

2025 KAMP 제조 데이터 분석 경진대회

# 불완전 데이터 수집 시나리오에 강건한 적응형 다변량 시계열 이상탐지 알고리즘

2025.11.19

브로콜리단

1



# CONTENTS

01 배경

04 실험 설정

02 문제 정의

05 실험 결과

03 알고리즘 소개

06 결론

2

## Background

- 소성가공 공정

- 소성가공은 금속 재료에 외력을 가하여 소성 변형을 일으켜 원하는 형상으로 제조하는 공정을 의미

- 과제 정의

- 스틸 튜브(Steel Tube) 압출 공정에서 수집된 센서 기반 다변량 시계열 데이터 분석을 통한 고정밀 품질 관리 및 불량 검출을 목적으로 함



# 01

## Background



- 브로콜리단이 분석한 실제 제조 현장에서의 Pain Point



① 제조 데이터의 **불완전성**



② 실제 데이터 대상 **낮은 탐지 성능**



③ 중소 제조 현장의 **높은 AI 도입 장벽**

❖ 데이터 불완전성: 데이터가 수집·저장 되는 과정에서 센서 고장, 통신 불량, 정전, 노이즈 등을 이유로 데이터 일부가 결측, 왜곡되어 '완전한 상태로 존재하지 않는 것'

# 01

## Background



- 브로콜리단이 분석한 실제 제조 현장에서의 Pain Point



OK

### ① 제조 데이터의 불완전성

✓ 불완전한 데이터에도 그대로 적용 가능한 이상감지 알고리즘의 필요



### ② 실제 데이터 대상 낮은 탐지 성능

✓ 연구용 외 실제 데이터에도 우수한 성능을 보이는 방법 필요



### ③ 중소 제조 현장의 높은 AI 도입 장벽

✓ 과도한 전처리를 필요로 하지 않는 방법 필요

❖ 데이터 불완전성: 데이터가 수집·저장 되는 과정에서 센서 고장, 통신 불량, 정전, 노이즈 등을 이유로 데이터 일부가 결측, 왜곡되어 '완전한 상태로 존재하지 않는 것'

# 01

## Background

- 브로콜리단의 CATCH-RG(CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator)



Our Contribution is...

1. 불완전한 데이터에도 바로 활용 가능한 End-to-End 적응형 다변량 시계열 알고리즘 제안
2. Reliability-Gated Channel Mask Generator를 비롯한 모듈 구성으로 결측/노이즈에 강건하도록 고안
3. 어떠한 전처리 과정도 사용하지 않고, 지정한 6개 시나리오 전반에서 매우 우수한 성능 달성
4. 기존의 Irregular Time Series Analysis 연구들과 달리, 결측치 Masking 및 학습과정이 필요 없음

# 02 문제 정의

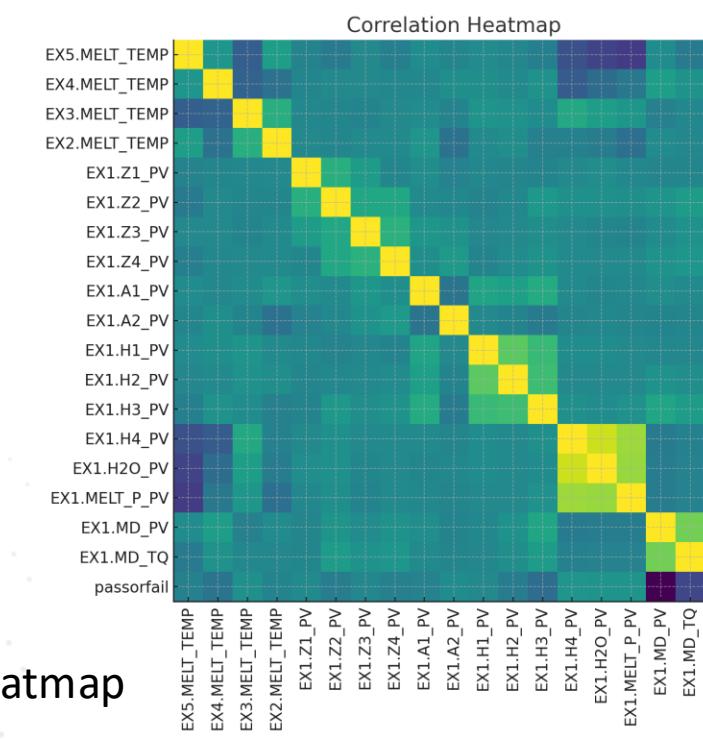
## • 데이터 분석 : 소성가공 품질 보증 데이터셋

– # of rows, columns: (17280, 20) /. Sampling rate: 0.2hz (5초당 1개)

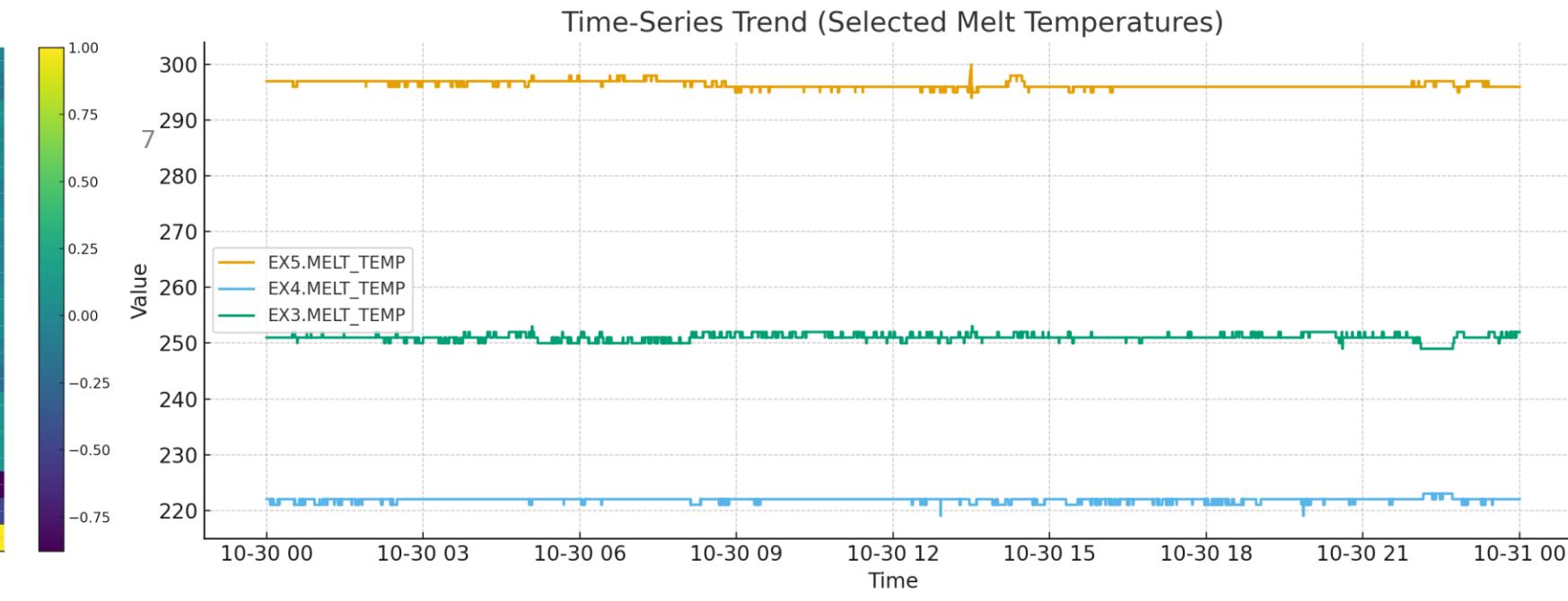
### – Variable Description

- 수지온도(MELT\_TEMP), 수지압력(EX1.MELT\_P\_PV), 모터속도(EX1.MD\_PV), 모터부하(EX1.MD\_TQ), 챔버 물온도(CHAMBER\_WATER\_TEMP) 등 공정 제어 연속값
- 품질 판정 라벨(passorfail), 타임스탬프(date)로 구성

