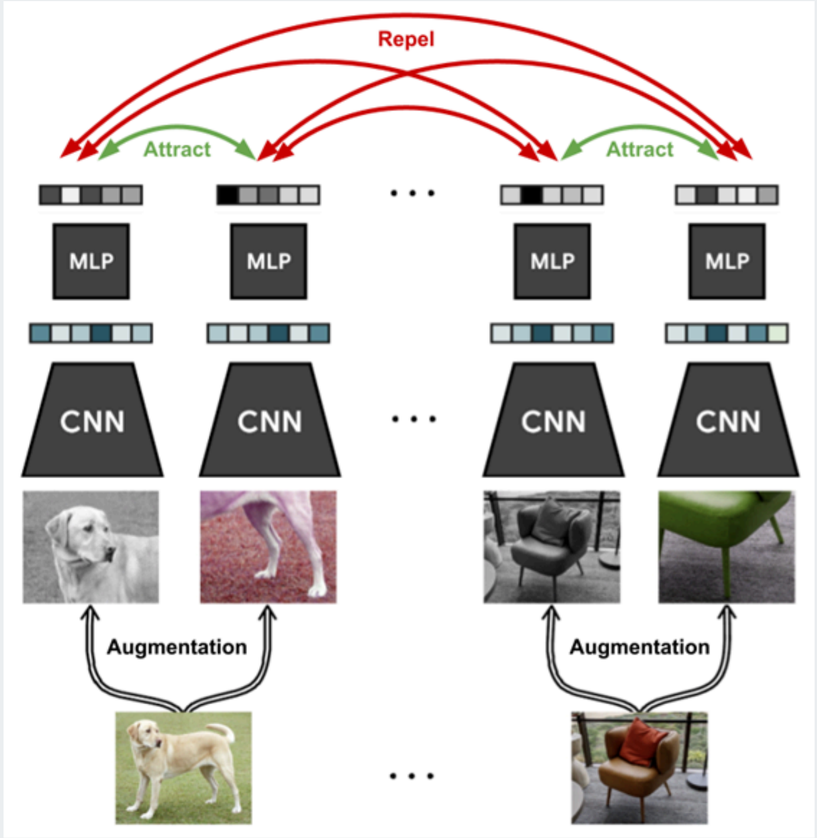
Related work

Related Work

Time Series Anomaly Detection

- Statistical method
- Machine learning method
  - clustering, density-based, classification
- Deep learning method
  - RNNs(LSTM), GANs, deep reinforcement learning

- Supervised methods
- Unsupervised methods
- reconstruction approach
  - Self-supervised learning



Contrastive Representation Learning

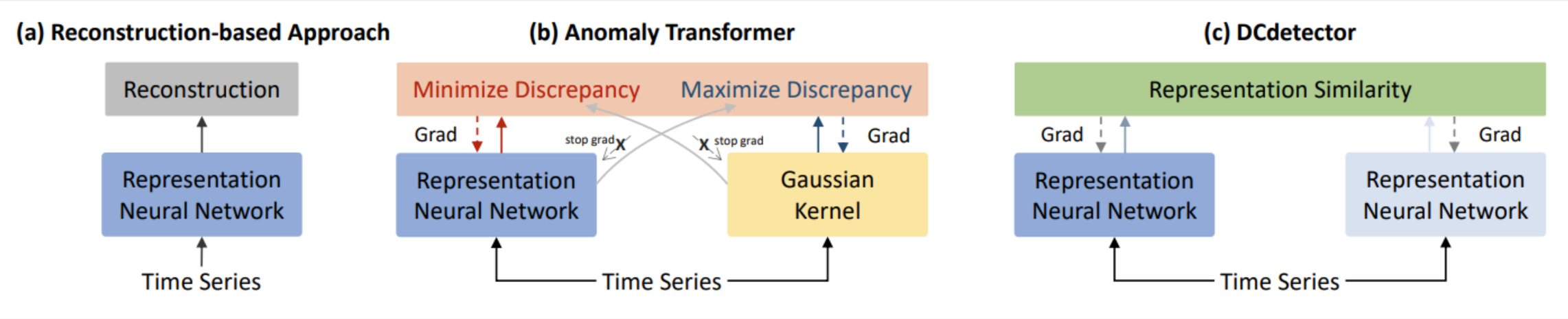
- Classical contrastive models create <positive, negative> sample pairs to learn a representation
- DCdetector is **free from negative samples**

3 Methodology

Overall Architecture
Dual Attention Contrastive Structure
Representation Discrepancy
Anomaly Criterion

Methodology

- **anomalies have less connection** or interaction with the **whole series** than their adjacent points



