Transformer-based Anomaly Detection in Multivariate Time Series

Jiyoon Lee

2023 . 01 . 27

Introduction

발표자 소개

❖ **이지윤 (Jiyoon Lee)**

- Data Mining & Quality Analytics Lab
- Ph.D. Candidates (2018.03 ~ Present)

❖ **Research Interest**

- Explainable neural network using Attention mechanism & Bayesian neural network
- Anomaly Detection in Multivariate Time Series

❖ **Contact**

- Tel: +82-2-3290-3769
- E-mail: jiyoonlee@korea.ac.kr

Contents

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[2] Xu, J., Wu, H., Wang, J., & Long, M. (2021). Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642.
[3] Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284.

Contents

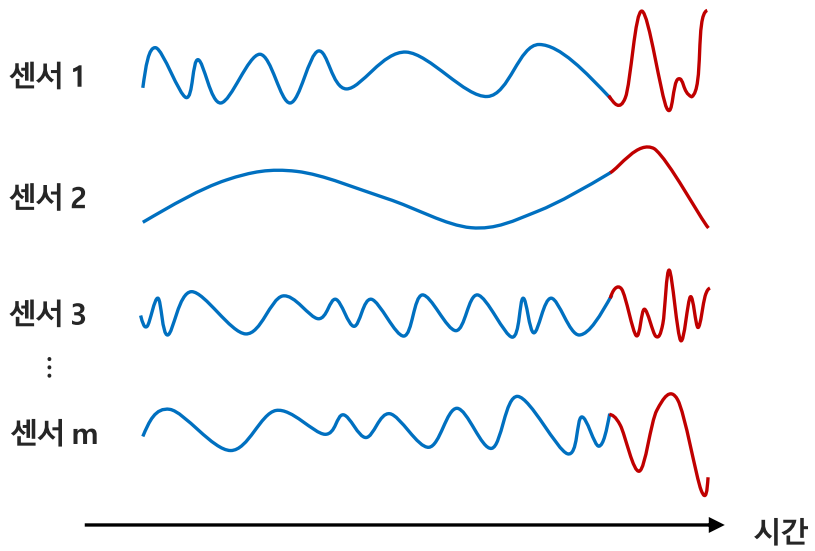
[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[2] Xu, J., Wu, H., Wang, J., & Long, M. (2021). Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642.
[3] Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284.

Introduction

Multivariate Time Series Dataset

❖ 다변량 시계열 데이터의 이상치 탐지

- 다변량 시계열 데이터: $\mathbb{X} = \{X_1, \ \dots, \ X_T\}, \ X_t = \{x_{1,t}, \ \dots, \ x_{m,t}\}$
- 이상치 탐지: 길이가 K인 time window $W_t = \{X_{t-K+1}, \ \dots, \ X_{t-1}, \ X_t\}$를 입력으로 각 시점의 이상치 여부를 판단



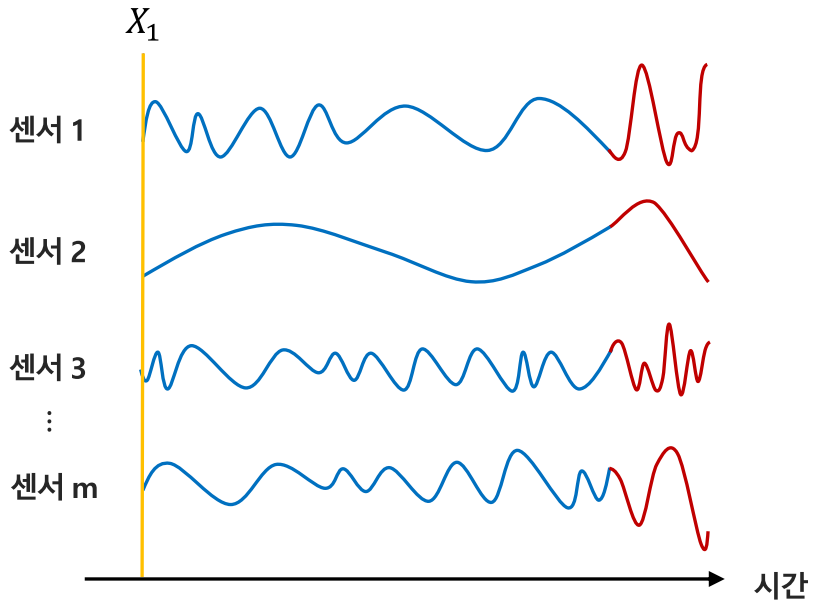
시간	센서 1	센서 2	센서 3	…	센서 m	Label
1	0.18	0.19	0.21	…	0.85	정상
2	0.21	0.21	0.23	…	0.95	정상
3	0.15	0.23	0.24	…	0.81	정상
4	0.13	0.21	0.22	…	0.82	정상
⋮	⋮	⋮	⋮	…	⋮	⋮
t-2	0.52	0.23	0.21	…	0.80	정상
t-1	0.83	0.21	0.14	…	1.92	불량
t	0.82	0.41	0.18	…	1.62	불량
t+1	0.85	0.35	0.17	…	1.98	불량
⋮	⋮	⋮	⋮	…	⋮	⋮
T	0.21	0.23	0.21	…	0.84	정상

Introduction

Multivariate Time Series Dataset

❖ 다변량 시계열 데이터의 이상치 탐지

- 다변량 시계열 데이터: $\mathbb{X} = \{X_1, \ \dots, \ X_T\}, \ X_t = \{x_{1,t}, \ \dots, \ x_{m,t}\}$
- 이상치 탐지: 길이가 K인 time window $W_t = \{X_{t-K+1}, \ \dots, \ X_{t-1}, \ X_t\}$를 입력으로 각 시점의 이상치 여부를 판단
  - 다변량 시계열 데이터가 지닌 순차성을 반영할 수 있음



X_1

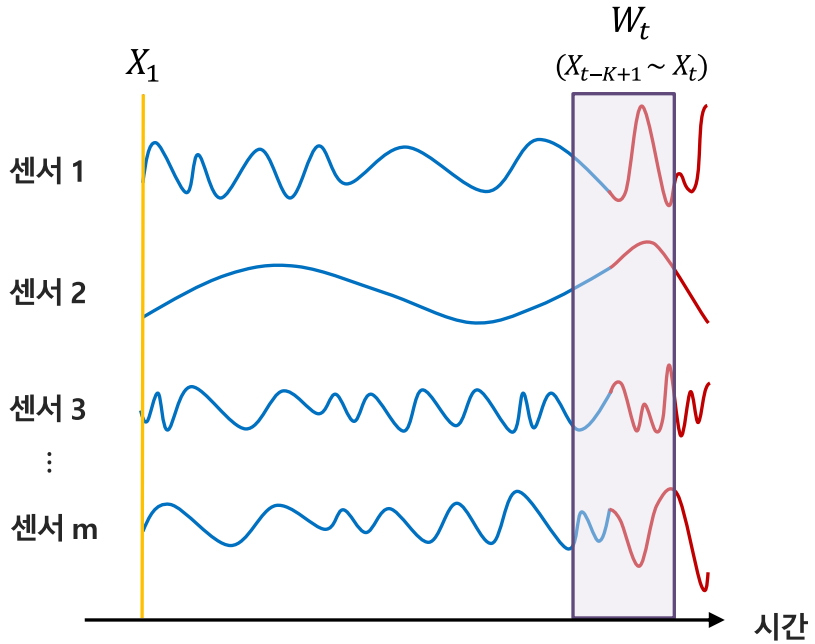
시간	센서 1	센서 2	센서 3	…	센서 m	Label
1	0.18	0.19	0.21	…	0.85	정상
2	0.21	0.21	0.23	…	0.95	정상
3	0.15	0.23	0.24	…	0.81	정상
4	0.13	0.21	0.22	…	0.82	정상
⋮	⋮	⋮	⋮	…	⋮	⋮
t-2	0.52	0.23	0.21	…	0.80	정상
t-1	0.83	0.21	0.14	…	1.92	불량
t	0.82	0.41	0.18	…	1.62	불량
t+1	0.85	0.35	0.17	…	1.98	불량
⋮	⋮	⋮	⋮	…	⋮	⋮
T	0.21	0.23	0.21	…	0.84	정상

Introduction

Multivariate Time Series Dataset

❖ 다변량 시계열 데이터의 이상치 탐지

- 다변량 시계열 데이터: $\mathbb{X} = \{X_1, \ \dots, \ X_T\}, \ X_t = \{x_{1,t}, \ \dots, \ x_{m,t}\}$
- 이상치 탐지: 길이가 K인 time window $W_t = \{X_{t-K+1}, \ \dots, \ X_{t-1}, \ X_t\}$를 입력으로 각 시점의 이상치 여부를 판단
  - 다변량 시계열 데이터가 지닌 순차성을 반영할 수 있음



W_t

시간	센서 1	센서 2	센서 3	…	센서 m	Label
1	0.18	0.19	0.21	…	0.85	정상
2	0.21	0.21	0.23	…	0.95	정상
3	0.15	0.23	0.24	…	0.81	정상
4	0.13	0.21	0.22	…	0.82	정상
⋮	⋮	⋮	⋮	…	⋮	⋮
t-2	0.52	0.23	0.21	…	0.80	정상
t-1	0.83	0.21	0.14	…	1.92	불량
t	0.82	0.41	0.18	…	1.62	불량
t+1	0.85	0.35	0.17	…	1.98	불량
⋮	⋮	⋮	⋮	…	⋮	⋮
T	0.21	0.23	0.21	…	0.84	정상