### – 매우 고품질의 데이터 셋임을 확인(결측치 없음, 변수 별 분포 안정도 높음)



[Fig 1] Correlation Heatmap

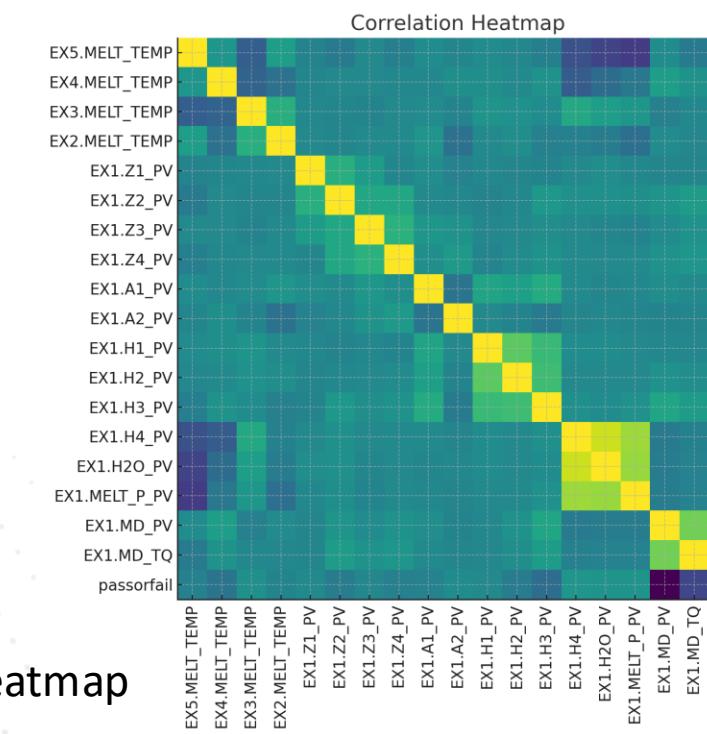


[Fig 2] Visualization of variables

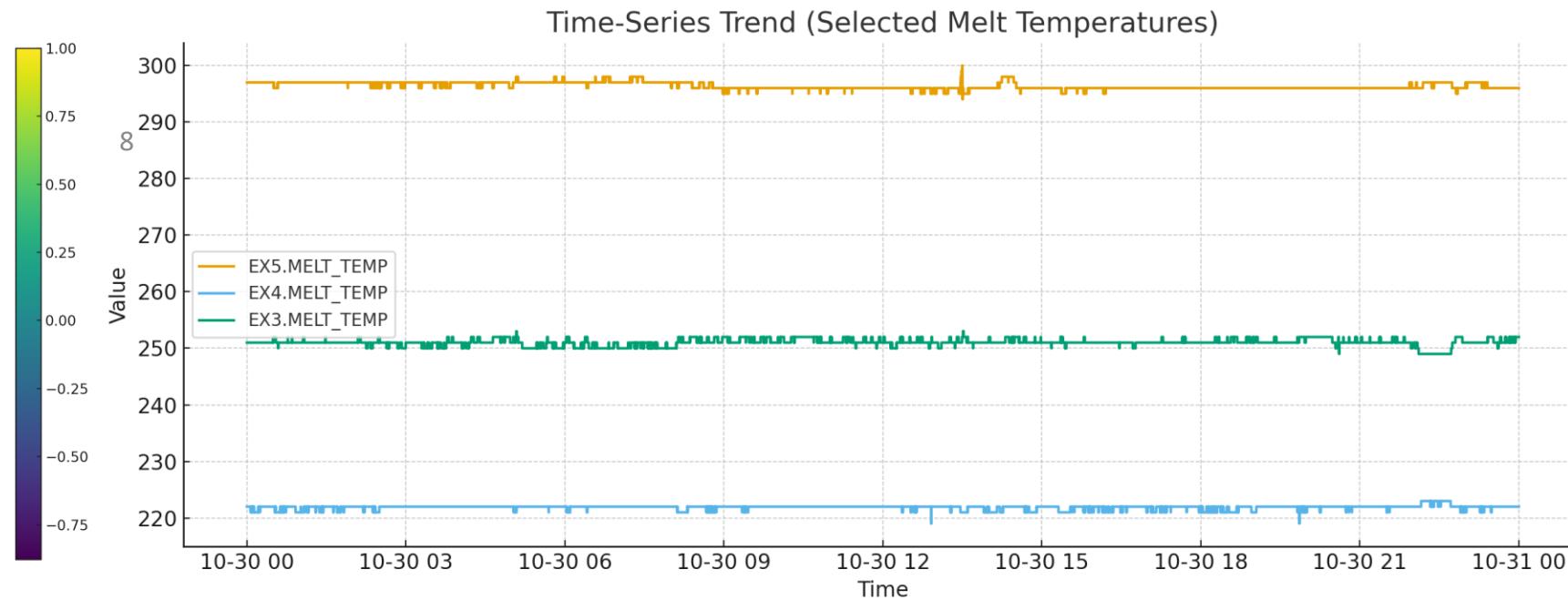
# 02 문제 정의

## • 데이터 분석 : 소성가공 품질 보증 데이터셋

- 원본 데이터셋은 결측치가 존재하지 않고 깔끔히 정제된 데이터 셋(Complete Dataset)임 확인
  - ❖ 실험·연구 환경에서 활용하기에 이상적인 데이터
- 실제 제조 현장에서 빈번히 발생하는 **데이터 불완전성**을 반영하지 못하는 한계가 있음
  - ❖ 위 데이터 그대로에서 높은 성능을 보이는 것 ≠ 실제 제조 현장에서의 높은 성능



[Fig 1] Correlation Heatmap



[Fig 2] Visualization of variables

## 02 문제 정의

### • 불완전 데이터셋 구축

- 본 연구는 완전한(Complete) 데이터셋에 제조 현장에서 빈번한 **데이터 수집 불완전성 시나리오**를 반영한 합성 데이터셋을 구축
- 각 시나리오 별 변경 강도는 1%, 5%, 10%, 20% 수준으로 조정하여 총 **24개의 데이터셋**으로 확장하여 생성
  1. **샘플링 주기·동기화 불일치(S1)**: 센서 별 측정 주기와 위상이 다르다고 가정, 공통 시간 격자에서 특정 시점을 비관측으로 처리함으로써 정렬 불일치를 재현
  2. **부분 결측 구간(S2)**: 일부 센서 장애나 유지보수로 인한 연속 결측 구간을 삽입
  3. **전사적 다운 타임(S3)**: 정전이나 네트워크 장애로 인한 전체 센서 데이터가 동시에 손실되는 상황을 모사
  4. **데이터 노이즈 (S4)**: 공정 기온, 작업자 수작업 오류 등의 자연적인 에러에 기반한 상황 구현
    - 가우시안 노이즈(S4-1)와 임펄스 스파이크(S4-2) 시나리오는 각각 백색 잡음과  $\pm(36)\sigma$  수준의 순간 이상을 주입하여 회로 잡음과 전원 서지를 재현
    - 시간지연(S4-3) 시나리오는 일부 센서에 0.5~2.0초의 고정 지연을 적용하여 시계열 비동기화를 반영

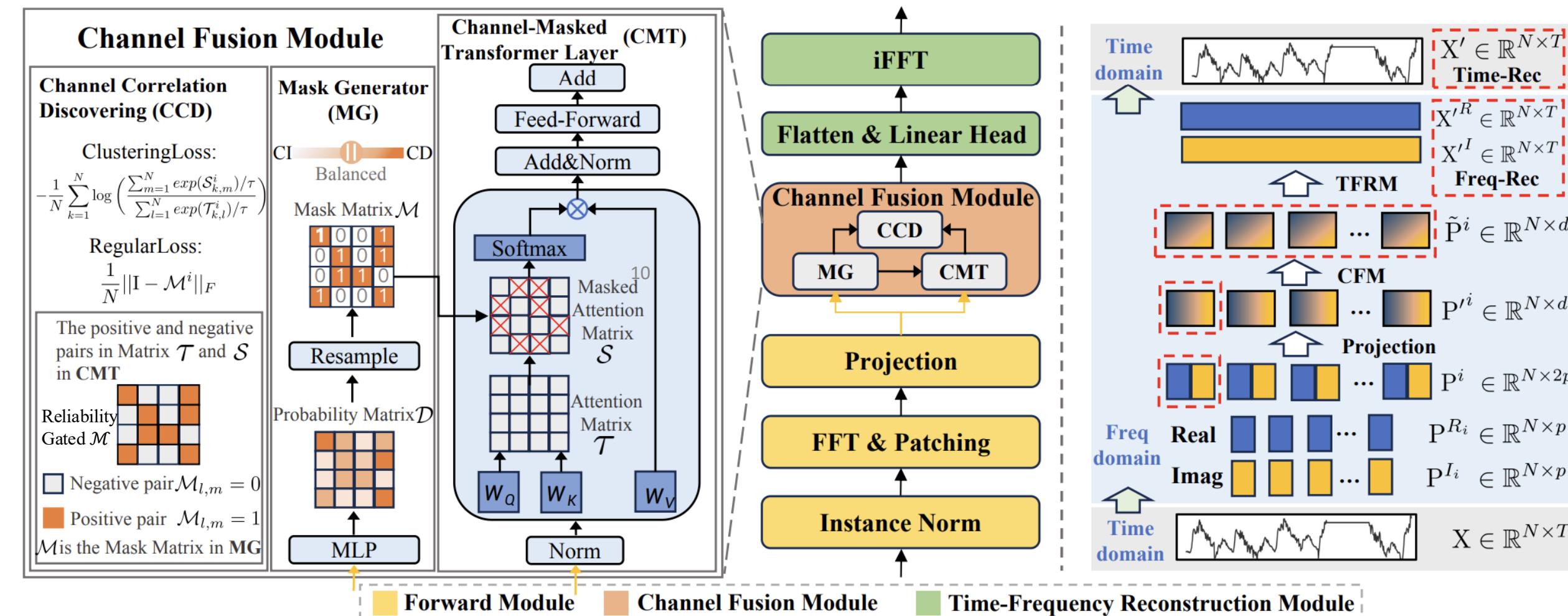
# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series

## Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Framework

- 1) Forward Module
- 2) Channel Fusion Module
- 3) Time-Frequency Reconstruction Module

[Fig 3] Framework of CATCH-RG



# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Forward Module

- Instance Norm Layer

- 학습 데이터와 테스트 데이터 간 통계적 분포 차이(distributional shift) 완화
    - 모델의 일반화 성능을 향상시켜 테스트 데이터 복원 성능 개선에 기여

- Fast Fourier Transform(FFT) & Patching

- FFT를 통해 입력 시계열  $X \in \mathbb{R}^{N \times T}$  을 frequency domain으로 변환
      - N = 시계열의 길이, T = 시계열의 차원
    - 실수부( $X^R$ ) 와 허수부 ( $X^I$ ) 를 모두 유지하여 최대 정보 보존
      - (참고)  $X^R \rightarrow$  코사인 성분의 진폭 의미
      - (참고)  $X^I \rightarrow$  사인 성분의 진폭 의미

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Forward Module

- Fast Fourier Transform(FFT) & Patching(cont'd)

- 이후 각 주파수 대역을 **fine-grained frequency patches**로 분할

- $P^{R_i}, P^{I_i} \in \mathbb{R}^{N \times p}$  는 각각  $X^R, X^L$  의 i 번째 patch를 의미