Methodology

Anomaly Transformer

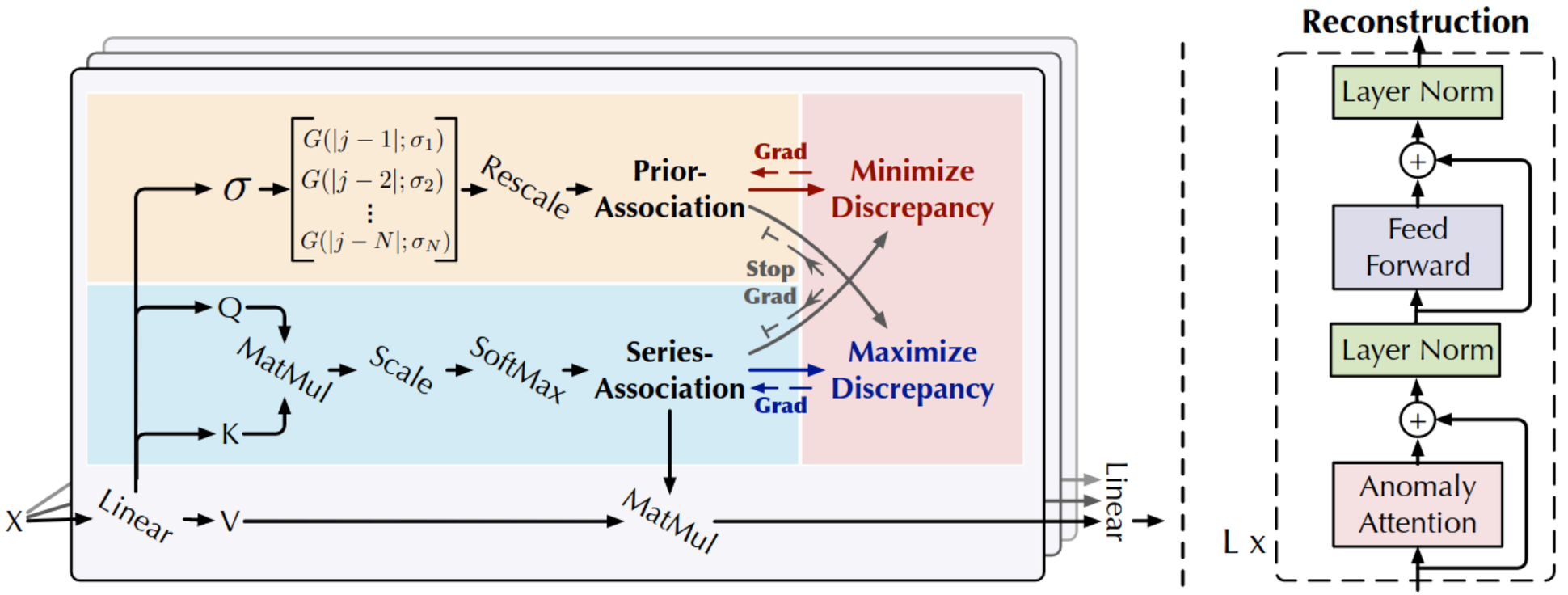


Figure 1: Anomaly Transformer architecture. Anomaly-Attention (left) models the prior-association and series-association simultaneously. In addition to the reconstruction loss, our model is also optimized by the minimax strategy with a specially-designed stop-gradient mechanism (gray arrows) to constrain the prior- and series- associations for more distinguishable association discrepancy.

Methodology

- leverages the designed **contrastive learning-based dual-branch attention** for discrepancy learning of anomalies in **different views** to enlarge the differences between anomalies and normal points

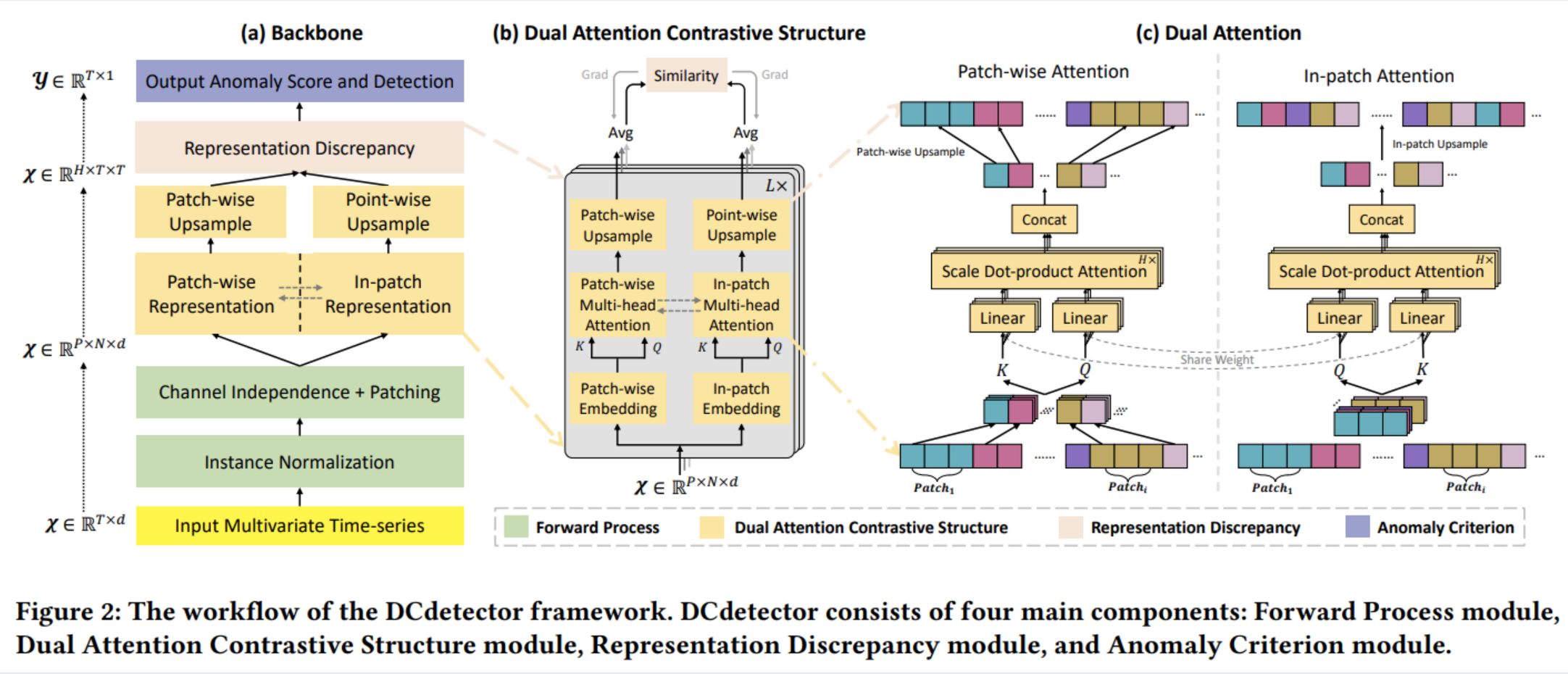
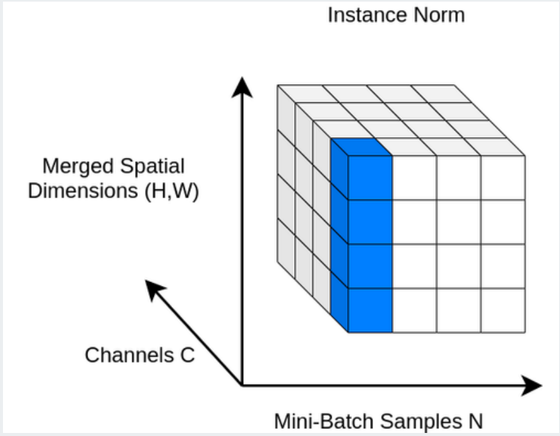