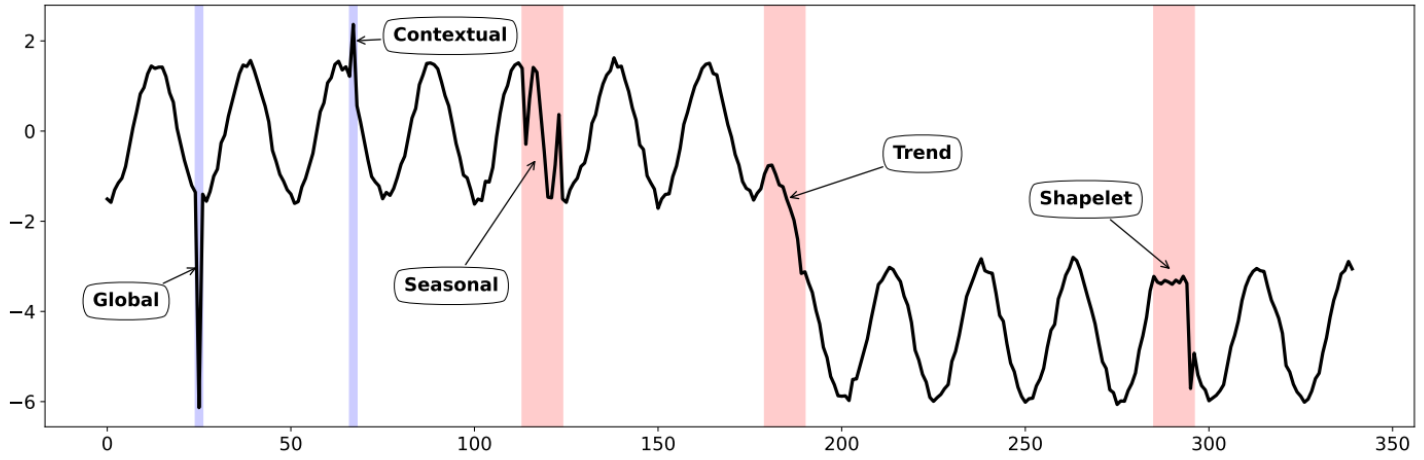
window size $K = 3$

Introduction

Multivariate Time Series Dataset

❖ 시계열 데이터에 대한 이상치 탐지 수행의 어려움

- 다양한 도메인에서 시계열 데이터 형태로 데이터가 수집되는 만큼 이상 유형이 다양
- 정상 및 불량을 명확히 구분하여 labeling 수행하는데 어려움이 있음
  - 연속적으로 수집되므로 정상과 이상이 경계가 모호함
  - 이상여부를 판별하기 위해서는 많은 양의 전문지식이 요구됨
  - 시간과 비용이 많이 소요됨
- 데이터가 지닌 시계열성과 다량의 변수사이 복잡한 관계를 반영할 수 있어야 함



Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2022). Deep Learning for Time Series Anomaly Detection: A Survey. *arXiv preprint arXiv:2211.05244*.

Introduction

Anomaly Detection in Time Series Data

❖ **시계열 관계를 반영하고, 고차원 데이터에 적합하도록 딥러닝 기반 방법론이 활발히 연구됨**

- [1] Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines (2021)
- [2] Deep Learning for Time Series Anomaly Detection: A Survey (2022)

Forecasting

Reconstruction

Hybrid

A[1]	MA[2]	Model	T/S[3]	Su/Un[4]	Input	Int[5]	P/S[6]	Stc[7]	Inc[8]
Forecasting	RNN (3.2.1)	LSTM-NDT [91]	T	Un	W	✓	Subseq		
		DeepLSTM [28]	T	Semi	P		Point		
		LSTM-PRED [65]	T	Un	W	✓	-		
		LGMAD [50]	T	Semi	P		Point		
		THOC [155]	T	Self	W		Subseq		
		AD-LTI [183]	T	Un	P		Point (frame)		
	CNN (3.2.2)	DeepAnt [133]	T	Un	W		Point + Subseq		
		TCN-ms [77]	T	Semi	W		Subseq		
	GNN (3.2.3)	GDN [46]	S	Un	W	✓	-		
		GTA* [35]	ST	Semi	-		-		
		GANF [42]	ST	Un	W				
	HTM (3.2.4)	RADM [49]	T	Un	W		-		
	Transformer (3.2.5)	SAND [159]	T	Semi	W		-		
		GTA* [35]	ST	Semi	-		-		
Reconstruction	AE (3.3.1)	AE/DAE [149]	T	Semi	P		Point		
		DAGMM [199]	S	Un	P		Point	✓	
		MSCRED [189]	ST	Un	W	✓	Subseq		
		USAD [10]	T	Un	W		Point		
		APAE [69]	T	Un	W		-		
		RANSynCoders [1]	ST	Un	P	✓	Point		✓
		CAE-Ensemble [22]	T	Un	W		Subseq		
		AMSL [194]	T	Self	W		-		
	VAE (3.3.2)	LSTM-VAE [140]	T	Semi	P		-		
		OmniAnomaly [162]	T	Un	W	✓	Point + Subseq	✓	
		STORN [158]	ST	Un	P		Point		
		GGM-VAE [73]	T	Un	W		Subseq		
		SISVAE [111]	T	Un	W		Point		
		VAE-GAN [135]	T	Semi	W		Point		
		VELC [188]	T	Un	-		-		
		TopoMAD [78]	ST	Un	W		Subseq	✓	
		PAD [31]	T	Un	W		Subseq		
		InterFusion [116]	ST	Un	W	✓	Subseq		
		MT-RVAE* [177]	ST	Un	W		-		
		RDSMM [112]	T	Un	W		Point + Subseq	✓	
	GAN (3.3.3)	MAD-GAN [110]	ST	Un	W		Subseq		
		BeatGAN [196]	T	Un	W		Subseq		
		DAEMON [34]	T	Un	W	✓	Subseq		
		FGANomaly [54]	T	Un	W		Point + Subseq		
		DCT-GAN* [113]	T	Un	W		-		
	Transformer (3.3.4)	Anomaly Transformer [185]	T	Un	W		Subseq		
		TranAD [171]	T	Un	W	✓	Subseq		
		DCT-GAN* [113]	T	Un	W		-		
		MT-RVAE* [177]	ST	Un	W		-		
Hybrid	AE (3.4.1)	CAE-M [193]	ST	Un	W		Subseq		
		NSIBF* [60]	T	Un	W		Subseq		
	RNN (3.4.2)	NSIBF* [60]	T	Un	W		Subseq		
		TAnoGAN [13]	T	Un	W		Subseq		
	GNN (3.4.3)	MTAD-GAT [195]	ST	Self	W	✓	Subseq		
		FuSAGNet [75]	ST	Semi	W		Subseq		

Reconstruction error · Prediction error · Dissimilarity

Autoencoder	
SPREAD (Gugulothu et al., 2018) [104]	●
S-RNNs (Kieu et al., 2019) [114]	●
LSTM-AE (Hsieh et al., 2019) [29]	●
MU-Net (Wen et al., 2019) [62]	●
MSCRED (Zhang et al., 2019) [51]	●
USAD (Audibert et al., 2020) [72]	●

GAN	
BeatGAN (Zhou et al., 2019) [115]	●
MAD-GAN (Li et al., 2019) [59]	●
WGAN-based (Choi et al., 2020) [53]	●
RSM-GAN (Khoshnevisan et al., 2020) [74]	●

VAE	
LSTM-VAE (Park et al., 2018) [112]	●
GGM-VAE (Guo et al., 2018) [113]	●
OmniAnomaly (Su et al., 2019) [68]	●

Transformer	
SAnD (Song et al., 2018) [119]	● ●
MTSM (Meng et al., 2019) [120]	●
GTA* (Chen et al., 2021) [109]	●

GNN	
MTAD-GAT (Zhao et al., 2020) [73]	● ●
GTA* (Chen et al., 2021) [109]	●
GDN (Deng et al., 2021) [108]	●

RNN	
LSTM-NDT (Hundman et al., 2018) [110]	●
LGMAD (Ding et al., 2019) [111]	●
THOC (Shen et al., 2020) [60]	●

TCN	
HS-TCN (Cheng et al., 2019) [116]	●
TCN-GMM (Liu et al., 2019) [117]	●
TCN-ms (He et al., 2019) [118]	●

HTM	
HTM-based (Wu et al., 2018) [121]	●
RADM (Ding et al., 2018) [69]	●

[1] Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. *IEEE Access*, *9*, 120043-120065.
[2] Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2022). Deep Learning for Time Series Anomaly Detection: A Survey. *arXiv e-prints*, arXiv-2211.

Introduction

Anomaly Detection in Time Series Data

❖ 시계열 관계를 반영하고, 고차원 데이터에 적합하도록 딥러닝 기반 방법론이 활발히 연구됨

- [1] Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines (2021)
- [2] Deep Learning for Time Series Anomaly Detection: A Survey (2022)

Forecasting

Reconstruction

Hybrid

A[1]	MA[2]	Model	T/S[3]	Su/Un[4]	Input	Int[5]	P/S[6]	Stc[7]	Inc[8]
Forecasting	RNN (3.2.1)	LSTM-NDT [91]	T	Un	W	✓	Subseq		
		DeepLSTM [28]	T	Semi	P		Point		
		LSTM-PRED [65]	T	Un	W	✓	-		
		LGMAD [50]	T	Semi	P		Point		
		THOC [155]	T	Self	W		Subseq		
		AD-LTI [183]	T	Un	P		Point (frame)		
	CNN (3.2.2)	DeepAnt [133]	T	Un	W		Point + Subseq		
		TCN-ms [77]	T	Semi	W		Subseq		
	GNN (3.2.3)	GDN [46]	S	Un	W	✓	-		
		GTA* [35]	ST	Semi	-		-		
		GANF [42]	ST	Un	W				
	HTM (3.2.4)	RADM [49]	T	Un	W		-		
	Transformer (3.2.5)	SAND [159]	T	Semi	W		-		
		GTA* [35]	ST	Semi	-		-		
Reconstruction	AE (3.3.1)	AE/DAE [149]	T	Semi	P		Point		
		DAGMM [199]	S	Un	P		Point	✓	
		MSCRED [189]	ST	Un	W	✓	Subseq		
		USAD [10]	T	Un	W		Point		
		APAE [69]	T	Un	W		-		
		RANSynCoders [1]	ST	Un	P	✓	Point		✓
		CAE-Ensemble [22]	T	Un	W		Subseq		
		AMSL [194]	T	Self	W		-		
	VAE (3.3.2)	LSTM-VAE [140]	T	Semi	P		-		
		OmniAnomaly [162]	T	Un	W	✓	Point + Subseq	✓	
		STORN [158]	ST	Un	P		Point		
		GGM-VAE [73]	T	Un	W		Subseq		
		SISVAE [111]	T	Un	W		Point		
		VAE-GAN [135]	T	Semi	W		Point		
		VELC [188]	T	Un	-		-		
		TopoMAD [78]	ST	Un	W		Subseq	✓	
		PAD [31]	T	Un	W		Subseq		
		InterFusion [116]	ST	Un	W	✓	Subseq		
		MT-RVAE* [177]	ST	Un	W		-		
		RDSMM [112]	T	Un	W		Point + Subseq	✓	
	GAN (3.3.3)	MAD-GAN [110]	ST	Un	W		Subseq		
		BeatGAN [196]	T	Un	W		Subseq		
		DAEMON [34]	T	Un	W	✓	Subseq		
		FGANomaly [54]	T	Un	W		Point + Subseq		
		DCT-GAN* [113]	T	Un	W		-		
	Transformer (3.3.4)	Anomaly Transformer [185]	T	Un	W		Subseq		
		TranAD [171]	T	Un	W	✓	Subseq		
		DCT-GAN* [113]	T	Un	W		-		
		MT-RVAE* [177]	ST	Un	W		-		
Hybrid	AE (3.4.1)	CAE-M [193]	ST	Un	W		Subseq		
		NSIBF* [60]	T	Un	W		Subseq		
	RNN (3.4.2)	NSIBF* [60]	T	Un	W		Subseq		
		TAnoGAN [13]	T	Un	W		Subseq		
	GNN (3.4.3)	MTAD-GAT [195]	ST	Self	W	✓	Subseq		
		FuSAGNet [75]	ST	Semi	W		Subseq		