- $L = [T - p]/s + 1$  는 총 패치 수, p는 패치 길이, s는 패치 간격

- 각  $P^{R_i}, P^{I_i}$  를 합쳐  $P^i \in \mathbb{R}^{N \times 2p}$  를 생성

- Projection

12

- 각 patch  $P^i$  를 고차원의 projection layer에 투영하여 고차원 임베딩을 생성

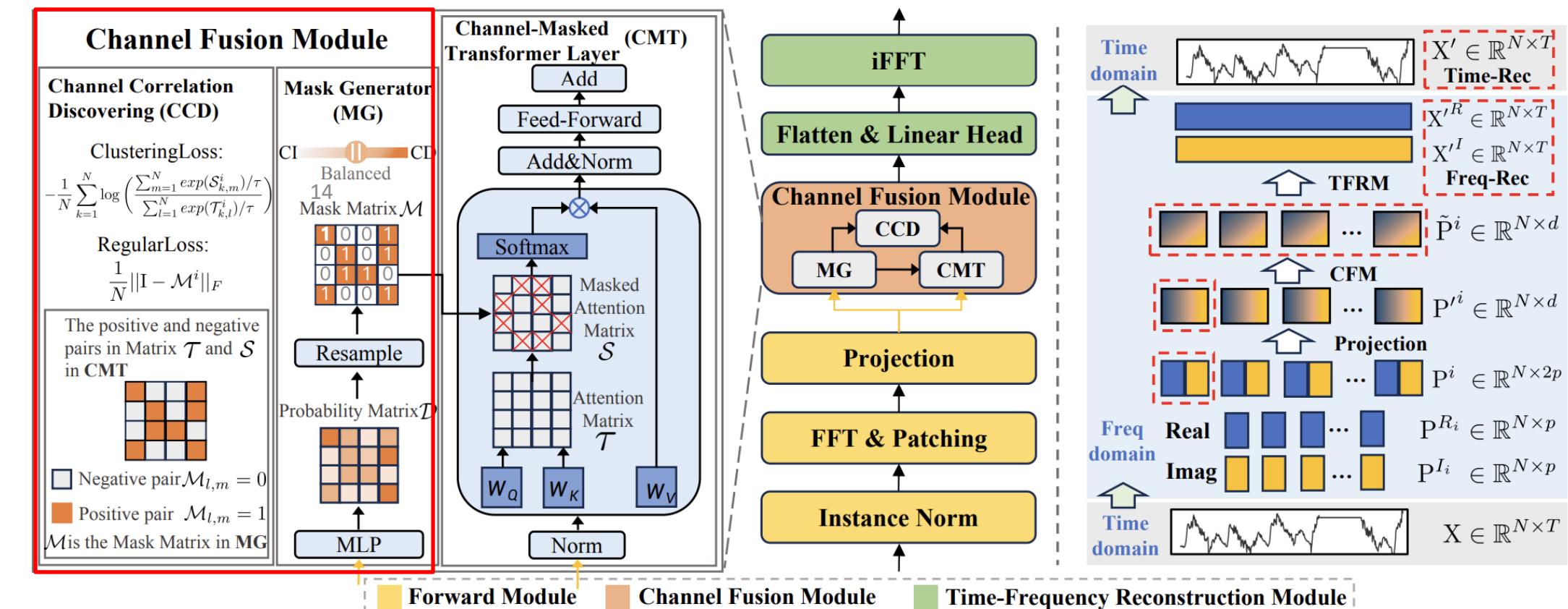
## 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Forward Module 이후 단계
  - Forward Module에서 도출된 임베딩  $P'^1, P'^2 \dots, P'^L$ 들을 Channel Fusion Module(CFM)에 input
    - N = 채널 수, d = attention block의 hidden dimension 수, L = patch 수
  - CFM은 patch 내의 채널들을 transformer의 입력 토큰으로 간주, band별 channel correlation을 학습
    - L개의 frequency band(patch)에 대한 병렬 처리 수행
    - 즉, 동일 band에서만 channel 간 attention을 수행
  - Time Frequency Reconstruction Module(TFRM)<sup>12</sup>은 실수부  $\{\tilde{P}^{R_1}, \tilde{P}^{R_2}, \dots, \tilde{P}^{R_L}\}$ , 허수부  $\{\tilde{P}^{I_1}, \tilde{P}^{I_2}, \dots, \tilde{P}^{I_L}\}$ 를 입력으로 받아  $X', X'^R, X'^I$ 를 출력하게 됨

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Framework

- 1) Forward Module
- 2) Channel Fusion Module
- 3) Time-Frequency Reconstruction Module



# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- **Channel Fusion Module**

- Reliability-Gated Channel Mask Generator (RG) : 데이터 불완전성에 강건하게 채널 간 상관관계를 나타내는 마스크 행렬 생성하는 단계
- Channel-Masked Transformer Layer(CMT) : 마스크 행렬을 이용해 유관 채널간의 관계를 학습하는 단계
- Channel Correlation Discovering(CCD) : MG가 만들어낸 마스크 행렬이 실제로 유관 채널 간의 관계를 반영하도록 학습

15

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

## • Reliability-Gated Channel Mask Generator

- Patch 별로 channel 간 correlation을 확률적으로 추정했지만, 실제 센서 환경에서는 노이즈·결측·전체 down 상황이 빈번히 발생 → 채널 통계 기반의 reliability-aware soft gating 을 도입, 신뢰도가 낮은 채널의 연결을 자동으로 억제

1. log-variance, abs-mean, diff, zero-ratio → 4D 통계로 channel reliability 추정
2. 신뢰도가 낮은 채널의 cross-channel 마스크를 확률적으로 약하게(혹은 거의 0에 가깝게) 만듦
3. 결측치는 숫자화 변환, 에러 방지 처리 적용
4. all-channel down 시에는 identity mask fallback 을 적용

→ 이로써 Reliability-Gated Channel Mask Generator는 단순 확률적 마스크를 넘어, 자체적으로 입력 품질을 판단하는 robust gating module 로 확장

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- **Reliability-Gated Channel Mask Generator**

- Channel association을 frequency patch마다 다르게 학습하는 것이 목적
  - frequency band(= patch)마다 binary mask matrix  $M_i \in \mathbb{R}^{N \times N}$ 를 생성
  - irrelevant channels의 adverse effect를 차단하고자 함
- CI–CD 사이의 중간 상태로써, binary mask  $M_i$ 가 두 극단 사이를 연속적으로 탐색하는 intermediate state로 가능
  - CI: identity mask ( $I$ )에 해당
  - CD: all-ones mask에 해당
- 각 패치의 임베딩  $P'^i$ 를 선형변환  $\rightarrow N \times N$  형태의 correlation score matrix  $D^i$ 로 변환
  - $D^i$  는 확률 행렬,  $P'^i$  는 hidden representation,  $\mathcal{M}^i$  는 i번째 패치의 이진 마스크 행렬
  - $\sigma$  는 sigmoid로, 모든 score를 확률로 바꿔  $D_{l,m}^i \in [0, 1]$  채널 l과 m이 연관될 확률로 해석
  - $D_{l,m}^i$  가 크면  $\mathcal{M}_{l,m}^i$  가 1에 가까워 지며, 이는 채널 l과 m간의 상관관계를 의미함

17

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Reliability-Gated Channel Mask Generator

- Diagonal 보정

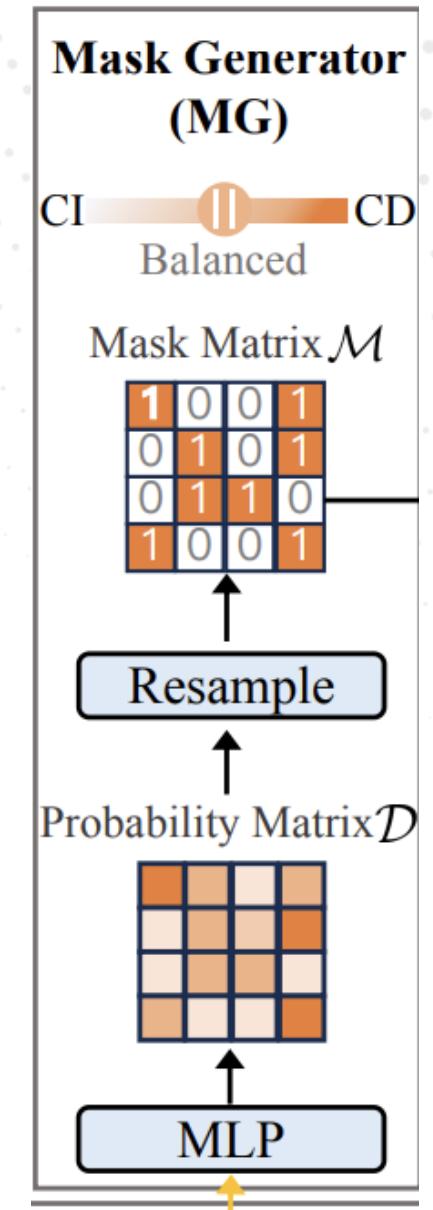
- 모든 채널의 자신 과의 점수( $l = m$ )는 항상 1로 설정  
→ self-correlation는 반드시 유지

$$\mathcal{D}^i = \sigma(\text{Linear}(\mathbf{P}'^i)), \mathcal{M}^i = \text{Resample}(\mathcal{D}^i), \text{where } \mathbf{P}'^i \in \mathbb{R}^{N \times d}, \mathcal{D}^i \in \mathbb{R}^{N \times N}, \text{ and } \mathcal{M}^i \in \mathbb{R}^{N \times N}$$

- Bernoulli resampling

- $D_{l,m}^i$  을 확률로 하여 0 또는 1을 샘플링  $\rightarrow M_{l,m}^i$ , (1<sup>18</sup>이면 두 채널이 연결, 0이면 연결 차단)
    - 그러나, 미분이 불가하므로 역전파가 불가능한 문제 발생

→ Gumbel Softmax Trick 이용



# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Reliability-Gated Channel Mask Generator

- Gumbel Softmax Trick

- 비연속적인 확률 샘플링(ex. Categorical 분포)을 연속적으로 근사하는 방법
    - Gumbel Distribution

$$z = \text{one\_hot} \left( \underset{i}{\arg \max} [g_i + \log \pi_i] \right)$$