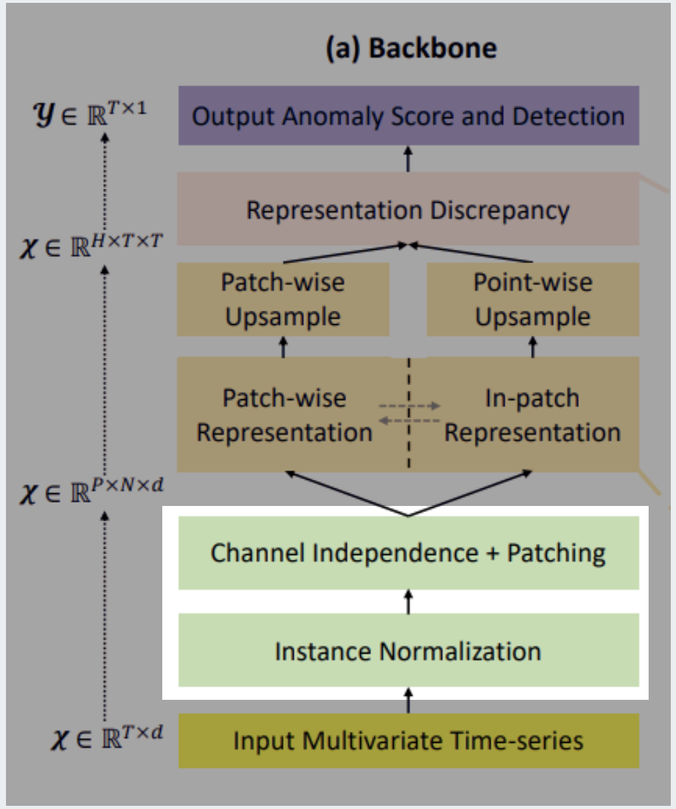
Figure 2: The workflow of the DCdetector framework. DCdetector consists of four main components: Forward Process module, Dual Attention Contrastive Structure module, Representation Discrepancy module, and Anomaly Criterion module.

Methodology

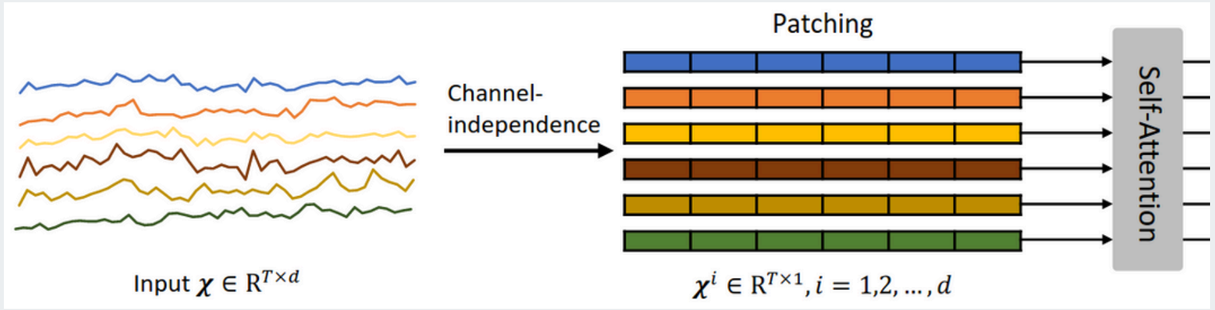
>>> Forward Process



- normalized by an **instance normalization** module
- The inputs to the instance normalization all come from the **independent channels**
- the multivariate time series input is **considered as a single time series** and **divided into patches**
  - reduce parameter numbers and overfitting issues



Forward Process

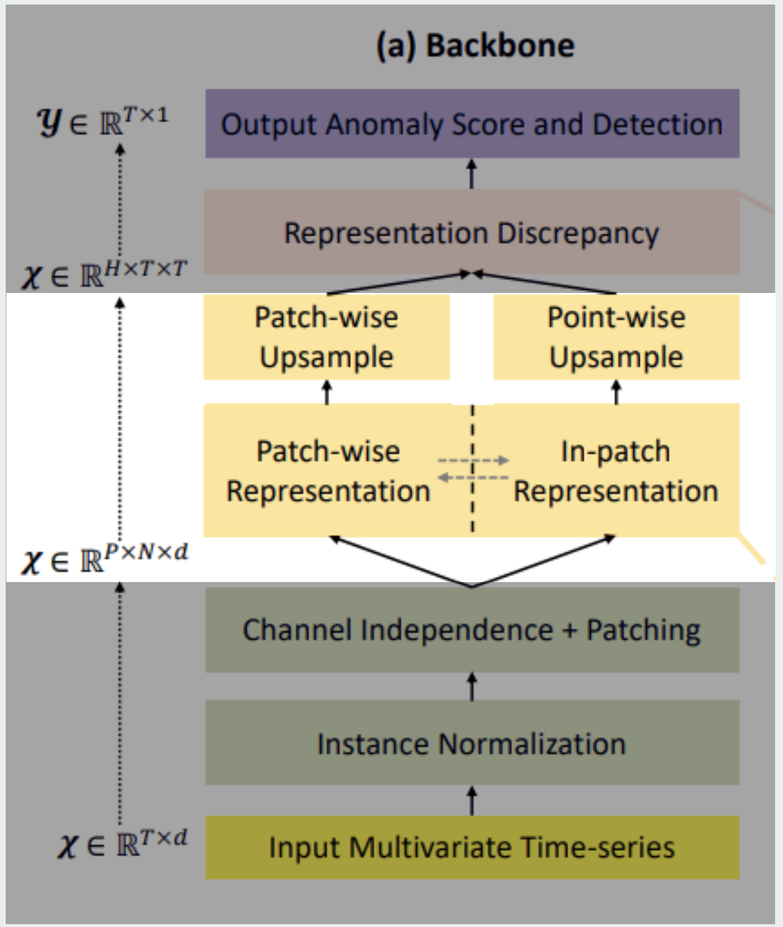


Methodology

>>> Dual Attention Contrastive Structure

- It learns the representation of inputs in **different views**
  - **Patch-wise representation**
  - **In-patch representation**
- its basic setting is similar to the contrastive methods **only using positive samples**
- **share the same self-attention network**