Reconstruction error ● Prediction error ● Dissimilarity

Autoencoder	
SPREAD (Gugulothu et al., 2018) [104]	●
S-RNNs (Kieu et al., 2019) [114]	●
LSTM-AE (Hsieh et al., 2019) [29]	●
MU-Net (Wen et al., 2019) [62]	●
MSCRED (Zhang et al., 2019) [51]	●
USAD (Audibert et al., 2020) [72]	●

VAE	
LSTM-VAE (Park et al., 2018) [112]	●
GGM-VAE (Guo et al., 2018) [113]	●
OmniAnomaly (Su et al., 2019) [68]	●

RNN	
LSTM-NDT (Hundman et al., 2018) [110]	●
LGMAD (Ding et al., 2019) [111]	●
THOC (Shen et al., 2020) [60]	●

Transformer	
SAnD (Song et al., 2018) [119]	● ●
MTSM (Meng et al., 2019) [120]	●
GTA* (Chen et al., 2021) [109]	●

TCN	
HS-TCN (Cheng et al., 2019) [116]	●
TCN-GMM (Liu et al., 2019) [117]	●
TCN-ms (He et al., 2019) [118]	●

GAN	
BeatGAN (Zhou et al., 2019) [115]	●
MAD-GAN (Li et al., 2019) [59]	●
WGAN-based (Choi et al., 2020) [53]	●
RSM-GAN (Khoshnevisan et al., 2020) [74]	●

GNN	
MTAD-GAT (Zhao et al., 2020) [73]	● ●
GTA* (Chen et al., 2021) [109]	●
GDN (Deng et al., 2021) [108]	●

HTM	
HTM-based (Wu et al., 2018) [121]	●
RADM (Ding et al., 2018) [69]	●

Transformer (3.3.4)	Anomaly Transformer [185] ●	T	Un	W		Subseq
	TranAD [171] ●	T	Un	W	✓	Subseq
	DCT-GAN* [113]	T	Un	W		-
	MT-RVAE* [177]	ST	Un	W		-

[1] Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. *IEEE Access*, *9*, 120043-120065.
[2] Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2022). Deep Learning for Time Series Anomaly Detection: A Survey. *arXiv e-prints*, arXiv-2211.

Contents

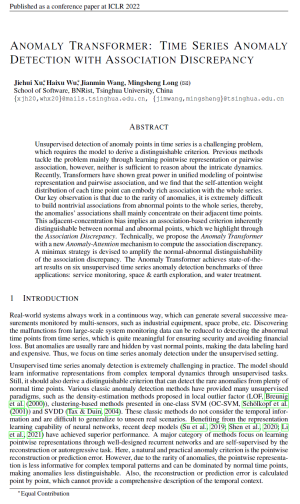
Anomaly Detection

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Lukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.



Published as a conference paper at ICLR 2022

ANOMALY TRANSFORMER: TIME SERIES ANOMALY DETECTION WITH ASSOCIATION DISCREPANCY

Jiehui Xu* Haixu Wu* Jianmin Wang, Mingsheng Long (✉)
School of Software, BNRist, Tsinghua University, China
{xjh20,whx20}@mails.tsinghua.edu.cn, {jimwang,mingsheng}@tsinghua.edu.cn

ABSTRACT

Unsupervised detection of anomaly points in time series is a challenging problem, which requires the model to derive a distinguishable criterion. Previous methods tackle the problem mainly through learning pointwise representation or pairwise association, however, neither is sufficient to reason about the intricate dynamics. Recently, Transformers have shown great power in unified modeling of pointwise representation and pairwise association, and we find that the self-attention weight distribution of each time point can embody rich association with the whole series. Our key observation is that due to the rarity of anomalies, it is extremely difficult to build nontrivial associations from abnormal points to the whole series, thereby, the anomalies' associations shall mainly concentrate on their adjacent time points. This adjacent-concentration bias implies an association-based criterion inherently distinguishable between normal and abnormal points, which we highlight through the *Association Discrepancy*. Technically, we propose the *Anomaly Transformer* with a new *Anomaly-Attention* mechanism to compute the association discrepancy. A minimax strategy is devised to amplify the normal-abnormal distinguishability of the association discrepancy. The Anomaly Transformer achieves state-of-the-art results on six unsupervised time series anomaly detection benchmarks of three applications: service monitoring, space & earth exploration, and water treatment.

1 INTRODUCTION

Real-world systems always work in a continuous way, which can generate several successive measurements monitored by multi-sensors, such as industrial equipment, space probe, etc. Discovering the malfunctions from large-scale system monitoring data can be reduced to detecting the abnormal time points from time series, which is quite meaningful for ensuring security and avoiding financial loss. But anomalies are usually rare and hidden by vast normal points, making the data labeling hard and expensive. Thus, we focus on time series anomaly detection under the unsupervised setting.

Unsupervised time series anomaly detection is extremely challenging in practice. The model should learn informative representations from complex temporal dynamics through unsupervised tasks. Still, it should also derive a distinguishable criterion that can detect the rare anomalies from plenty of normal time points. Various classic anomaly detection methods have provided many unsupervised paradigms, such as the density-estimation methods proposed in local outlier factor (LOF, Breunig et al. (2000)), clustering-based methods presented in one-class SVM (OC-SVM, Schölkopf et al. (2001)) and SVDD (Tax & Duin, 2004). These classic methods do not consider the temporal information and are difficult to generalize to unseen real scenarios. Benefiting from the representation learning capability of neural networks, recent deep models (Su et al., 2019; Shen et al., 2020; Li et al., 2021) have achieved superior performance. A major category of methods focus on learning pointwise representations through well-designed recurrent networks and are self-supervised by the reconstruction or autoregressive task. Here, a natural and practical anomaly criterion is the pointwise reconstruction or prediction error. However, due to the rarity of anomalies, the pointwise representation is less informative for complex temporal patterns and can be dominated by normal time points, making anomalies less distinguishable. Also, the reconstruction or prediction error is calculated point by point, which cannot provide a comprehensive description of the temporal context.

*Equal Contribution

TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data

Shreshth Tuli Imperial College London London, UK s.tuli20@imperial.ac.uk
Giuliano Casale Imperial College London London, UK g.casale@imperial.ac.uk
Nicholas R. Jennings Loughborough University London, UK n.r.jennings@lboro.ac.uk

ABSTRACT

Efficient anomaly detection and diagnosis in multivariate time-series data is of great importance for modern industrial applications. However, building a system that is able to quickly and accurately pinpoint anomalous observations is a challenging problem. This is due to the lack of anomaly labels, high data volatility and the demands of ultra-low inference times in modern applications. Despite the recent developments of deep learning approaches for anomaly detection, only a few of them can address all of these challenges. In this paper, we propose TranAD, a deep transformer network based anomaly detection and diagnosis model which uses attention-based sequence encoders to swiftly perform inference with the knowledge of the broader temporal trends in the data. TranAD uses focus score-based self-conditioning to enable robust multi-modal feature extraction and adversarial training to gain stability. Additionally, model-agnostic meta learning (MAML) allows us to train the model using limited data. Extensive empirical studies on six publicly available datasets demonstrate that TranAD can outperform state-of-the-art baseline methods in detection and diagnosis performance with data and time-efficient training. Specifically, TranAD increases F1 scores by up to 17%, reducing training times by up to 99% compared to the baselines.

PVLDB Reference Format:
Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. PVLDB, 15(6): 1201 - 1214, 2022.
doi:10.14778/3514061.3514067

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/imperial-qore/TranAD.

1 INTRODUCTION

Modern IT operations generate enormous amounts of high dimensional sensor data used for continuous monitoring and proper functioning of large-scale datasets. Traditionally, data mining experts have studied and highlighted data that do not follow usual trends to report faults. Such reports have been crucial for system management models for resilient fault tolerance and robust database design [47]. However, with the advent of big-data analytics and deep learning, this problem has become of interest to data mining

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 15, No. 6 ISSN 2150-8097.
doi:10.14778/3514061.3514067

researchers and to aid experts in handling increasing amounts of data. One particular use case is in artificial intelligence for Industry-4.0 databases, with a specific focus on service reliability [38] that has automated fault detection, recovery and management of modern systems. Detecting data faults, or any type of behavior not conforming to the expected trends, is an active research discipline referred to as anomaly detection in multivariate time series [11]. Many data-driven industries, including ones related to distributed computing, Internet of Things (IoT), robotics and urban resource management [4, 46] are now adopting machine learning based unsupervised methods for anomaly detection.

Challenges. The problem of anomaly detection is becoming increasingly challenging in large-scale databases due to the increasing data modality [18, 28, 54]. In particular, the increasing number of sensors and devices in contemporary IoT platforms with increasing data volatility creates the requirement for significant amounts of data for accurate inference. However, due to the rising federated learning paradigm with geographically distant clusters, synchronizing databases across devices is expensive, causing limited data availability for training [48, 57]. Further, next-generation applications need ultra-fast inference speeds for quick recovery and optimal Quality of Service (QoS) [6, 58]. Time-series databases are generated using several engineering artifacts (servers, robots, etc.) that interact with the environment, humans or other systems. As a result, the data often displays both stochastic and temporal trends [43]. It thus becomes crucial to distinguish outliers due to stochasticity and only pinpoint observations that do not adhere to the observed temporal trends. Moreover, the lack of labeled data and anomaly diversity makes the problem challenging as we cannot use supervised learning models, which have shown to be effective in other areas of data mining [15]. Firstly, it is not only important to detect anomalies but also the root causes, i.e., the specific data sources leading to abnormal behavior [23]. This complicates the problem further as we need to perform multi-class prediction (whether there is an anomaly and from which source if so) [60].