미분불가능

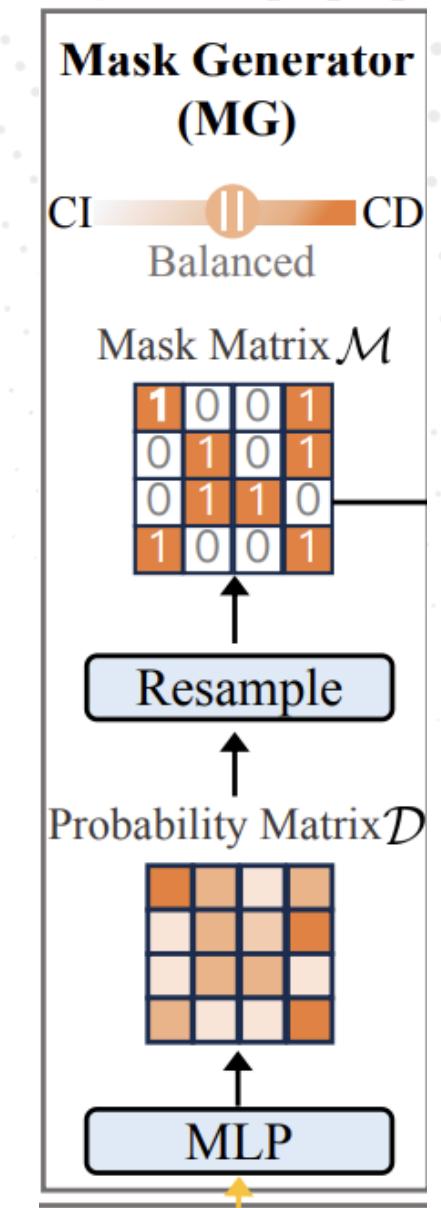
- $z$  를 샘플링하는 분포
    - $\pi_i$  : Class i에 속할 확률,  $g_1, \dots, g_k \sim Gumbel(0,1)^l$  : 무작위 노이즈

- Gumbel Softmax

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^k \exp((\log(\pi_j) + g_j)/\tau)} \quad \text{for } i = 1, \dots, k.$$

- Gumbel Distribution 을 softmax 형식으로 근사한 수식
    - $\tau$ : temperature hyperparameter → 작을수록 sharper ( $\approx$  one-hot encoding)
    - $y_i$ : 각 클래스가 선택될 확률 값의 근사치

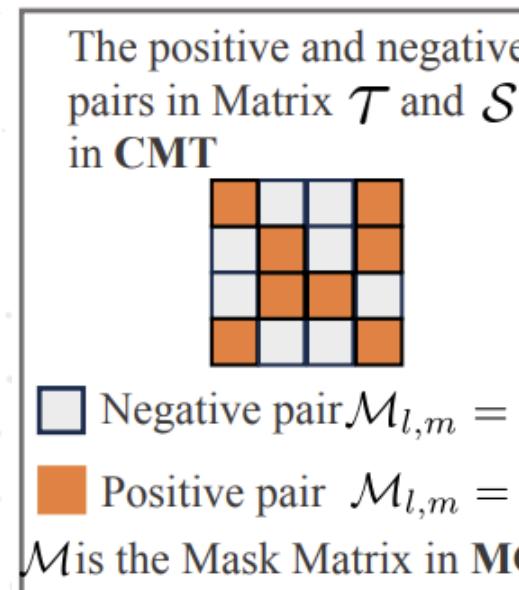
- CATCH도 Gumbel Softmax Trick을 사용해서 Bernoulli Resampling 수행



# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

## • Channel-Masked Transformer Layer(CMT)

- RG가 만든  $M_i$ 를 masked attention에 통합하여, 관련성이 높은 채널들 간의 fine-grained interrelationships를 학습.
- FFT 후 일부 주파수 성분의 진폭이 매우 크면, attention이 그 곳에만 과도하게 집중 문제 존재
- 이를 완화하기 위해, 각 patch  $P'_i$ 에 대해 attention 전에 LayerNorm 적용



$$P^{*i} = \text{LayerNorm}(P'^i) = (P'^i - \text{Mean}_{n=1}^N(P'^i_{n,:})) / \sqrt{\text{Var}_{n=1}^N(P'^i_{n,:})}$$

when  $P'^i \in \mathbb{R}^{N \times d}$  and  $P^{*i} \in \mathbb{R}^{N \times d}$

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- **Channel-Masked Transformer Layer(CMT)**

- 그리고  $P^{*i}$ 에 대해, Vanilla Attention을 수행하여  $Q^i, K^i, V^i$ 를 구함

$$Q^i = P^{*i} \cdot W^Q, K^i = P^{*i} \cdot W^K, V^i = P^{*i} \cdot W^V, W^Q, W^K, W^V \in \mathbb{R}^{d \times d}.$$

- 이후, 기본 Attention matrix  $T^i$ 를 구함

$$\mathcal{T}^i = Q^i \cdot (K^i)^T, \text{ where } \mathcal{T}^i \in \mathbb{R}^{N \times N}$$

- 이때  $T^i$ 는 채널 간 attention matrix로,  $T_{l,m}^i$ 는 channel l과 m 사이의 attention score를 의미

21

- 그리고 MG가 생성한 binary mask를 attention에 반영하여 irrelevant한 채널의 attention 효과 차단

$$\mathcal{S}^i = \mathcal{T}^i \odot \mathcal{M}^i + (1 - \mathcal{M}^i) \odot (-\infty)$$

- $M_{i,l,m} = 1$  : channel l과 m이 연관 있음 (positive pair)
    - $M_{i,l,m} = 0$  : 관계 없음 (negative pair  $\rightarrow$  attention 차단)
    - $\odot$ : element-wise product

## 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- **Channel-Masked Transformer Layer(CMT)**

- 그리고  $S^i$ 에 대해, normalized attention score를 구한 뒤,  $V^i$ 와 행렬곱 하여 최종적으로  $\tilde{P}_i$ 를 도출

$$\text{MaskedScores}^i = \mathcal{S}^i / \sqrt{d}, \tilde{\mathbf{P}}^i = \text{Softmax}(\text{MaskedScores}^i) \cdot \mathbf{V}^i, \text{ where } \tilde{\mathbf{P}}^i \in \mathbb{R}^{N \times d}$$

- 이때,  $\tilde{P}_i$ 는 CMT에서 처리된 i번째 patch representation을 의미함

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Channel-Correlation Discovering(CCD)

- Mask 가 “1(relevant)”로 표시한 채널 쌍의 attention score를 직접 높이도록 유도
    - attention을 현재 발견된 최적의 channel correlation 방향으로 정렬
    - **irrelevant noise 채널을 고립시켜 강건성 확보 가능**
    - Mask가 유관하다고 표시한 채널 쌍의 attention score를 직접 높임

$$\text{ClusteringLoss} = -\frac{1}{N} \sum_{k=1}^N \log \left( \frac{\sum_{m=1}^N \exp(\mathcal{S}_{k,m}^i)/\tau}{\sum_{l=1}^N \exp(\mathcal{T}_{k,l}^i)/\tau} \right), \quad (9) \quad \text{RegularLoss} = \frac{1}{N} \|\mathbf{I} - \mathcal{M}^i\|_F, \quad (10)$$

- CCD는 MG가 생성한 channel correlation을 반복적으로 refine
  - 동시에 CMT의 **attention capacity**를 조정하여 patch-wise channel interrelationships를 충분히 포착하도록 함

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- **Channel-Correlation Discovering( CCD )**

- RegularLoss는 Mask Generator가 모든 채널을 연결하는 방향으로 degeneration하는 것을 방지하고자 함. 즉, 연결 수를 제한하여 sparsity를 유지
- 결과적으로 mask가 과도하게 dense해지는 것을 막음 → 더 명확한 channel grouping 가능

$$\text{ClusteringLoss} = -\frac{1}{N} \sum_{k=1}^N \log \left( \frac{\sum_{m=1}^N \exp(\mathcal{S}_{k,m}^i)/\tau}{\sum_{l=1}^N \exp(\mathcal{T}_{k,l}^i)/\tau} \right), \quad (9) \quad \text{RegularLoss} = \frac{1}{N} \|\mathbf{I} - \mathcal{M}^i\|_F, \quad (10)$$