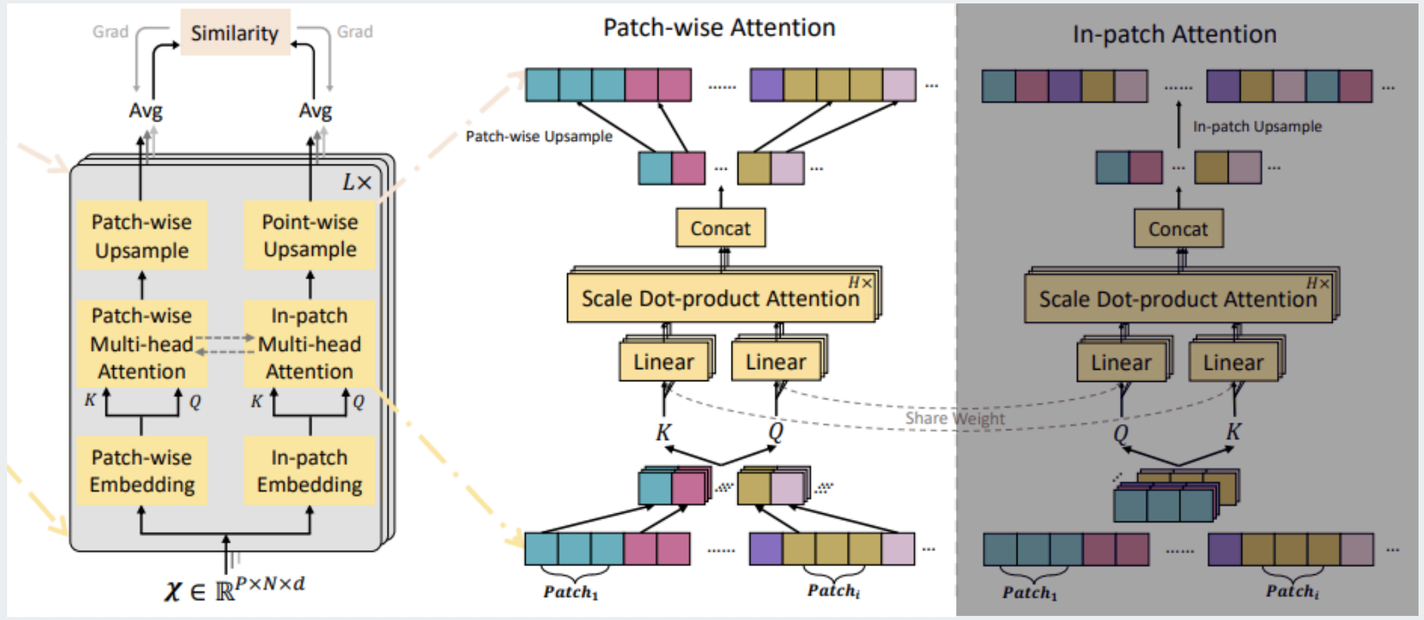
- input time series $\mathcal{X} \in \mathbb{R}^{T \times d}$ are patched as $\mathcal{X} \in \mathbb{R}^{P \times N \times d}$
- P is the size of patches and N is the number of patches
- fuse the channel information with the batch dimension
  → $\mathcal{X} \in \mathbb{R}^{P \times N}$



Methodology

>>> Dual Attention Contrastive Structure - Patch-wise

- input shape: N x P
- **a single patch is considered as a unit**
- the **dependencies among patches** are modeled by a multi-head self-attention network
- embedded operation will be applied in the patch_size (P) dimension → $\mathcal{X}_{\mathcal{N}} \in \mathbb{R}^{N \times d_{model}}$
  (패치 안에 있는 P개의 값들을 하나의 벡터 표현으로 압축)



WQi, WKi는 patch-wise와 in-patch에서 동일

$$\mathcal{Q}_{\mathcal{N}i}, \mathcal{K}_{\mathcal{N}i} = \mathcal{W}_{\mathcal{Q}i}\mathcal{X}_{\mathcal{N}i}, \mathcal{W}_{\mathcal{K}i}\mathcal{X}_{\mathcal{N}i} \quad 1 \le i \le H,$$

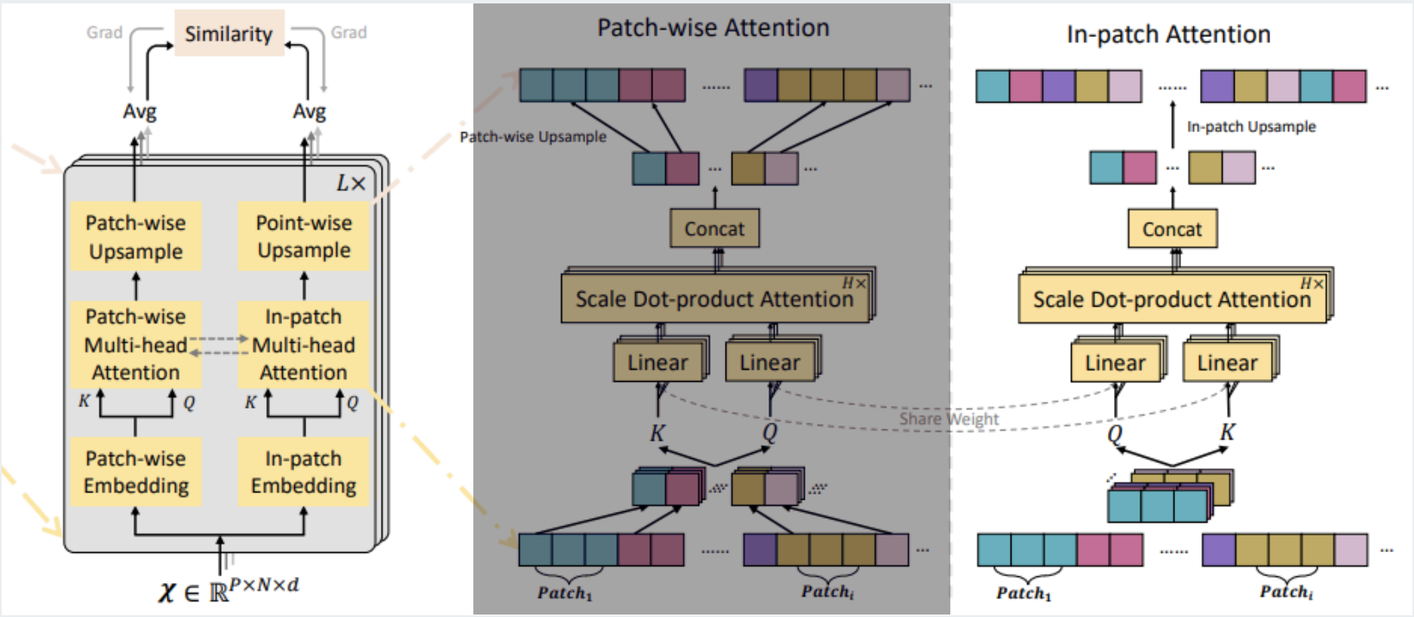
$$Attn_{\mathcal{N}_i} = Softmax\left(\frac{\mathcal{Q}_{\mathcal{N}i}\mathcal{K}_{\mathcal{N}i}^{T}}{\sqrt{\frac{d_{model}}{H}}}\right),$$

$$Attn_{\mathcal{N}} = \text{Concat}(Attn_{\mathcal{N}_1}, \cdots, Attn_{\mathcal{N}_H})W_{\mathcal{N}}^{O},$$