Existing solutions. The above discussed challenges have led to the development of a myriad of unsupervised learning solutions for automated anomaly detection. Researchers have developed reconstruction-based methods that predominantly aim to encapsulate the temporal trends and predict the time-series data in an unsupervised fashion, then use the deviation of the prediction with the ground-truth data as anomaly scores. Based on various extreme value analysis methods, such approaches classify timestamps with high anomaly scores as abnormal [4, 10, 14, 28, 29, 45, 60, 62]. The way prior works generate a predicted time-series from a given one varies from one work to another. Traditional approaches, like SAND [10], use clustering and statistical analysis to

Transformer **Anomaly Transformer** **Tran AD**

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[2] Xu, J., Wu, H., Wang, J., & Long, M. (2021). Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642.
[3] Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284.

Transformer

Introduction

❖ Attention is all you need (2017, NeurIPS)

- 2023년 1월 기준 62,992회 인용
- 후속 연구로 다양한 관점에서의 모델 구조 변형 및 응용 연구에 활용

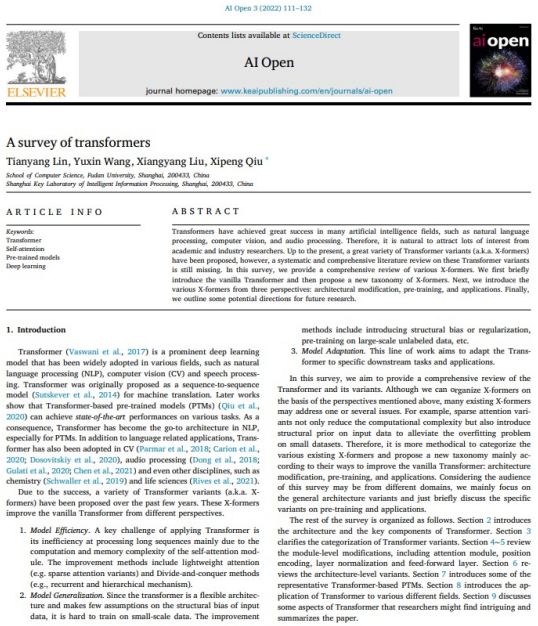
Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Transformer



AI Open 3 (2022) 111–132
Contents lists available at ScienceDirect
AI Open
journal homepage: www.keaipublishing.com/en/journals/ai-open

A survey of transformers

Tianyang Lin, Yuxin Wang, Xiangyang Liu, Xipeng Qiu *
School of Computer Science, Fudan University, Shanghai, 200433, China
Shanghai Key Laboratory of Intelligent Information Processing, Shanghai, 200433, China

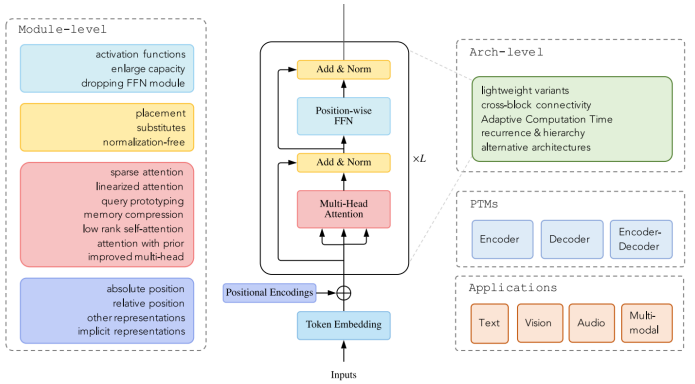
ARTICLE INFO

Keywords:
Transformer
Self-attention
Pre-trained models
Deep learning

ABSTRACT

Transformers have achieved great success in many artificial intelligence fields, such as natural language processing, computer vision, and audio processing. Therefore, it is natural to attract lots of interest from academic and industry researchers. Up to the present, a great variety of Transformer variants (a.k.a. X-formers) have been proposed, however, a systematic and comprehensive literature review on these Transformer variants is still missing. In this survey, we provide a comprehensive review of various X-formers. We first briefly introduce the vanilla Transformer and then propose a new taxonomy of X-formers. Next, we introduce the various X-formers from three perspectives: architectural modification, pre-training, and applications. Finally, we outline some potential directions for future research.

Survey of Transformer



Transformer Variants

Jay Alammar

Visualizing machine learning one concept at a time.
@JayAlammar on Twitter. YouTube Channel

Transformer Visualization

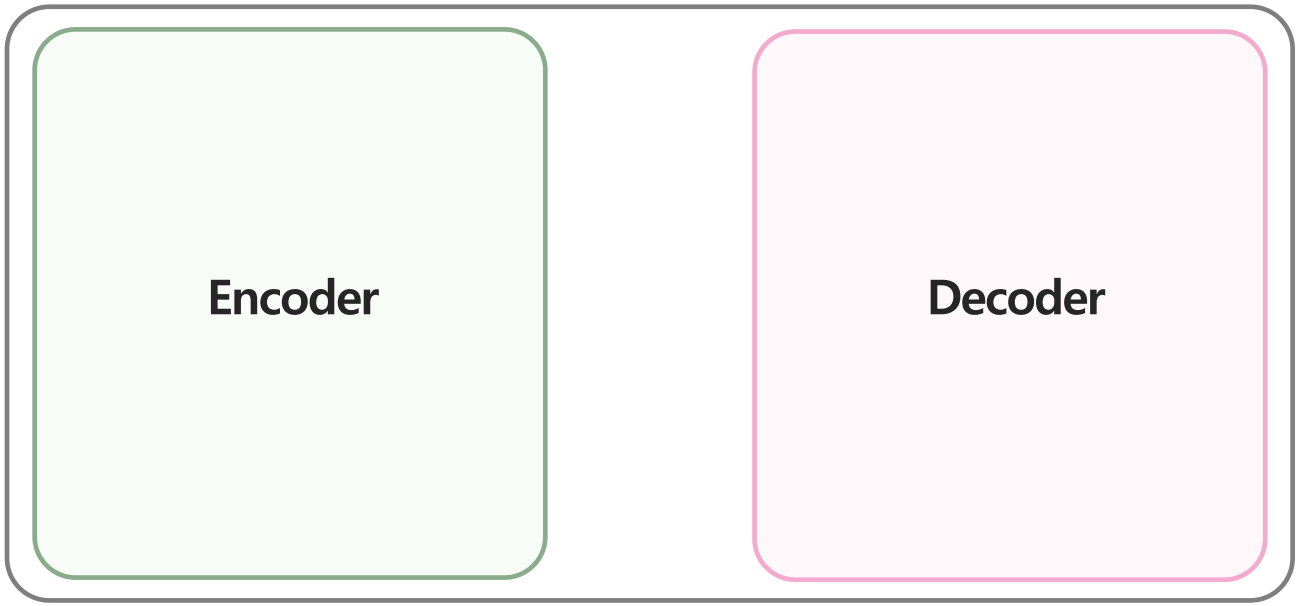
Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*.
https://jalammar.github.io/illustrated-transformer/