- $N$ : 채널 개수
- $I$ : identity matrix
- $\|\cdot\|_F$  : Frobenius norm

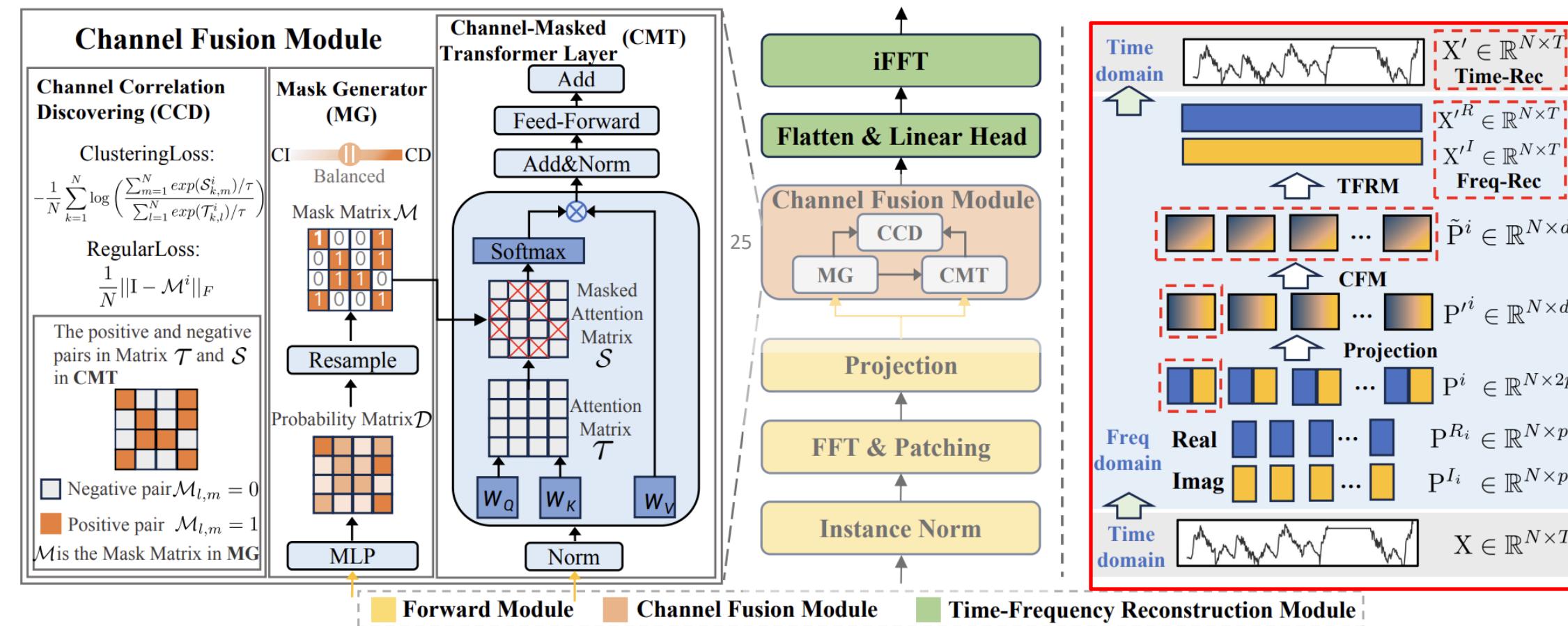
24

- 두 손실의 결합을 통해 MG는 CI(완전 독립)와 CD(완전 결합) 사이의 중간 공간에서 적절한 patch-wise channel correlation 을 발견하도록 유도됨.
- 동시에,  $W_Q, W_K$  가 fine-grained representation을 학습하며 attention mechanism이 더욱 강화

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Framework

- 1) Forward Module
- 2) Channel Fusion Module
- 3) Time-Frequency Reconstruction Module

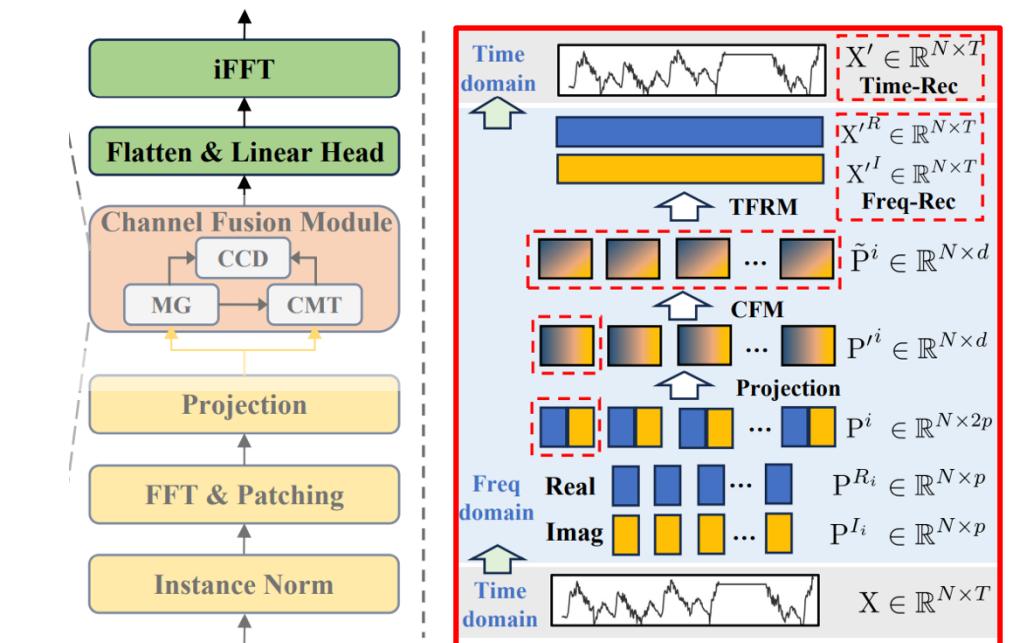


# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Time-Frequency Reconstruction Module

- Time-Frequency Reconstruction Moduled은 Flatten & Linear Head Layer 와 iFFT Layer로 구성
- CFM에서 fine-grained channel correlations를 모두 추출한 이후, patch-wise representation들을 flatten하여 실수, 허수부 각각에 대해 MLP projection을 적용해 frequency spectrum 전체를 복원
- 이어서 inverse FFT(iFFT) 를 통해 time-domain signal을 재구성

$$\begin{aligned}
 X'^R &= \text{Projection}_R(\text{FlattenHead}(\{\tilde{P}^{R_1}, \tilde{P}^{R_2}, \dots, \tilde{P}^{R_L}\})), \\
 X'^I &= \text{Projection}_I(\text{FlattenHead}(\{\tilde{P}^{I_1}, \tilde{P}^{I_2}, \dots, \tilde{P}^{I_L}\})), \\
 X' &= \text{iFFT}(X'^R, X'^I),
 \end{aligned}$$



# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- **Time-Frequency Reconstruction Module**

- 그리고 **point anomaly**와 **subsequence anomaly**를 모두 학습할 수 있도록 **time domain**과 **frequency domain** 각각에 대해 손실 함수를 아래와 같이 정의

$$\text{RecLoss}^{time} = \|\mathbf{X} - \mathbf{X}'\|_F^2 \quad (14) \quad \text{RecLoss}^{freq} = \|\mathbf{X}^R - \mathbf{X}'^R\|_1 + \|\mathbf{X}^I - \mathbf{X}'^I\|_1$$

- 시간 도메인에서는 시점별 차이를 정확히 복원해야 하므로 MSE 사용  $\rightarrow$  point-wise 복원 능력 강화.
- 주파수 도메인은 수치적 스케일과 분포가 상대적으로 불균일하므로 MAE 사용  $\rightarrow$  진폭 및 위상 차이에 대해 강건하게 복원 가능화

# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

- Joint-Bi level Optimization

- CATCH는 point anomaly와 subsequence anomaly를 동시에 탐지하기 위해 joint bi-level training process을 설계함.

- mask → model → mask → model ... 순으로 iterative refinement
  - 결과적으로 attention mechanism이 실제 의미 있는 channel dependencies에 점점 정렬되도록 유도

- 아래 두 가지 과업을 순차적으로 수행해 강건한 이상 탐지 수행 가능화

- 시간/주파수 복원 능력 강화 → anomaly detection 성능 향상<sup>28</sup>

- Channel Correlation Discovering(CCD) 의 효과적 학습

- 적절한 channel mask 탐색

---

**Algorithm 1** Bi-level Gradient Descent Optimization

---

```
1: Input: Model parameters  $\theta_{\text{model}}$ ,  $\theta_{\text{mask}}$ , learning rate  $\eta_{\text{model}}$ ,  $\eta_{\text{mask}}$ , number of iterations  $N_O$ ,  $N_I$ , loss function  $\mathcal{L} = \text{RecLoss}^{time} + \lambda_1 \cdot \text{RecLoss}^{freq} + \lambda_2 \cdot \text{ClusteringLoss} + \lambda_3 \cdot \text{RegularLoss}$ 
2: Initialize:  $\theta_{\text{model}} \leftarrow \text{initial value}$ ,  $\theta_{\text{mask}} \leftarrow \text{initial value}$ 
3: For  $i = 1$  to  $N_O$  Do
4:   Outer Loop: Update the mask generator parameters
5:    $\theta_{\text{mask}} \leftarrow \theta_{\text{mask}} - \eta_{\text{mask}} \cdot \nabla_{\theta_{\text{mask}}} \mathcal{L}(\theta_{\text{model}}, \theta_{\text{mask}})$  ▷ Update the Mask Generator
6:   For  $j = 1$  to  $N_I$  Do
7:     Inner Loop: Update the model parameters
8:      $\theta_{\text{model}} \leftarrow \theta_{\text{model}} - \eta_{\text{model}} \cdot \nabla_{\theta_{\text{model}}} \mathcal{L}(\theta_{\text{model}}, \theta_{\text{mask}})$  ▷ Update the model
9:   EndFor
10: EndFor
11: Output: Optimized parameters  $\theta_{\text{model}}, \theta_{\text{mask}}$ 
```

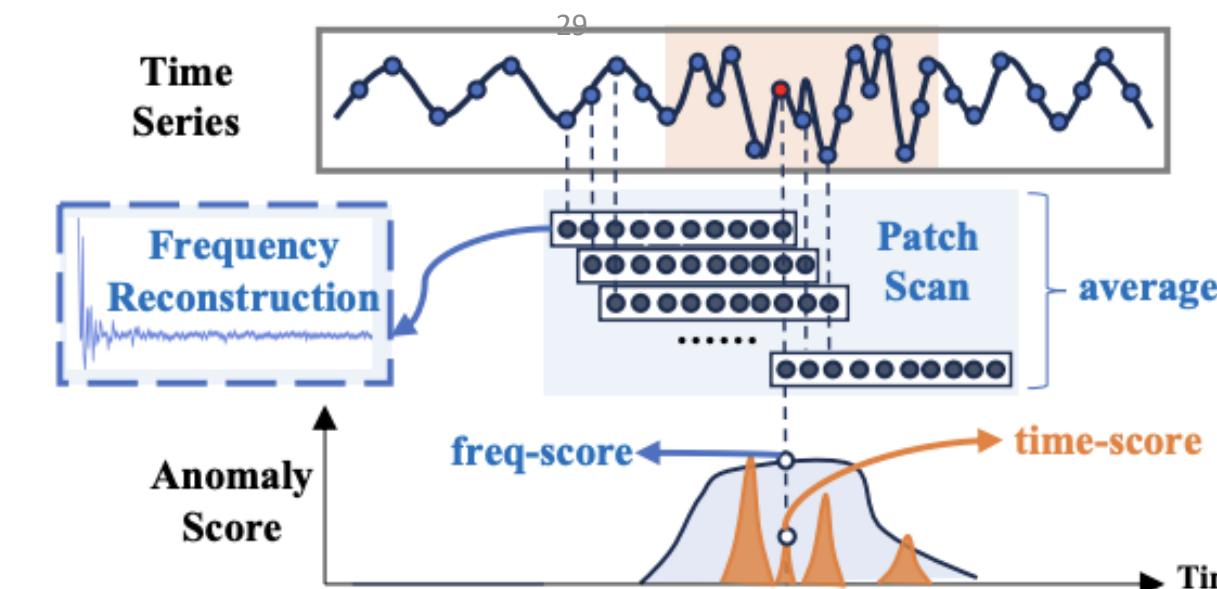
---

## 제안 알고리즘 소개

# CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

### • Anomaly Scoring

- 기존 TSAD 모델들은 보통 point-to-point scoring 방식을 사용
  - 각 timestamp 별로 reconstruction error를 계산해 이상 점수를 부여
  - 그러나 time domain에서의 point anomaly에만 민감, subsequence anomaly 탐지에는 한계 존재
- 일부 연구들은 이를 보완하기 위해 window-granularity scoring 을 도입
  - 입력 window 전체에 대해 frequency domain 기반 score를 추가
  - 그러나 이 방법은 subsequence anomaly의 정확한 경계를 알 수 없어 탐지 오류 또는 누락 가능