Methodology

>>> Dual Attention Contrastive Structure - In-patch

- input shape: P x N
- the **dependencies of points in the same patch** are gained by a multi-head self-attention network
- embedded operation will be applied in the patch_number (N) dimension → $\mathcal{X}_{\mathcal{P}} \in \mathbb{R}^{P \times d_{model}}$
  (이 위치의 포인트가 여러 패치에 걸쳐 어떻게 나타나는지를 요약)



WQi, WKi는 patch-wise와 in-patch에서 동일

$$Q_{\mathcal{P}i}, \mathcal{K}_{\mathcal{P}i} = \mathcal{W}_{Qi}\mathcal{X}_{\mathcal{P}i}, \mathcal{W}_{\mathcal{K}i}\mathcal{X}_{\mathcal{P}i} \quad 1 \le i \le H,$$

$$Attn_{\mathcal{P}_i} = Softmax\left(\frac{Q_{\mathcal{P}i}\mathcal{K}_{\mathcal{P}i}^{T}}{\sqrt{\frac{d_{model}}{H}}}\right),$$

$$Attn_{\mathcal{P}} = \text{Concat}(Attn_{\mathcal{P}_1}, \cdots, Attn_{\mathcal{P}_H})W_{\mathcal{P}}^{O},$$

Methodology

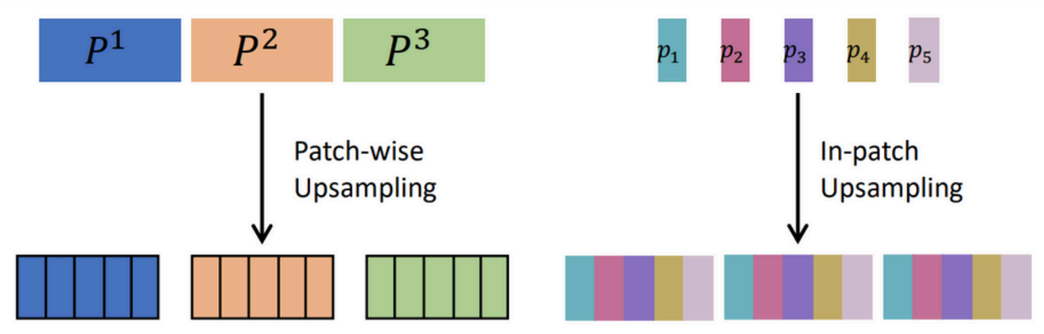
>>> Dual Attention Contrastive Structure - Up-sampling

Patch-wise

- patch-wise attention ignores the relevance among points in a patch
- For the patch-wise branch, repeating is done inside patches for up-sampling
  (패치 단위 결과를 패치 안의 P개 포인트로 반복)

In-patch

- in-patch attention ignores the relevance among patches
- For the in-patch branch, repeating is done from "one" patch to a full number of patches
  (포인트 관계 결과를 N개의 패치에 반복)



Methodology

>>> Dual Attention Contrastive Structure

Multi-scale design

- patching and repeating up-sampling operations inevitably lead to information loss
- The final representation concatenates results in **different scales (i.e., patch sizes)**

patch-wise in-patch

$$\mathcal{N} = \text{Upsampling}(Attn_{\mathcal{N}}), \quad \mathcal{P} = \text{Upsampling}(Attn_{\mathcal{P}}).$$

Contrastive Structure

- Patch-wise and in-patch branches output representations of the **same input time series in two different views**
- normal points can maintain their representation under permutations while the anomalies can not
  → learn **permutation invariant representation**

Methodology

>>> Representation Discrepancy

Loss Function

- formalize a loss function based on **KL divergence** to measure the similarity of such two representations
- Stop-gradient (labeled as 'Stopgrad') operation is also used in loss function

in-patch patch-wise

$$\mathcal{L}_{\mathcal{P}}\{\mathcal{P},\mathcal{N};\mathcal{X}\} = \sum KL(\mathcal{P}, \text{Stopgrad}(\mathcal{N})) + KL(\text{Stopgrad}(\mathcal{N}), \mathcal{P}), \tag{8}$$

$$\mathcal{L}_{\mathcal{N}}\{\mathcal{P},\mathcal{N};\mathcal{X}\} = \sum KL(\mathcal{N}, \text{Stopgrad}(\mathcal{P})) + KL(\text{Stopgrad}(\mathcal{P}), \mathcal{N}), \tag{9}$$

Total Loss

$$\mathcal{L} = \frac{\mathcal{L}_{\mathcal{N}} - \mathcal{L}_{\mathcal{P}}}{len(\mathcal{N})}.$$

Model Collapse

- with only single-type inputs, DCdetector does **not fall into a trivial solution (model collapse)**
- DCdetector still **works without stop gradient** operation

  → two branches are totally asymmetric

Methodology

>>> Anomaly Criterion

- the **distances of representation** results from different views for **normal points are less** than anomalies
- It is a point-wise anomaly score, and **anomalies result in higher scores** than normal points

$$\text{AnomalyScore}(\mathcal{X}) = \sum KL(\mathcal{P}, \text{Stopgrad}(\mathcal{N})) + KL(\mathcal{N}, \text{Stopgrad}(\mathcal{P})).$$

$$\mathcal{Y}_i = \begin{cases} 1\text{: anomaly} & \text{AnomalyScore}(\mathcal{X}_i) \geq \delta \\ 0\text{: normal} & \text{AnomalyScore}(\mathcal{X}_i) < \delta. \end{cases}$$

4 Experiment

Benchmark Datasets
Baselines and Evaluation Criteria
Implementation Details
Main Results
Model Analysis