Transformer

Model Architecture

❖ **Transformer는 Encoder-Decoder 구조로 구성**

Output : I am a student



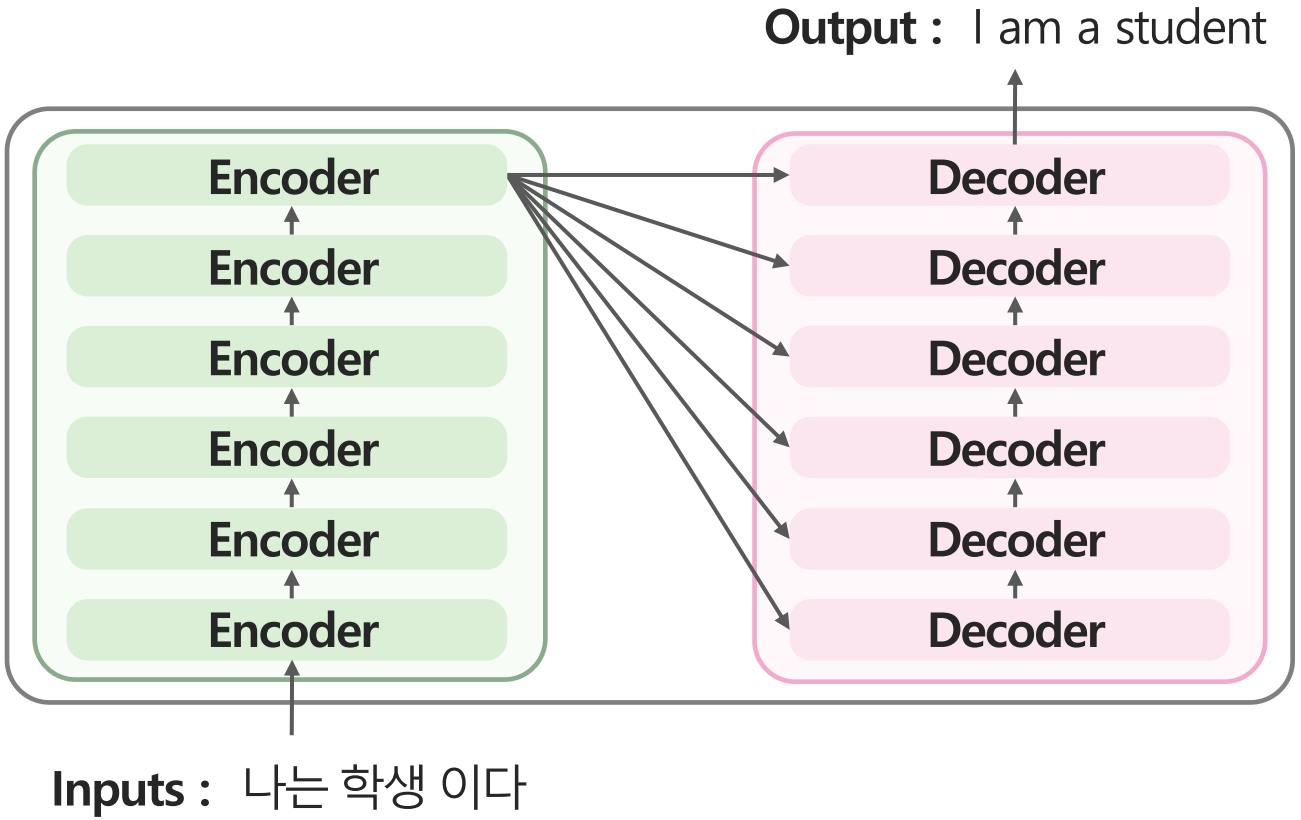
Inputs : 나는 학생 이다

Transformer

Model Architecture

❖ **Transformer는 Encoder-Decoder 구조로 구성**

- 각 N개의 모듈로 구성되며 구조는 동일

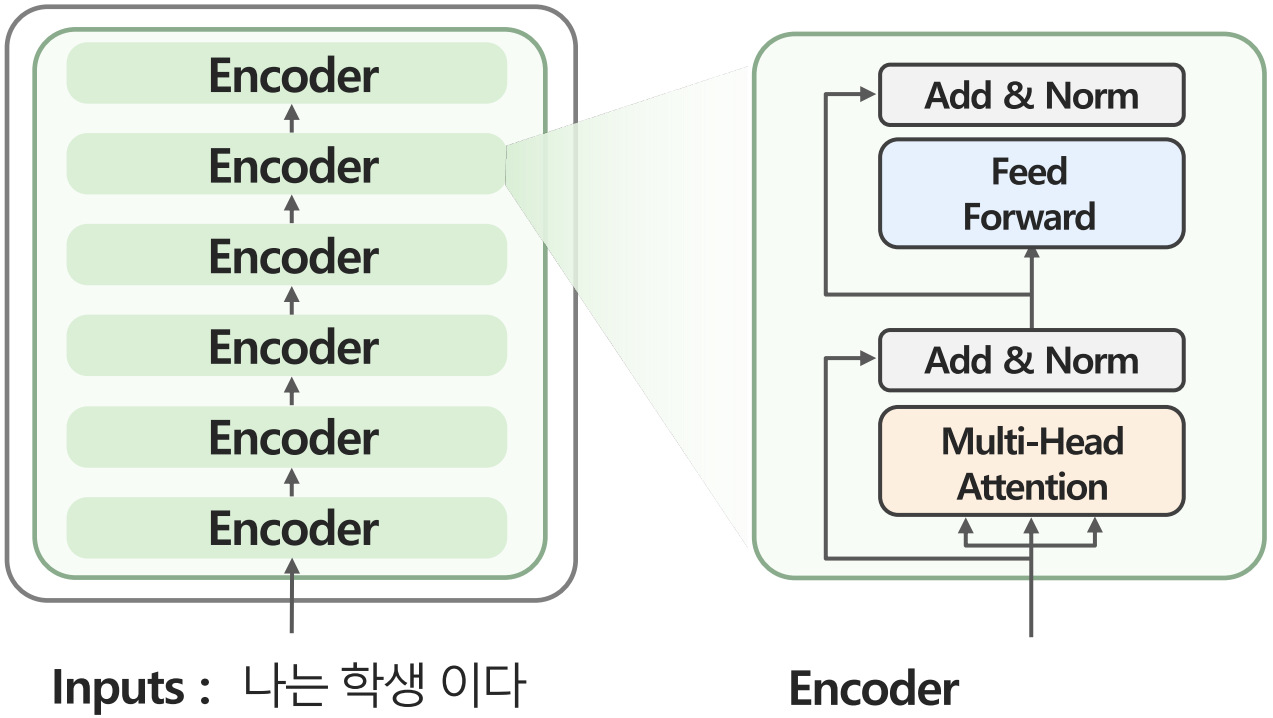


Transformer

Model Architecture

❖ **Transformer는 Encoder-Decoder 구조로 구성**

- Encoder는 (1) multi-head attention, (2) feed forward network로 구성

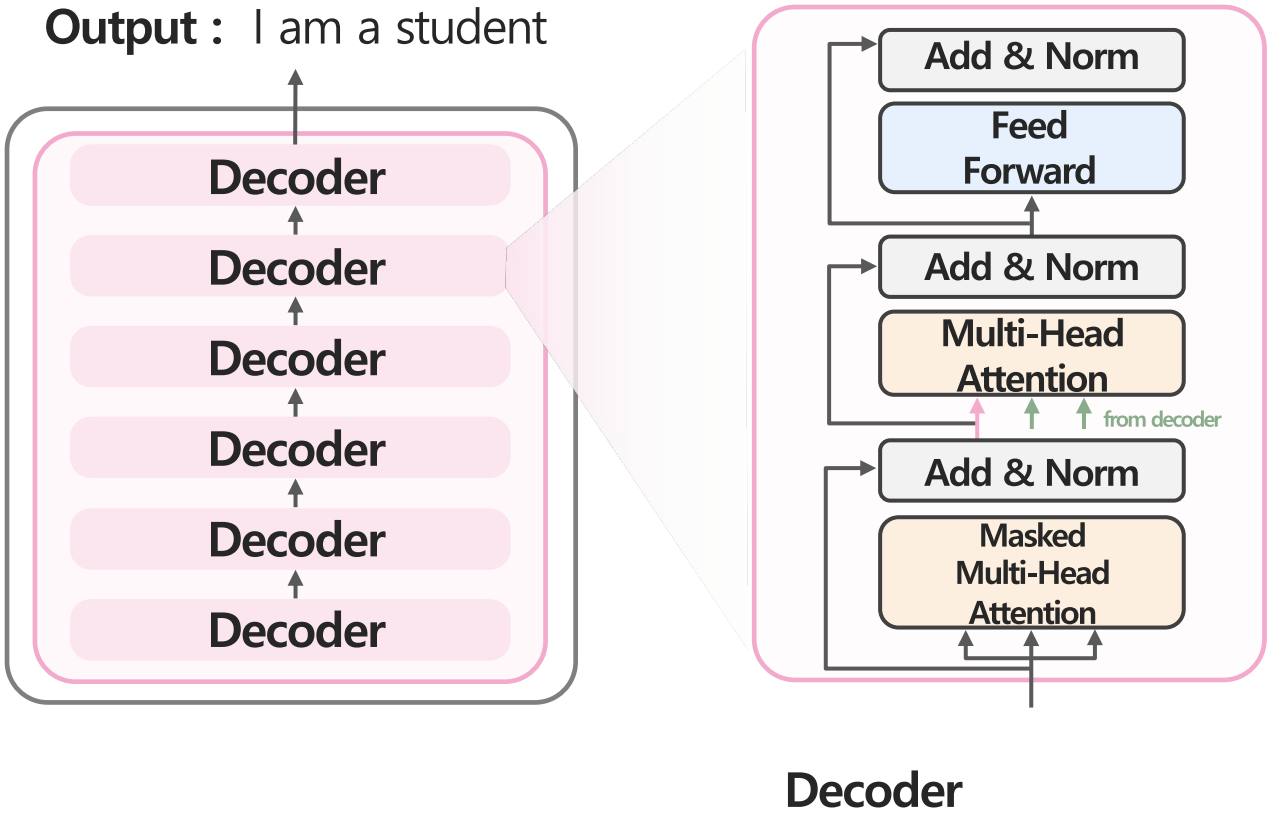


Transformer

Model Architecture

❖ **Transformer는 Encoder-Decoder 구조로 구성**

- Decoder는 (1) masked multi-head attention, (2) multi-head attention, (3) feed forward network로 구성