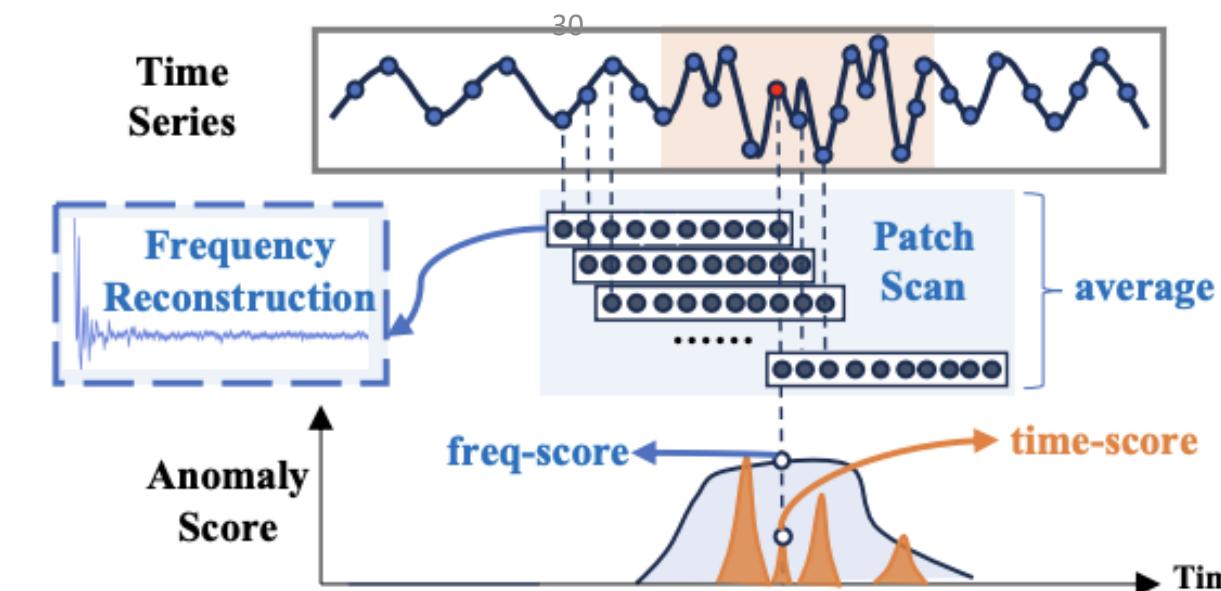
# 제안 알고리즘 소개 CATCH-RG : Channel-Aware Multivariate Time-Series Anomaly Detection via Patching with Reliability-Gated Channel Mask Generator

## • Anomaly Scoring

- 그래서 CATCH는 stride = 1로 frequency patch를 뽑아, 모든 시점이 여러 patch 내에서 평가될 수 있도록 함.
- 특정 시점이 포함된 모든 subsequence를 고려하므로, **제조 데이터 셋에 빈번한 노이즈, 결측 등의 문제에 강건하게 이상 여부를 판별할 수 있음.**

$$\text{AnomalyScore} = \text{time-score} + \lambda_{score} \cdot \text{freq-score}$$

- Time-score : Point 단위 reconstruction error 기반 점수
- Frequency score : FFT로 변환한 주파수 패치에 대한 reconstruction error 기반점수



# 04

## 실험 설정

- **Data Preprocessing**

- 작업자가 현장에서 그대로 사용하여도 높은 성능을 보장
- Scaling 외의 어떠한 유리한 전처리(ex. 컬럼 선택, 파생변수 수동 설정 등)도 수행하지 않음

- **Dataset**

- 소성가공 데이터 셋의 4가지 시나리오로 전처리한 데이터 셋 활용
  - **S1 : 수집 주기/동기 불일치** - 센서마다 데이터 찍는 속도·타이밍이 달라 같은 시각에 값이 비어 있거나 어긋나는 상황을 재현
  - **S2 : 부분 결측 구간** - 특정 센서만 한동안 끊기는 연속 빈 구간(센서 고장·임시 중단 등)을 재현
  - **S3 : 전역 결측(다운타임)** - 정전·네트워크 장애처럼 모든 센서가 동시에 비는 시간대를 재현
  - **S4 : 노이즈·스파이크·시간 지연** - 환경 간섭·글리치로 생기는 작은 흔들림(노이즈), 갑작스런 팀(스파이크), 늦게 찍히는 값(Lag)을 재현

31

- **비교 방법론**

- Dlinear, Nlinear, TimesNet, iTransformer, PatchTST

# 04

## 실험 설정

- **Hyperparameter Settings**
  - DLinear
    - Batch\_size: 128, d\_ff: 16, d\_model: 8, e\_layers: 1
    - horizon: 0, norm: true, num\_epochs: 10, seq\_len: 80
  - NLinear
    - Batch\_size: 128, d\_ff: 16, d\_model: 8, e\_layers: 1
    - horizon: 0, norm: true, num\_epochs: 10, seq\_len: 80
  - iTransformer
    - Batch\_size: 64, d\_ff: 512, d\_model: 256, e\_layers: 1
    - horizon: 0, lr: 0.0005, norm: true, num\_epochs: 5, seq\_len: 64

32

# 04

## 실험 설정

- **Hyperparameter Settings**
  - PatchTST
    - Batch\_size: 128, d\_ff: 128, d\_model: 128, e\_layers: 3
    - horizon: 0, norm: true, num\_epochs: 15, seq\_len: 100
  - TimesNet
    - Batch\_size: 128, d\_ff: 64, d\_model: 64
    - e\_layers: 2, horizon: 0, norm: true, num\_epochs: 10, seq\_len: 100
  - CATCH-RG
    - Mlr: 5e-05, Batch\_size: 128, cf\_dim: 64, d\_ff: 256
    - d\_model: 128, e\_layers: 3, head\_dim: 64, lr: 0.0005
    - n\_heads: 2, num\_epochs: 5, patch\_size: 16, patch\_stride: 8, seq\_len: 192

33

# 실험 설정

- **Evaluation Metrics**

- 강건하고 엄밀한 이상 탐지 성능 비교를 위해, 최신 시계열 이상탐지 연구들이 채택하고 있는 Point 기반 지표 및 Range 기반 지표를 함께 비교
  - 특정 평가지표에 특화되지 않은 강건한 이상탐지 성능을 비교하고자 함
- **VUS-PR**(Volume Under Surface – Precision Recall Curve)
- **VUS-ROC** (Volume Under Surface of ROC Curve)
- **Range-F1**(Range-based F1-score)
- **AUC-PR**(Area Under the Precision–Recall Curve)
- **AUC-ROC**(Area Under the Receiver Operating Characteristic Curve)
- **Point-F1**(Point-wise F1-score)

34

# 실험 결과

- Scenario 1

- 수집 주기/동기 불일치: 센서마다 데이터 찍는 속도·타이밍이 달라 같은 시각에 값이 비어 있거나 어긋나는 시나리오
  - 강도 1%, 5%, 10%, 20% : 데이터 불일치 정도를 의미
- 6개의 지표에서 모두 CATCH-RG가 우수한 성능을 보임을 확인