Experiment

>>> Benchmark Datasets

8 benchmark dataset

- SMD(Server Machine Dataset, Su et al. (2019))
- PSM (Pooled Server Metrics, Abdulaal et al. (2021))
- MSL (Mars Science Laboratory rover)
- SMAP (Soil Moisture Active Passive satellite)
- SWaT (Secure Water Treatment, Mathur & Tippenhauer (2016))

- NIPS-TS-SWAN
- NIPS-TS-GECCO
- UCR (univariate)

Benchmark	Source	Dimension	Window	Patch Size	#Training	#Test (Labeled)	AR (%)
MSL	NASA Space Sensors	55	90	[3,5]	58,317	73,729	10.5
SMAP	NASA Space Sensors	25	105	[3,5,7]	135,183	427,617	12.8
PSM	eBay Server Machine	25	60	[1,3,5]	132,481	87,841	27.8
SMD	Internet Server Machine	38	105	[5,7]	708,405	708,420	4.2
SWaT	Infrastructure System	51	105	[3,5,7]	495,000	449,919	12.1
NIPS-TS-SWAN	Space (Solar) Weather	38	36	[1,3]	60,000	60,000	32.6
NIPS-TS-GECCO	Water Quality for IoT	9	90	[1,3,5]	69,260	69,261	1.1
UCR	Various Natural Sources	1	105	[3,5,7]	2,238,349	6,143,541	0.6

Experiment

>>> Baselines

26 baseline

- Reconstruction-based
  - AutoEncoder, LSTM-VAE, OmniAnomaly, BeatGan, InterFusion, **Anomaly Transformer**
- Autoregression-based
  - VAR, Autoregression, LSTN-RNN, LSTM, CL-MPPCA
- Density-estimation
  - LOF, MPPCACD, DACMM
- Clustering-based
  - Deep-SVDD, THOC, ITAD
- Classic Methods
  - OCSVM, OCSVM-based subsequence clustering, IForest, IForest-based subsequence clustering, Gradient boosting regression
- Change point detection and time series segmentation
  - BOCPD, U-Time, TS-CP2

Experiment

>>> Evaluation Criteria

- various evaluation criteria, including the commonly-used evaluation measures: **accuracy, precision, recall, F1-score**
- **affiliation precision/recall**
  - the distance between ground truth and prediction events
- **Volume under the surface (VUS)**
  - takes anomaly events into consideration based on ROC curve

Experiment

>>> Implementation details

- Encoder Layer = 3
- channel number of hidden states = 256
- Number of head = 1
- threshold $\delta = 1$
- various patch size and window size
- Optimizer = Adam(lr = 1e-4)
- Epoch = 3 for all datasets
- Batch = 128

Experiment

Main Results

Dataset	SMD			MSL			SMAP			SWaT			PSM		
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LOF	56.34	39.86	46.68	47.72	85.25	61.18	58.93	56.33	57.60	72.15	65.43	68.62	57.89	90.49	70.61
OCSVM	44.34	76.72	56.19	59.78	86.87	70.82	53.85	59.07	56.34	45.39	49.22	47.23	62.75	80.89	70.67
U-Time	65.95	74.75	70.07	57.20	71.66	63.62	49.71	56.18	52.75	46.20	87.94	60.58	82.85	79.34	81.06
IForest	42.31	73.29	53.64	53.94	86.54	66.45	52.39	59.07	55.53	49.29	44.95	47.02	76.09	92.45	83.48
DAGMM	67.30	49.89	57.30	89.60	63.93	74.62	86.45	56.73	68.51	89.92	57.84	70.40	93.49	70.03	80.08
ITAD	86.22	73.71	79.48	69.44	84.09	76.07	82.42	66.89	73.85	63.13	52.08	57.08	72.80	64.02	68.13
VAR	78.35	70.26	74.08	74.68	81.42	77.90	81.38	53.88	64.83	81.59	60.29	69.34	90.71	83.82	87.13
MMPCACD	71.20	79.28	75.02	81.42	61.31	69.95	88.61	75.84	81.73	82.52	68.29	74.73	76.26	78.35	77.29
CL-MPPCA	82.36	76.07	79.09	73.71	88.54	80.44	86.13	63.16	72.88	76.78	81.50	79.07	56.02	**99.93**	71.80
TS-CP2	87.42	66.25	75.38	86.45	68.48	76.42	87.65	83.18	85.36	81.23	74.10	77.50	82.67	78.16	80.35
Deep-SVDD	78.54	79.67	79.10	91.92	76.63	83.58	89.93	56.02	69.04	80.42	84.45	82.39	95.41	86.49	90.73
BOCPD	70.9	82.04	76.07	80.32	87.20	83.62	84.65	85.85	85.24	89.46	70.75	79.01	80.22	75.33	77.70
LSTM-VAE	75.76	90.08	82.30	85.49	79.94	82.62	92.20	67.75	78.10	76.00	89.50	82.20	73.62	89.92	80.96
BeatGAN	72.90	84.09	78.10	89.75	85.42	87.53	92.38	55.85	69.61	64.01	87.46	73.92	90.30	93.84	92.04
LSTM	78.55	85.28	81.78	85.45	82.50	83.95	89.41	78.13	83.39	86.15	83.27	84.69	76.93	89.64	82.80
OmniAnomaly	83.68	86.82	85.22	89.02	86.37	87.67	92.49	81.99	86.92	81.42	84.30	82.83	88.39	74.46	80.83
InterFusion	87.02	85.43	86.22	81.28	92.70	86.62	89.77	88.52	89.14	80.59	85.58	83.01	83.61	83.45	83.52
THOC	79.76	90.95	84.99	88.45	90.97	89.69	92.06	89.34	90.68	83.94	86.36	85.13	88.14	90.99	89.54
AnomalyTrans	**88.47**	**92.28**	**90.33**	91.92	96.03	93.93	93.59	**99.41**	96.41	89.10	99.28	94.22	96.94	97.81	97.37
DCdetector	83.59	91.10	87.18	**93.69**	**99.69**	**96.60**	**95.63**	98.92	**97.02**	**93.11**	**99.77**	**96.33**	**97.14**	98.74	**97.94**