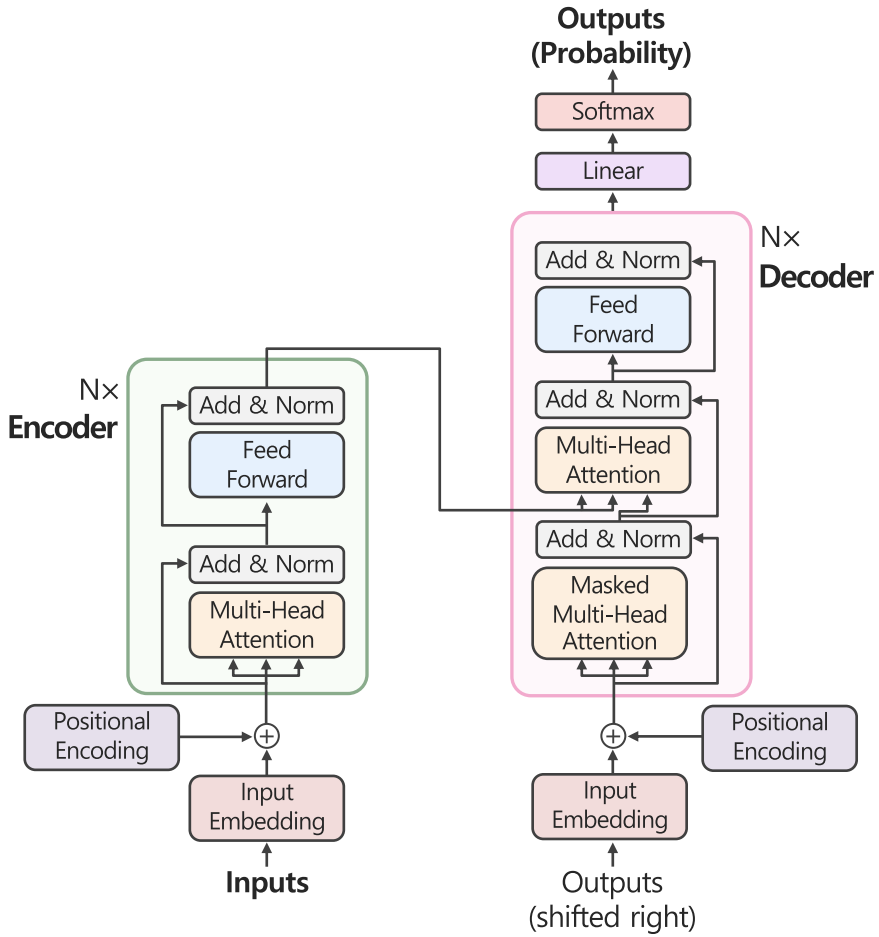


Decoder

Transformer

Model Architecture

❖ **Transformer는 Encoder-Decoder 구조로 구성**



① **Embedding**

② **Positional Encoding**

③ **Encoder**

- **Self-Attention**
- **Feed Forward**

④ **Decoder**

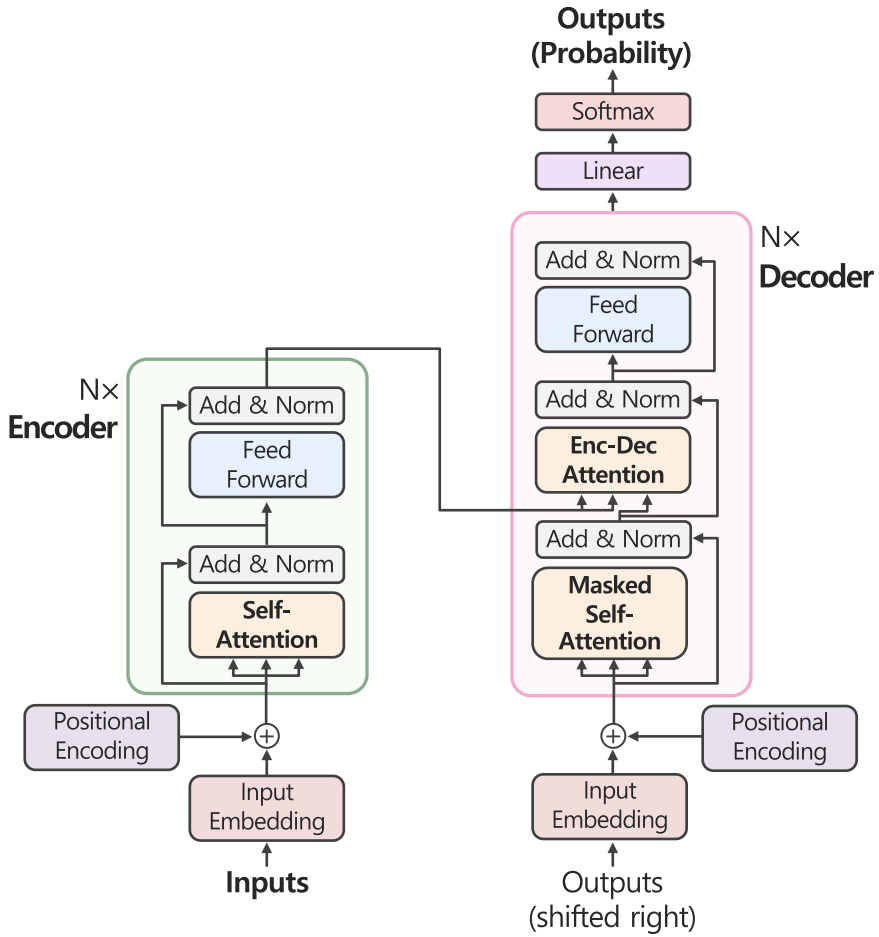
- **Masked Self-Attention**
- **Encoder-Decoder Attention**
- **Feed Forward**

⑤ **Prediction**

Transformer

Model Architecture

❖ Transformer는 Encoder-Decoder 구조로 구성



① Embedding

② Positional Encoding

③ Encoder

- Self-Attention
- Feed Forward

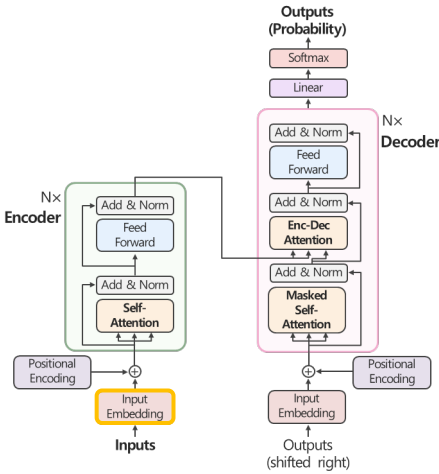
④ Decoder

- Masked Self-Attention
- Encoder-Decoder Attention
- Feed Forward

⑤ Prediction

Transformer

① Embedding



❖ **단어(token) 형태의 데이터를 수치로 변환**

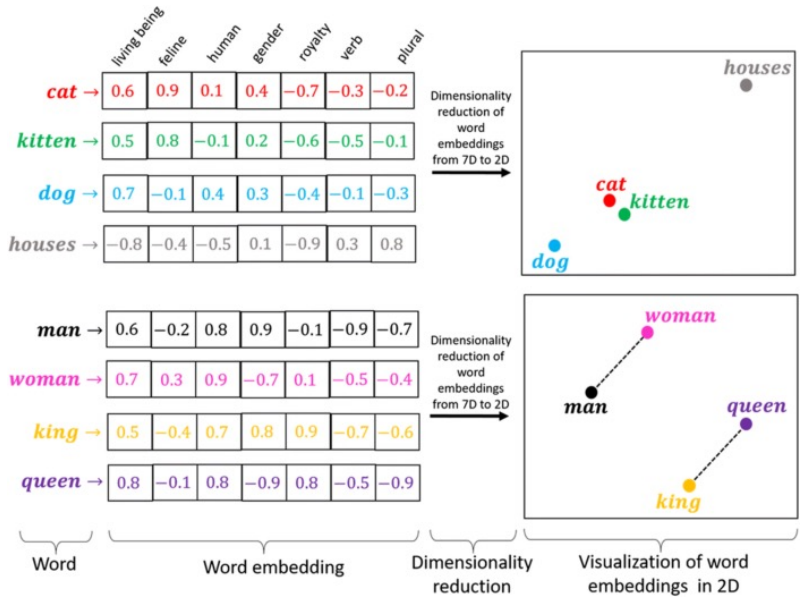
- 초기에는 one-hot vector형태로 입력되며, embedding layer를 통해 학습
- 유사 단어는 유사한 값을 지니도록 embedding 수행

Inputs : 나는 학생 이다

"나는"

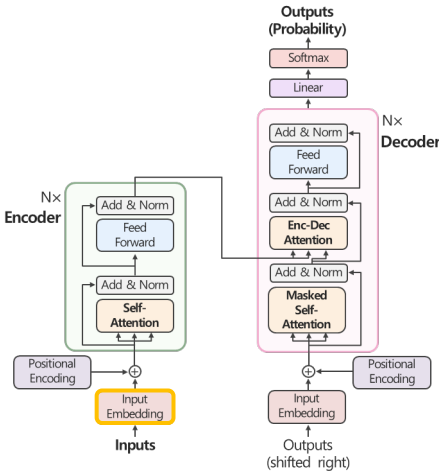
	나는	…	이다	…	학생	…	# of vocab
Initial Embedding	0	1	0	0	0	0	0	0	0

Word Embedding x_1 $[1, D_{embedding}]$



Transformer

① Embedding



❖ **단어(token) 형태의 데이터를 수치로 변환**

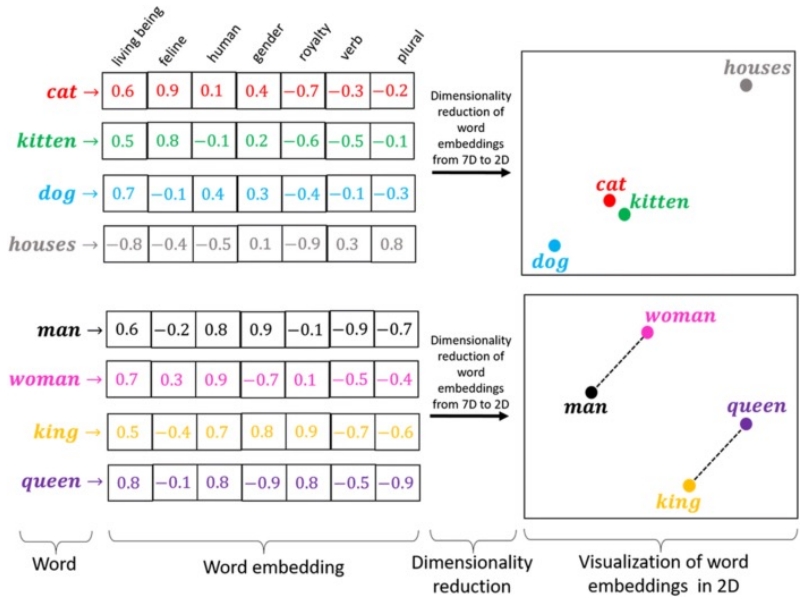
- 초기에는 one-hot vector형태로 입력되며, embedding layer를 통해 학습
- 유사 단어는 유사한 값을 지니도록 embedding 수행

Inputs : 나는 학생 이다

"학생"

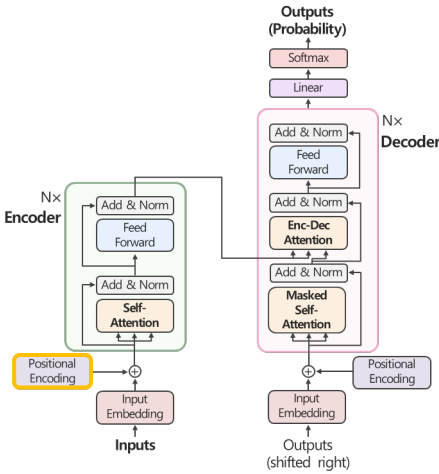
	나는	…	이다	…	학생	…	# of vocab
Initial Embedding	0	0	0	0	0	0	1	0	0