\*Best **bold**, Second Best underline

Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1	Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1
DLinear	1%	0.8423	0.0244	0.8626	0.0287	0.0662	0.0585	TimesNet	1%	<u>0.9713</u>	<u>0.1016</u>	<u>0.9779</u>	<u>0.1351</u>	<u>0.2298</u>	0.0513
	5%	0.7900	0.0171	0.8193	0.0199	0.0488	0.0845		5%	<u>0.8899</u>	<u>0.0264</u>	<u>0.9035</u>	<u>0.0325</u>	<u>0.0636</u>	0.0262
	10%	0.6265	0.0091	0.6548	0.0101	0.0260	0.0166		10%	<u>0.7999</u>	<u>0.0275</u>	<u>0.8347</u>	<u>0.0346</u>	<u>0.0526</u>	0.1935
	20%	0.5179	0.0066	0.5837	0.0087	0.0159	0.0164		20%	<u>0.7180</u>	<u>0.0281</u>	<u>0.7864</u>	<u>0.0395</u>	<u>0.1115</u>	0.1500
NLinear	1%	0.8376	0.0265	0.8940	0.0367	0.0703	0.0447	PatchTST	1%	0.8360	0.0329	0.8789	0.0412	0.1094	0.0732
	5%	0.6237	0.0088	0.7025	0.0116	0.0196	<u>0.0249</u>		5%	0.6628	0.0113	0.7351	0.0143	0.0299	0.0301
	10%	0.5408	0.0068	0.5847	0.0076	0.0155	0.0484		10%	0.4954	0.0062	0.5999	0.0092	0.0183	0.0303
	20%	0.4836	0.0061	0.5615	0.0081	0.0143	0.0173		20%	0.5511	0.0084	0.6454	0.0112	0.0294	0.0663
iTransformer	1%	0.8002	0.0489	0.8891	0.0752	0.1214	<u>0.1939</u>	CATCH-RG (Ours)	1%	<b>0.9961</b>	<b>0.6227</b>	<b>0.9981</b>	<b>0.7164</b>	<b>0.5999</b>	<b>0.3596</b>
	5%	0.7172	0.0178	0.8038	0.0239	0.0473	<u>0.1132</u>		5%	<b>0.9959</b>	<b>0.6190</b>	<b>0.9982</b>	<b>0.7341</b>	<b>0.6028</b>	<b>0.3725</b>
	10%	0.5775	0.0100	0.6553	0.0115	0.0293	0.0682		10%	<b>0.9960</b>	<b>0.6291</b>	<b>0.9983</b>	<b>0.7391</b>	<b>0.6009</b>	<b>0.4023</b>
	20%	0.5537	0.0067	0.6394	0.0091	0.0162	0.0125		20%	<b>0.9966</b>	<b>0.6098</b>	<b>0.9983</b>	<b>0.7203</b>	<b>0.6006</b>	<b>0.3805</b>

# 실험 결과

- Scenario 2

– 부분 결측 구간: 특정 센서만 한동안 끊기는 연속 빈 구간(센서 고장·임시 중단 등)을 재현한 시나리오

- 강도 1%, 5%, 10%, 20% : 데이터 결측 정도를 의미

– 6개의 지표에서 모두 CATCH-RG가 우수한 성능을 보임을 확인

\*Best **bold**, Second Best underline

Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1	Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1
DLinear	1%	0.8268	0.0234	0.8600	0.0284	0.0667	0.0671	TimesNet	1%	<u>0.9704</u>	<u>0.0914</u>	<u>0.9751</u>	<u>0.1245</u>	<u>0.2009</u>	<u>0.0303</u>
	5%	0.7413	0.0142	0.7743	0.0156	0.0383	0.0355		5%	<u>0.8762</u>	<u>0.0321</u>	<u>0.8893</u>	<u>0.0378</u>	<u>0.0616</u>	<u>0.1935</u>
	10%	0.4425	0.0101	0.5444	0.0083	0.0314	0.1502		10%	<u>0.8017</u>	<u>0.0215</u>	<u>0.8345</u>	<u>0.0217</u>	<u>0.0439</u>	<u>0.1764</u>
	20%	0.6989	0.0097	0.7470	0.0126	0.0238	0.0186		20%	<u>0.8017</u>	<u>0.0460</u>	<u>0.8507</u>	<u>0.0567</u>	<u>0.1239</u>	<u>0.1153</u>
NLinear	1%	0.8096	0.0264	0.8727	0.0336	0.0760	0.0493	PatchTST	1%	0.6334	0.0107	0.7373	0.0179	0.0459	0.0268
	5%	0.6444	0.0089	0.6842	0.0098	0.0222	<sup>36</sup> 0.0172		5%	0.4369	0.0054	0.5299	0.0072	0.013	0.0237
	10%	0.4991	0.0059	0.5885	0.0081	0.0149	0.0177		10%	0.5988	0.0087	0.6516	0.0097	0.0196	0.0566
	20%	0.5832	0.0074	0.6720	0.0107	0.0171	0.0159		20%	0.6274	0.0096	0.7058	0.0125	0.0225	0.0314
iTransformer	1%	0.7924	0.0469	0.8694	0.0612	0.1067	0.1768	CATCH-RG (Ours)	1%	<b>0.9958</b>	<b>0.6200</b>	<b>0.9980</b>	<b>0.7228</b>	<b>0.5934</b>	<b>0.3842</b>
	5%	0.6878	0.0121	0.7411	0.0139	0.0297	0.0267		5%	<b>0.9955</b>	<b>0.6239</b>	<b>0.9980</b>	<b>0.7146</b>	<b>0.5826</b>	<b>0.3734</b>
	10%	0.5632	0.0076	0.6707	0.0113	0.0187	0.0319		10%	<b>0.9963</b>	<b>0.6311</b>	<b>0.9983</b>	<b>0.7343</b>	<b>0.5985</b>	<b>0.3663</b>
	20%	0.5598	0.0074	0.6502	0.0101	0.0162	0.0187		20%	<b>0.9963</b>	<b>0.6112</b>	<b>0.9982</b>	<b>0.7122</b>	<b>0.5963</b>	<b>0.3519</b>

# 실험 결과

- Scenario 3

- 전역 결측(다운타임): 정전·네트워크 장애처럼 모든 센서가 동시에 비는 시간대를 재현한 시나리오

- 강도 1%, 5%, 10%, 20% : 데이터 결측 정도를 의미

- 6개의 지표에서 CATCH-RG는 유의미한 성능을 보임을 확인

\*Best **bold**, Second Best underline

Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1	Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1
DLinear	1%	<u>0.8465</u>	0.0284	0.8737	0.0326	0.0762	0.0276	TimesNet	1%	0.7579	0.0212	0.8146	0.0440	0.1161	0.0210
	5%	0.6260	0.0081	0.6582	0.0093	0.0190	0.0182		5%	0.6985	0.0124	0.7347	0.0173	0.0413	0.0585
	10%	0.3633	0.0072	0.5757	0.0107	0.0292	0.1306		10%	0.5875	0.0090	0.7272	0.0169	0.0302	0.0211
	20%	0.4105	0.0059	0.5299	0.0081	0.0219	0.0441		20%	0.5598	0.0087	0.6295	0.0118	0.0295	0.0292
NLinear	1%	0.8175	0.0679	0.8511	0.0572	0.1867	0.2853	PatchTST	1%	0.5641	0.0069	0.6418	0.0095	0.0171	0.0063
	5%	0.5830	0.0073	0.6292	0.0085	0.0171	<u>0.0158</u>		5%	0.5114	0.0095	0.5626	0.0076	0.0292	0.1624
	10%	0.4774	0.0056	0.6256	0.0099	0.0143	0.0140		10%	0.5534	0.0113	0.6867	0.0143	0.0444	0.1155
	20%	0.4621	0.0056	0.5543	0.0078	0.0133	0.0132		20%	0.5807	<u>0.0089</u>	0.6343	0.0096	<u>0.0225</u>	<u>0.0895</u>
iTransformer	1%	0.8302	<u>0.0756</u>	0.8840	0.0873	0.1688	0.1939	CATCH-RG (Ours)	1%	<b>0.9307</b>	<b>0.0816</b>	<b>0.9501</b>	<b>0.1077</b>	<b>0.2337</b>	<b>0.0595</b>
	5%	<u>0.7145</u>	<u>0.0463</u>	<b>0.7787</b>	<b>0.0429</b>	<b>0.0956</b>	<b>0.1939</b>		5%	<b>0.7261</b>	<u>0.0139</u>	<u>0.7719</u>	<u>0.0167</u>	<u>0.0408</u>	<u>0.0226</u>
	10%	<b>0.6455</b>	<b>0.0385</b>	<b>0.7800</b>	<b>0.0282</b>	<b>0.0530</b>	<b>0.1939</b>		10%	<u>0.6354</u>	<u>0.0086</u>	<u>0.7560</u>	<u>0.0145</u>	<u>0.0209</u>	<u>0.0159</u>
	20%	<b>0.6545</b>	<b>0.0150</b>	<b>0.7448</b>	<b>0.0211</b>	<b>0.0461</b>	<b>0.0638</b>		20%	<u>0.5872</u>	0.0075	<u>0.6662</u>	<u>0.0113</u>	0.0223	0.0510

# 실험 결과

- Scenario 4\_1

- 노이즈·스파이크·시간 지연: 환경 간섭·글리치로 생기는 작은 흔들림(노이즈), 갑작스런 텁(스파이크), 늦게 찍히는 값(지연 /Lag)을 재현한 시나리오 (강도 1%, 5%, 10%, 20% : 노이즈 포함 정도를 의미)
- 4개의 지표에서 모두 CATCH-RG가 우수한 성능을 보임을 확인

\*Best **bold**, Second Best underline

Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1	Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1
DLinear	1%	0.8327	0.0944	0.8604	0.0455	0.1344	<u>0.3532</u>	TimesNet	1%	<u>0.9817</u>	0.4727	<u>0.9955</u>	0.5241	<u>0.6099</u>	<u>0.3758</u>
	5%	0.8327	0.0944	0.8604	0.0457	0.1344	<u>0.3532</u>		5%	<u>0.9813</u>	0.4709	<u>0.9954</u>	0.5235	<u>0.6077</u>	<u>0.3766</u>
	10%	0.8327	0.0944	0.8605	0.0455	0.1344	<u>0.3532</u>		10%	<u>0.9817</u>	0.4727	<u>0.9955</u>	0.5238	<u>0.6107</u>	<u>0.3819</u>
	20%	0.8328	0.0944	0.8608	0.0456	0.1344	<u>0.3532</u>		20%	<u>0.9816</u>	0.4760	<u>0.9957</u>	0.5245	<u>0.6153</u>	<u>0.3752</u>
NLinear	1%	0.8309	<u>0.4823</u>	0.8951	0.5040	<b>0.6145</b>	0.3741	PatchTST	1%	0.9282	0.3646	0.9786	0.4641	0.4519	0.3944
	5%	0.8309	<u>0.4822</u>	0.8951	0.5040	<b>0.6127</b>	0.3741		5%	0.9287	0.3648	0.9787	0.4638	0.4519	0.3944
	10%	0.8310	<u>0.4822</u>	0.8951	0.4927	<b>0.6127</b>	0.3741		10%	0.9273	0.3648	0.9782	0.4635	0.4524	0.3944
	20%	0.8313	<u>0.4815</u>	0.8954	0.4919	<b>0.6127</b>	0.3741		20%	0.9230	0.3688	0.9763	0.4592	0.4593	0.4082
iTransformer	1%	0.8121	0.3010	0.9057	0.3817	0.4444	0.3430	CATCH-RG (Ours)	1%	<b>0.9956</b>	<b>0.5682</b>	<b>0.9979</b>	<b>0.6863</b>	0.5733	0.3382
	5%	0.8178	0.3057	0.9087	0.3773	0.4482	0.3435		5%	<b>0.9958</b>	<b>0.5671</b>	<b>0.9979</b>	<b>0.6869</b>	0.5724	0.3387
	10%	0.8367	0.3007	0.9178	0.3836	0.4678	0.3441		10%	<b>0.9955</b>	<b>0.5638</b>	<b>0.9978</b>	<b>0.6758</b>	0.5818	0.3437
	20%	0.8429	0.2800	0.9264	0.3704	0.4395	0.3389		20%	<b>0.9956</b>	<b>0.5675</b>	<b>0.9979</b>	<b>0.6882</b>	0.5789	0.3504