Dataset	Method	Acc	F1	Aff-P [31]	Aff-R [31]	R_A_R [52]	R_A_P [52]	V_ROC [52]	V_PR [52]
MSL	AnomalyTrans	98.69	93.93	51.76	95.98	90.04	87.87	88.20	86.26
	DCdetector	**99.06**	**96.60**	**51.84**	**97.39**	**93.17**	**91.64**	**93.15**	**91.66**
SMAP	AnomalyTrans	99.05	96.41	51.39	**98.68**	**96.32**	94.07	**95.52**	93.37
	DCdetector	**99.21**	**97.02**	**51.46**	98.64	96.03	**94.18**	95.19	**93.46**
SWaT	AnomalyTrans	98.51	94.22	**53.03**	**98.08**	**97.89**	93.47	**97.92**	93.49
	DCdetector	**99.09**	**96.33**	52.40	97.67	96.63	**94.06**	96.95	**94.34**
PSM	AnomalyTrans	98.68	97.37	**55.35**	80.28	**91.83**	**93.03**	**88.71**	**90.71**
	DCdetector	**98.95**	**97.94**	54.71	**82.93**	91.55	92.93	88.41	90.58

Experiment

Main Results

Dataset	NIPS-TS-GECCO			NIPS-TS-SWAN		
Metric	P	R	F1	P	R	F1
OCSVM*	2.1	34.1	4.0	19.3	0.1	0.1
MatrixProfile	4.6	18.5	7.4	16.7	17.5	17.1
GBRT	17.5	14.0	15.6	44.7	37.5	40.8
LSTM-RNN	34.3	27.5	30.5	52.7	22.1	31.2
Autoregression	39.2	31.4	34.9	42.1	35.4	38.5
OCSVM	18.5	**74.3**	29.6	47.4	49.8	48.5
IForest*	39.2	31.5	39.0	40.6	42.5	41.6
AutoEncoder	42.4	34.0	37.7	49.7	52.2	50.9
AnomalyTrans	25.7	28.5	27.0	90.7	47.4	62.3
IForest	**43.9**	35.3	39.1	56.9	**59.8**	58.3
DCdetector	38.3	59.7	**46.6**	**95.5**	59.6	**73.4**

Dataset	UCR				
Metric	Acc	P	R	F1	Count
AnomalyTrans	99.49	60.41	**100**	73.08	42
DCdetector	**99.51**	**61.62**	**100**	**74.05**	**46**

Experiment

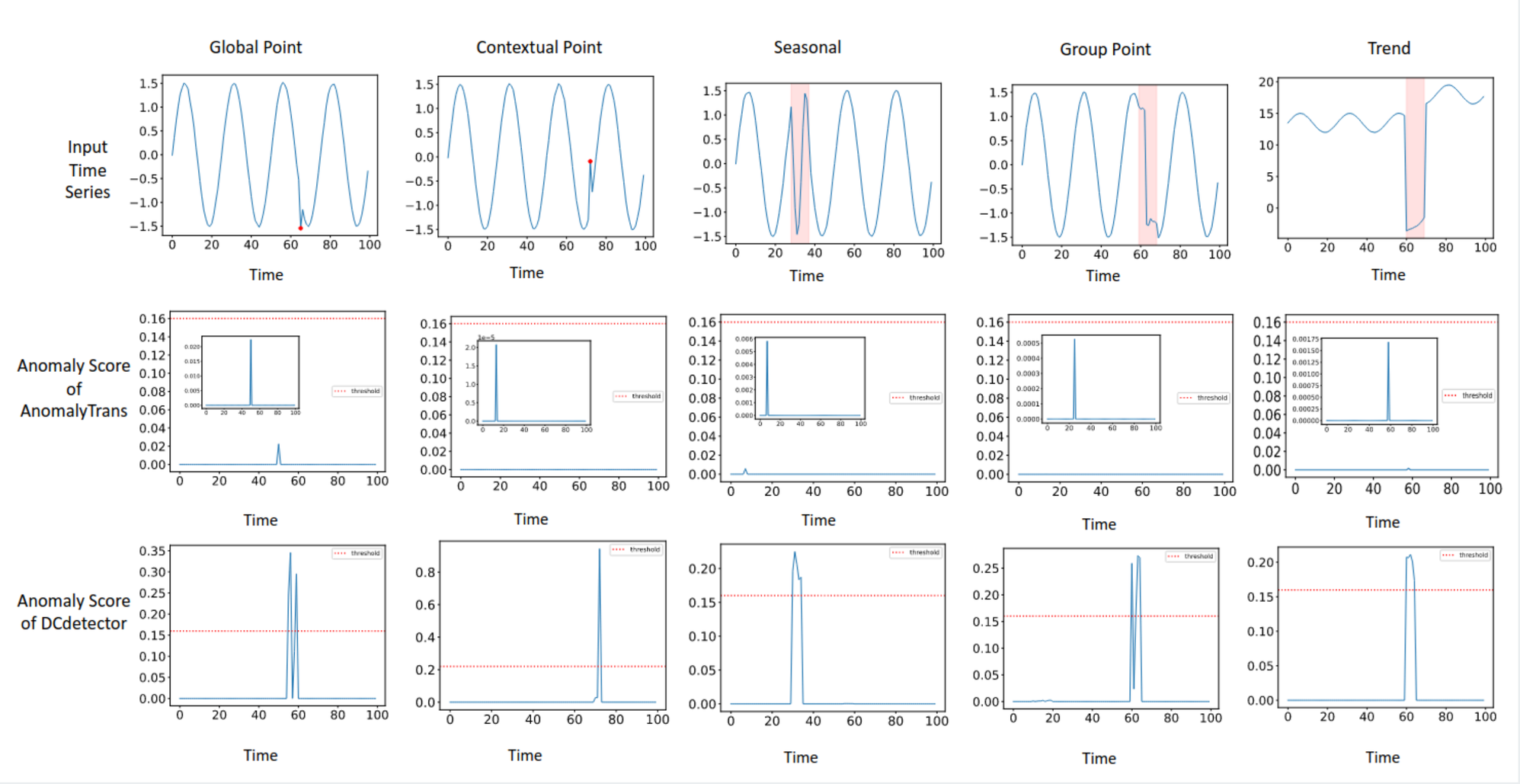
>>> Model Analysis - ablation studies

Stop Gradient		MSL			SMAP			PSM		
Patch-wise Branch	In-patch Branch	P	R	F1	P	R	F1	P	R	F1
✘	✘	91.99	89.98	90.97	94.49	96.56	95.51	96.86	97.51	97.18
✔	✘	91.27	72.61	80.88	94.46	93.17	93.81	97.15	98.51	97.83
✘	✔	92.18	96.27	94.18	94.37	98.19	96.24	96.98	98.04	97.51
✔	✔	93.69	99.69	**96.60**	95.63	98.92	**97.02**	97.14	98.74	**97.94**