Word Embedding x_2 $[1, D_{embedding}]$



Transformer

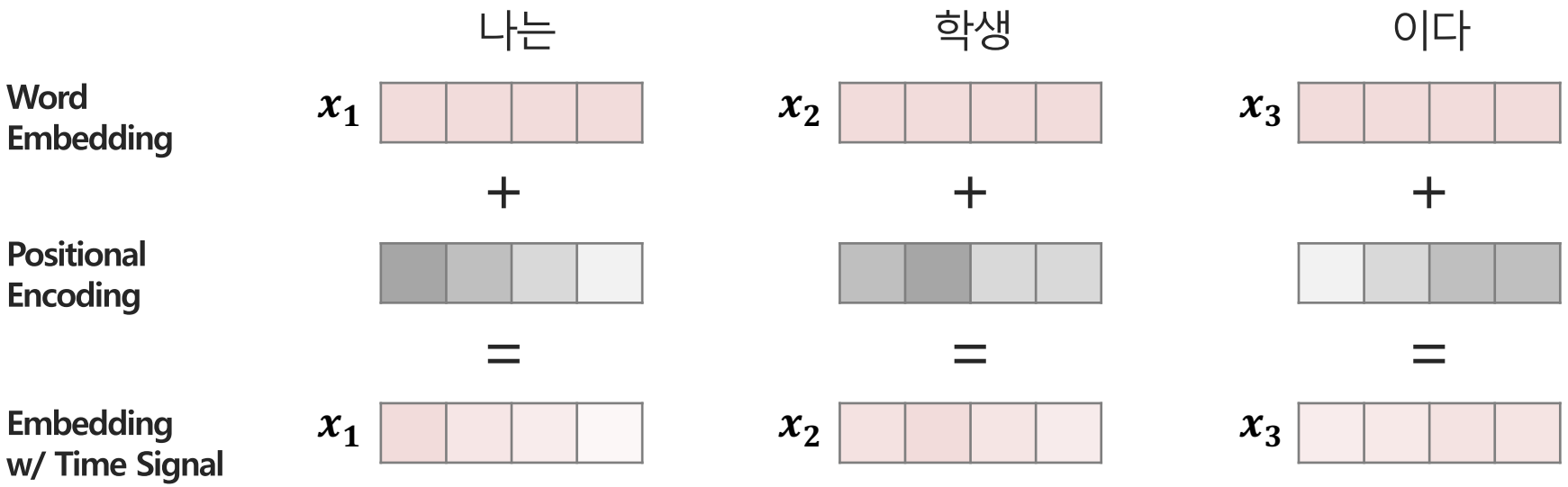
② Positional Encoding



❖ **단어 사이 순차성을 반영하기 위한 기법**

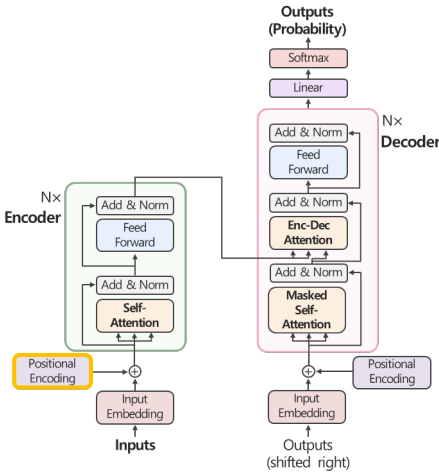
- RNN계열 방법론과는 달리 입력값을 순차적으로 처리하지 않음 (병렬적 처리)
- 순차성을 부여하고 -1~1사이 범위를 갖도록 삼각함수 활용

Inputs : 나는 학생 이다



Transformer

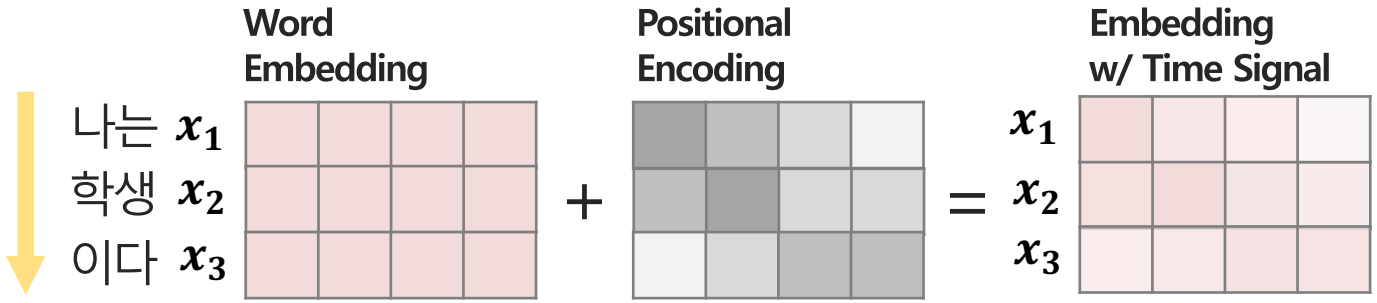
② Positional Encoding



❖ **단어 사이 순차성을 반영하기 위한 기법**

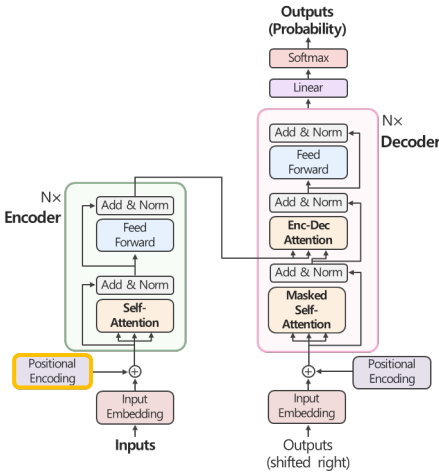
- RNN계열 방법론과는 달리 입력값을 순차적으로 처리하지 않음 (병렬적 처리)
- 순차성을 부여하고 -1~1사이 범위를 갖도록 삼각함수 활용

Inputs : 나는 학생 이다



Transformer

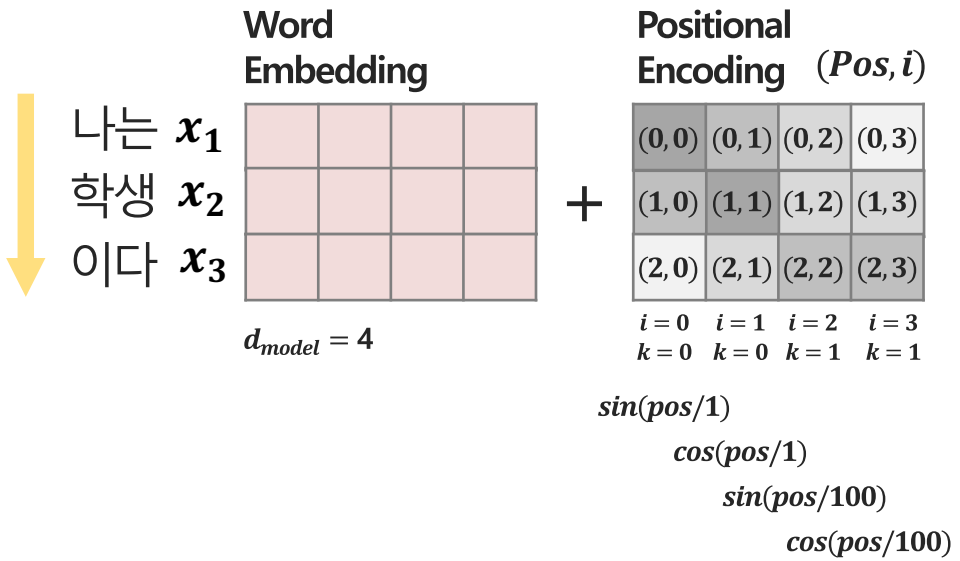
② Positional Encoding



❖ **단어 사이 순차성을 반영하기 위한 기법**

- RNN계열 방법론과는 달리 입력값을 순차적으로 처리하지 않음 (병렬적 처리)
- 순차성을 부여하고 -1~1사이 범위를 갖도록 삼각함수 활용

Inputs : 나는 학생 이다



$$PE_{(pos,i)} = sin\left(\frac{pos}{10000^{2k/d_{model}}}\right) \quad , \; if \;\; i = 2k$$

차원의 순서가 짝수일 때

$$PE_{(pos,i)} = cos\left(\frac{pos}{10000^{2k/d_{model}}}\right) \quad , \; if \;\; i = 2k+1$$

차원의 순서가 홀수일 때

d_{model} : 출력 차원 (embedding vector 차원)

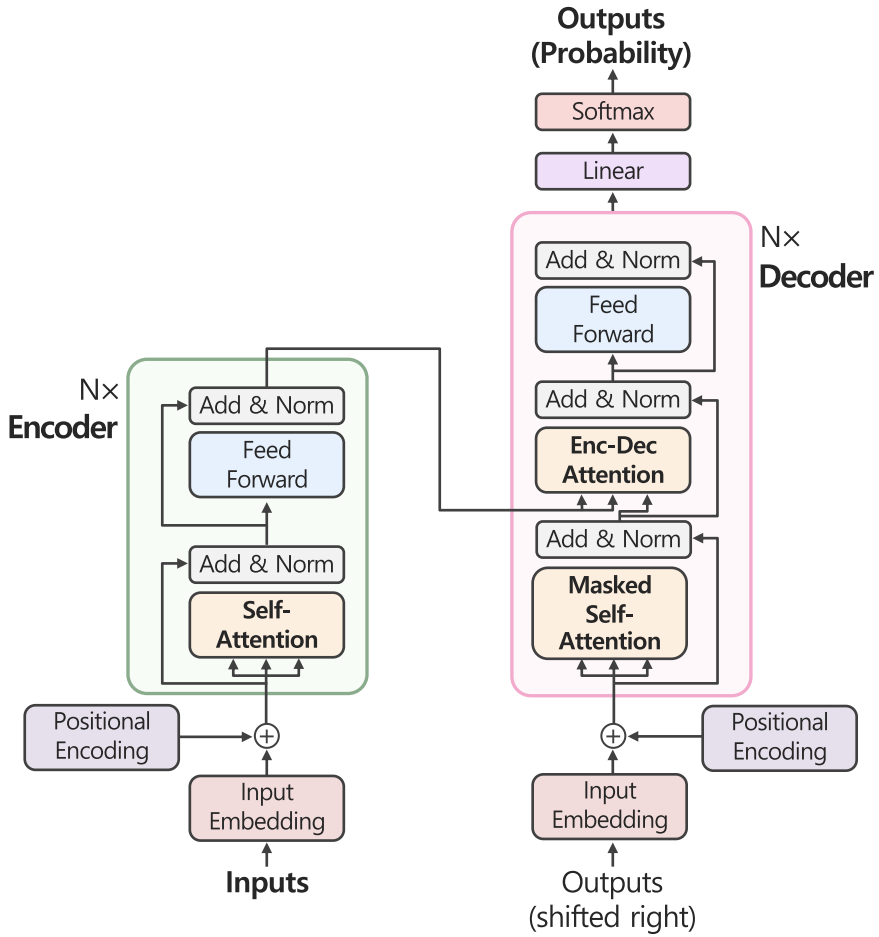
pos : 입력 시퀀스 데이터에서의 embedding vector 위치

i : embedding vector 내 차원의 순서

Transformer

Model Architecture

❖ Transformer는 Encoder-Decoder 구조로 구성



① Embedding

② Positional Encoding

③ Encoder

- Self-Attention
- Feed Forward

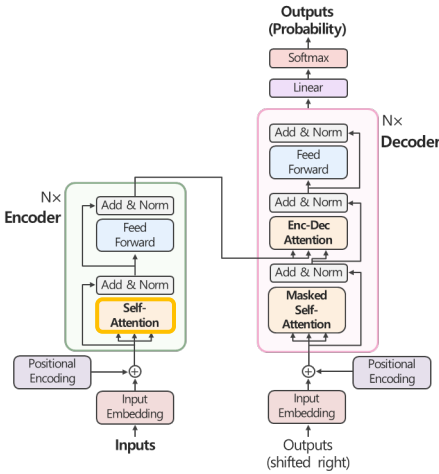
④ Decoder

- Masked Self-Attention
- Encoder-Decoder Attention
- Feed Forward

⑤ Prediction

Transformer

③ Encoder : Self-Attention



❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- 각 단어에 해당되는 Key, Query, Value matrix 학습