# 실험 결과

- Scenario 4\_2

- 노이즈·스파이크·시간 지연: 환경 간섭·글리치로 생기는 작은 흔들림(노이즈), 갑작스런 텁(스파이크), 늦게 찍히는 값(지연 /Lag)을 재현한 시나리오 (강도 1%, 5%, 10%, 20% : 노이즈 포함 정도를 의미)
- 6개의 지표에서 모두 CATCH-RG가 우수한 성능을 보임을 확인

\*Best **bold**, Second Best underline

Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1	Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1
DLinear	1%	0.8347	0.0867	0.8625	0.0452	0.1300	0.3532	TimesNet	1%	<u>0.9687</u>	0.2872	<u>0.9883</u>	<u>0.2860</u>	0.3903	0.2711
	5%	0.8348	0.0769	0.8628	0.0456	0.1202	0.3246		5%	<u>0.9619</u>	0.2302	<u>0.9757</u>	0.2003	0.2535	0.2711
	10%	0.8243	0.0721	0.8593	0.0500	0.1265	0.3246		10%	<u>0.9395</u>	0.2489	<u>0.9643</u>	0.2356	0.3035	0.3346
	20%	0.8202	0.0615	0.8532	0.0462	0.1025	0.2377		20%	<u>0.9305</u>	0.2267	<u>0.9576</u>	0.2007	0.2538	0.2711
NLinear	1%	0.8387	<u>0.4334</u>	0.9038	0.3660	0.5066	0.3655	PatchTST	1%	0.9212	0.2234	0.9697	0.2027	0.3062	0.2631
	5%	0.8511	<u>0.3728</u>	0.9144	<u>0.2991</u>	0.4338	0.3583		5%	0.9003	0.1857	0.9532	0.1474	0.2322	0.2255
	10%	0.8443	<u>0.3099</u>	0.9104	<u>0.2651</u>	0.3899	0.3434		10%	0.8999	0.2085	0.9482	0.1719	0.2702	0.3451
	20%	0.8507	<u>0.2870</u>	0.9099	<u>0.2199</u>	0.3678	0.3405		20%	0.8761	0.2214	0.9269	0.1713	0.2993	0.3500
iTransformer	1%	0.7367	0.3283	0.8461	0.2593	0.4268	0.2	CATCH-RG (Ours)	1%	<b>0.9952</b>	<b>0.5476</b>	<b>0.9978</b>	<b>0.6642</b>	<b>0.5781</b>	<b>0.3740</b>
	5%	0.7872	0.3378	0.8813	0.2673	0.4303	0.2008		5%	<b>0.9963</b>	<b>0.5918</b>	<b>0.9982</b>	<b>0.6913</b>	<b>0.6137</b>	<b>0.3585</b>
	10%	0.7550	0.3047	0.8703	0.2484	0.3976	0.1992		10%	<b>0.9961</b>	<b>0.5991</b>	<b>0.9981</b>	<b>0.7039</b>	<b>0.6060</b>	<b>0.3608</b>
	20%	0.7448	0.1952	0.8490	0.1243	0.2799	0.1974		20%	<b>0.9967</b>	<b>0.6022</b>	<b>0.9984</b>	<b>0.7373</b>	<b>0.6180</b>	<b>0.4216</b>

# 실험 결과

- Scenario 4\_3

- 노이즈·스파이크·시간 지연: 환경 간섭·글리치로 생기는 작은 흔들림(노이즈), 갑작스런 텁(스파이크), 늦게 찍히는 값(지연 /Lag)을 재현한 시나리오 (강도 0.5s, 1.0s, 1.5s, 2.0s : 시간 지연 정도를 의미)
- 6개의 지표에서 대체로 CATCH-RG가 우수한 성능을 보임을 확인

\*Best **bold**, Second Best underline

Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1	Model	Scenario	AUROC	AUPRC	VUSROC	VUSPR	PointF1	RangeF1
DLinear	0.5s	0.8311	0.0772	0.8597	0.0379	0.1333	0.3306	TimesNet	0.5s	<u>0.9824</u>	0.3412	<u>0.9928</u>	0.3548	0.4681	0.2253
	1.0s	0.8308	0.0689	0.8595	0.0379	0.1322	0.3363		1.0s	<u>0.9686</u>	<u>0.3154</u>	<u>0.9911</u>	<u>0.3452</u>	0.4580	0.2200
	1.5s	0.8313	0.0628	0.8597	0.0379	0.1311	<u>0.3385</u>		1.5s	<u>0.9748</u>	<u>0.3006</u>	<u>0.9916</u>	<u>0.3459</u>	0.4540	0.2045
	2.0s	0.8319	0.0582	0.8599	0.0377	0.1300	<u>0.3363</u>		2.0s	<u>0.9739</u>	<u>0.2851</u>	<u>0.9914</u>	<u>0.3399</u>	0.4478	0.2200
NLinear	0.5s	0.8304	<u>0.4007</u>	0.8946	<u>0.3969</u>	0.5463	<b>0.3445</b>	PatchTST	0.5s	0.9126	0.3012	0.9706	0.3940	0.4017	0.2134
	1.0s	0.8310	0.2983	0.8952	0.3211	<b>0.4977</b>	0.3190		1.0s	0.9113	0.2616	0.9690	0.3592	0.3529	0.2066
	1.5s	0.8326	0.2391	0.8966	0.2914	<b>0.5070</b>	0.2996		1.5s	0.8970	0.2551	0.9621	0.3487	0.3458	0.3218
	2.0s	0.8334	0.2261	0.8973	0.2838	0.4909	0.2950		2.0s	0.8982	0.2711	0.9623	0.3839	0.3909	0.3176
iTransformer	0.5s	0.7895	0.1914	0.8886	0.2889	0.3980	0.2897	CATCH-RG (Ours)	0.5s	<b>0.9945</b>	<b>0.4891</b>	<b>0.9976</b>	<b>0.6895</b>	<b>0.5589</b>	0.3351
	1.0s	0.7858	0.2375	0.8818	0.2720	0.4128	0.1964		1.0s	<b>0.9931</b>	<b>0.4689</b>	<b>0.9957</b>	<b>0.5231</b>	<u>0.4595</u>	<b>0.3417</b>
	1.5s	0.7953	0.2406	0.8794	0.2570	0.4434	0.2802		1.5s	<b>0.9936</b>	<b>0.5185</b>	<b>0.9958</b>	<b>0.5445</b>	<u>0.4830</u>	<b>0.3440</b>
	2.0s	0.8024	0.2277	0.8919	0.2349	0.4086	0.2011		2.0s	<b>0.9960</b>	<b>0.6275</b>	<b>0.9980</b>	<b>0.6799</b>	<b>0.6160</b>	<b>0.3499</b>

# 06 결과 해석 및 기대효과

- 결과 해석 및 기대 효과

- 브로콜리단이 제안한 CATCH-RG는 6개의 시나리오로 구성된 24개의 데이터 셋에서, **강건하게 우수한 성능을 달성**
- CATCH-RG는 복잡한 전처리나 마스크 처리 없이 불완전한 데이터 셋에서 강건한 성능을 달성하여 실제 산업 현장에서 적용 가능성이 매우 높을 것으로 기대
  - ❖ 복잡한 전처리 과정 없이도 모델이 자체적으로 불완전 데이터에 강건하게 학습
  - ❖ 데이터 관리 역량이 부족한 중소기업에서도 **최소한의 리소스로 시스템을 운영·유지보수 가능**  
41
- CATCH-RG의 end-to-end의 이상탐지 구조는 실제 다양한 현장 상황(설비 정비, 라인 중단, 정전 등)에서 오경보 비율 감소를 통해 AI 기반 시스템에 대한 현장의 전반적인 신뢰도 향상 할 것으로 기대

# 06 참고

- 본 자료에 활용된 모든 코드 소스는 source code에서 확인하실 수 있습니다.

❖ **Github:** <https://github.com/jinnnju/CATCH-RG.git>



jinnnju / CATCH-RG

- 실험에 대한 윤리성을 준수하였음을 서약합니다.

# References

- [1] The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark, Liu et al. Neurips 2025
- [2] VUS: Effective and Efficient Accuracy Measures for Time-Series Anomaly Detection, Paul et al. preprint, 2025
- [3] Volume Under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection, Paparrizos et al. VLDB 2022
- [4] CATCH: Channel-Aware Multivariate Time Series Anomaly Detection via Frequency Patching, Wu et al. ICLR 2025
- [5] A review of irregular time series data handling with gated recurrent neural networks, Philip B et al. Neurocomputing, Volume 441, 2021
- [6] Time-IMM: A Dataset and Benchmark for Irregular Multimodal Multivariate Time Series, Chang et al. Neurips 2025
- [7] Rethinking Irregular Time Series Forecasting: A Simple yet Effective Baseline, Liu et al. preprint, 2025

# 감사합니다