Forward Process		MSL			SMAP			PSM		
Bilateral Filter	Instance Norm	P	R	F1	P	R	F1	P	R	F1
✘	✘	92.58	96.68	94.59	94.65	97.38	96.00	97.01	97.79	97.40
✔	✘	92.64	98.74	95.59	94.48	98.48	96.44	97.11	98.44	97.77
✘	✔	93.69	99.69	**96.60**	95.63	98.92	**97.02**	97.14	98.74	**97.94**
✔	✔	92.28	98.82	95.44	95.11	97.06	96.08	96.88	97.82	97.35

Experiment

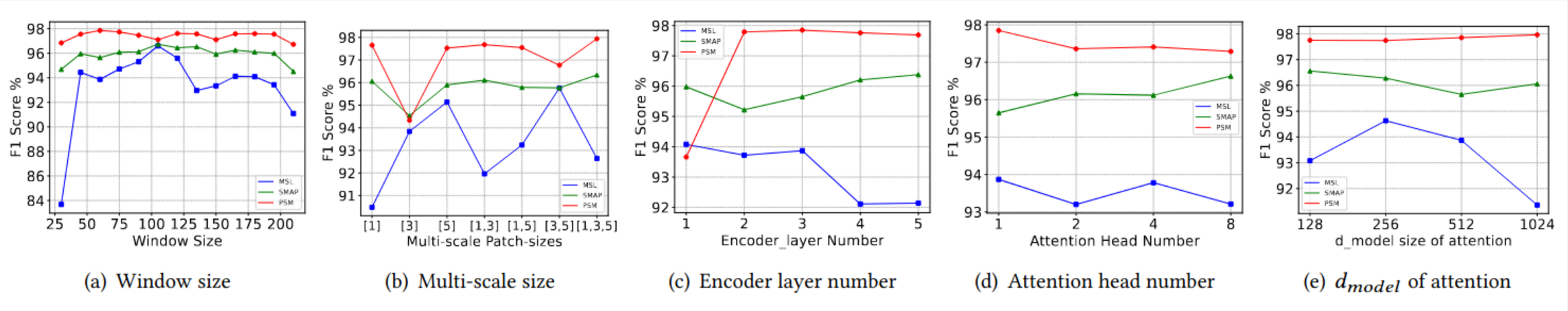
Model Analysis - visual analysis



Experiment

>>> Model Analysis - parameter sensitivity

- robust with a **wide range of window sizes** (from 30 to 210)
- the **multi-scale design** contributes to the final performance, and different patch-size combinations lead to different performances
- DCdetector achieves the best performance with a **small attention head number and d_model size**

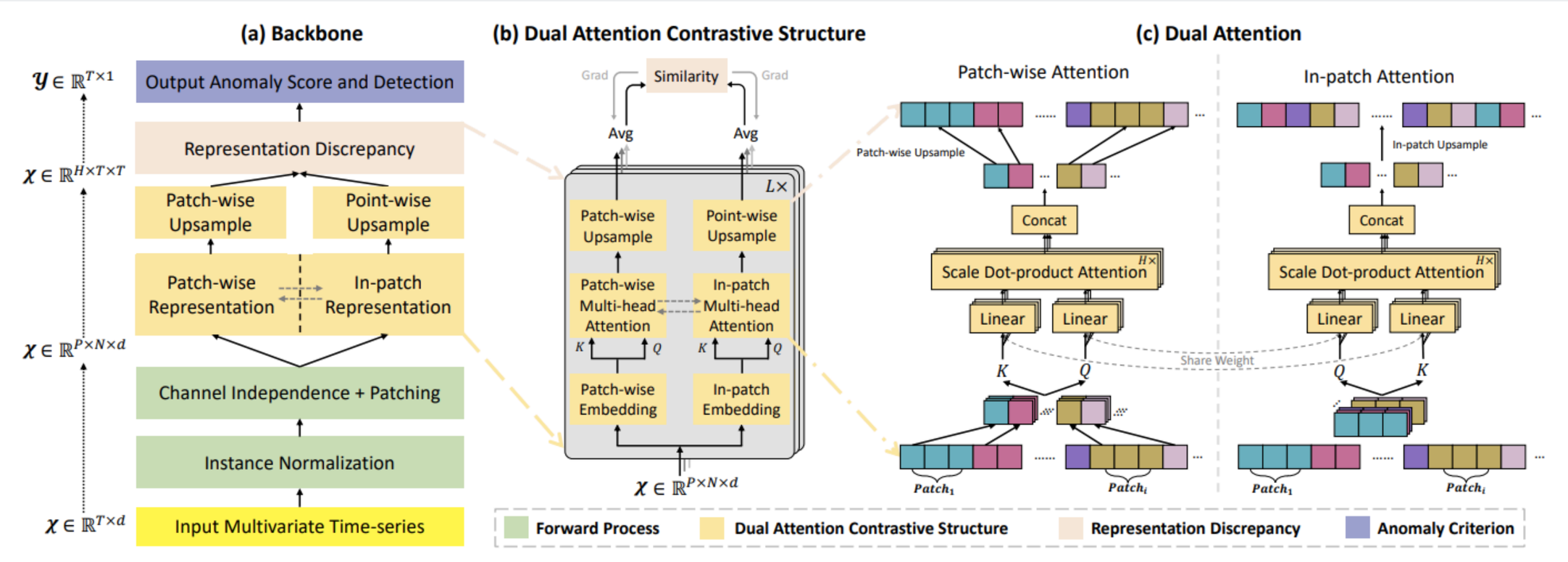


(a) Window size (b) Multi-scale size (c) Encoder layer number (d) Attention head number (e) d_{model} of attention

5 Conclusion

Conclusion

Conclusion



- **contrastive learning-based dual-branch attention structure** → permutation invariant representation
- **multiscale** and **channel independence patching**
- **pure contrastive loss function** without reconstruction error

THANK YOU

Division of Artificial Intelligence Engineering, SMWU
정채리 (SNSec Lab.)
26.01.23