나는 학생 이다

Embedding w/ Time Signal x_1 (1×4) x_2 x_3

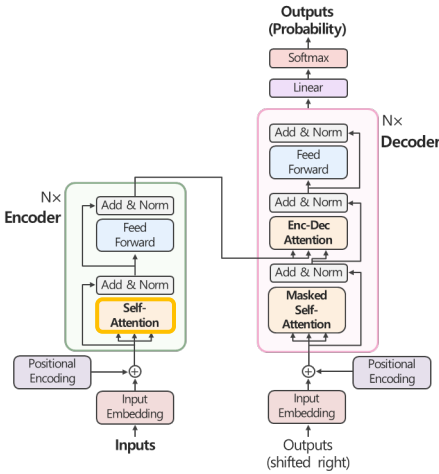
Queries - Encoder q_1 (1×2) q_2 q_3

(4×2)

W_E^Q W_E^K W_E^V

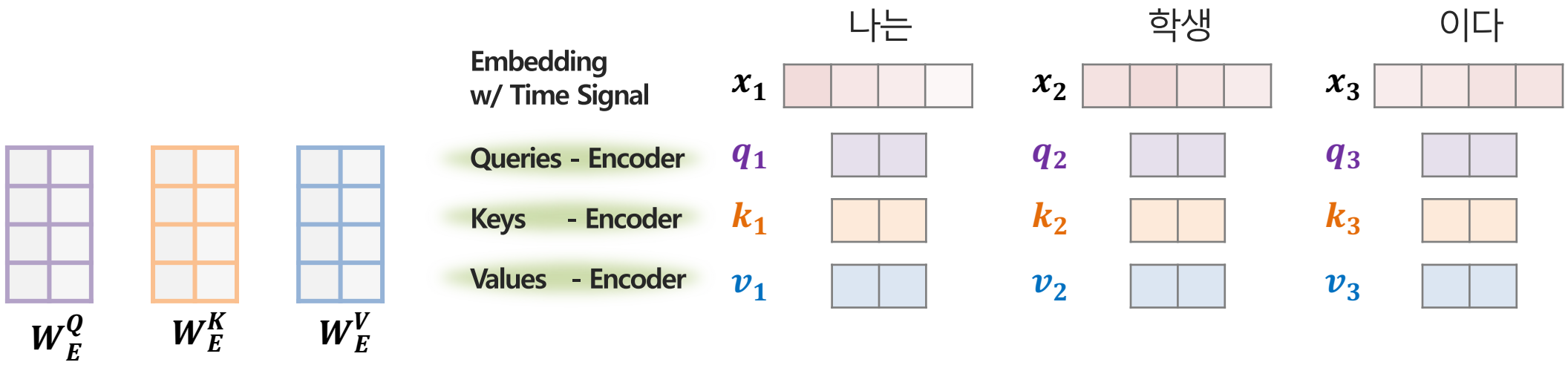
Transformer

③ Encoder : Self-Attention



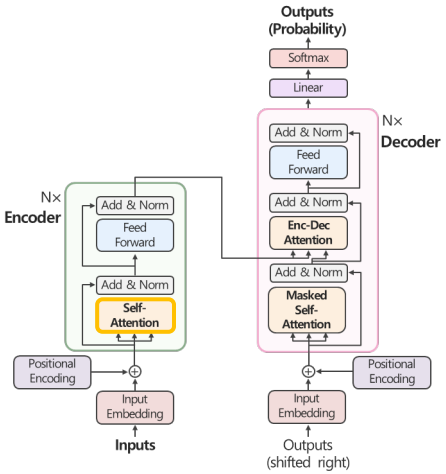
❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- 각 단어에 해당되는 Key, Query, Value matrix 학습



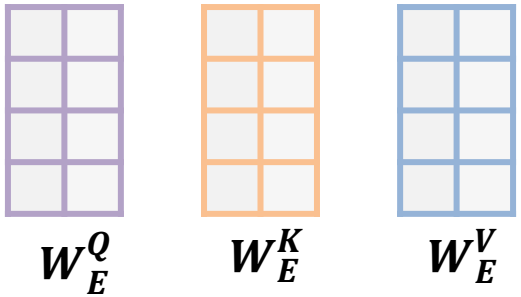
Transformer

③ Encoder : Self-Attention



❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- 현시점의 Query vector와 모든 Key vectors와 비교하여 유사도 산출



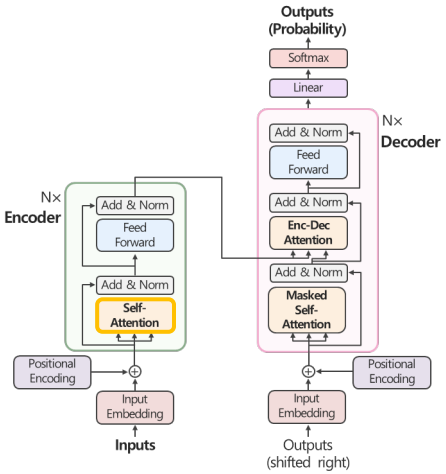
W_E^Q W_E^K W_E^V

$$d_k = \frac{d_{model}\,(512)}{num\ heads\ (8)} = 64$$

	나는	학생	이다
Embedding w/ Time Signal	x_1	x_2	x_3
Queries - Encoder	q_1	q_2	q_3
Keys - Encoder	k_1	k_2	k_3
Values - Encoder	v_1	v_2	v_3
Score	$q_1 \cdot k_1 = 136$	$q_1 \cdot k_2 = 116$	$q_1 \cdot k_3 = 64$
Divide by 8 ($\sqrt{d_k}$)	136 / 8 = 17	116 / 8 = 14	64 / 8 = 8
Softmax	0.952	0.047	0.000
Softmax × Value	v'_1	v'_2	v'_3
Summation	z_1		

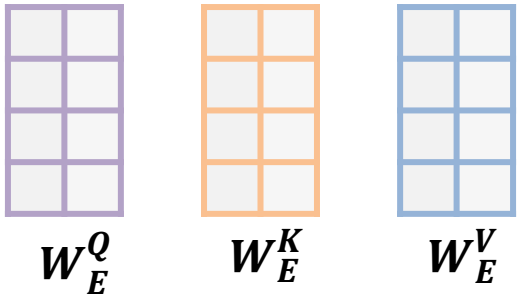
Transformer

③ Encoder : Self-Attention



❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- 현시점의 Query vector와 모든 Key vectors와 비교하여 유사도 산출

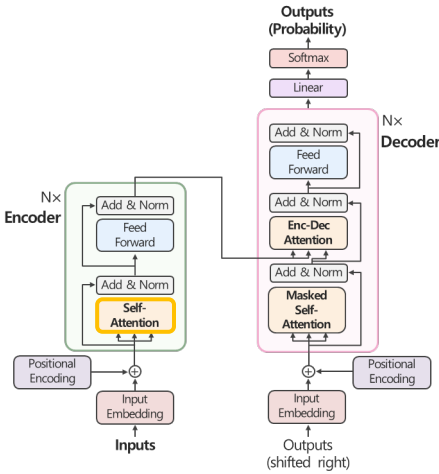


$$d_k = \frac{d_{model}\,(512)}{num\ heads\ (8)} = 64$$

	나는	학생	이다
Embedding w/ Time Signal	x_1	x_2	x_3
Queries - Encoder	q_1	q_2	q_3
Keys - Encoder	k_1	k_2	k_3
Values - Encoder	v_1	v_2	v_3
Score	$q_2 \cdot k_1 = 125$	$q_2 \cdot k_2 = 149$	$q_2 \cdot k_3 = 71$
Divide by 8 ($\sqrt{d_k}$)	125 / 8 = 16	149 / 8 = 19	71 / 8 = 9
Softmax	0.047	0.953	0.000
Softmax × Value	v'_1	v'_2	v'_3
Summation		z_2	

Transformer

③ Encoder : Self-Attention



❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- 실제 연산은 matrix 단위에서 수행

$$X \times W_E^Q = Q$$

나는 x_1
학생 x_2
이다 x_3

(3×4) × (4×2) = (3×2)

$$X \times W_E^K = K$$

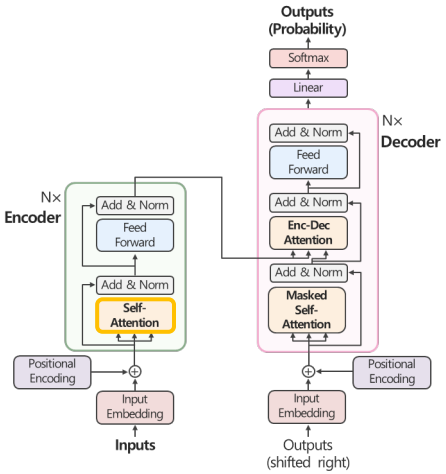
나는 x_1
학생 x_2
이다 x_3

$$X \times W_E^V = V$$

나는 x_1
학생 x_2
이다 x_3

Transformer

③ Encoder : Self-Attention



❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- 실제 연산은 matrix 단위에서 수행

나는 x_1
학생 x_2
이다 x_3

X (3×4) × W_E^Q (4×2) = Q (3×2)

X (3×4) × W_E^K (4×2) = K (3×2)

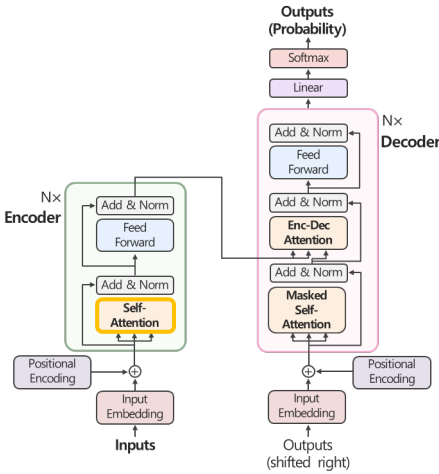
X (3×4) × W_E^V (4×2) = V (3×2)

$$\text{Softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) = \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix}$$

Attention score

Transformer

③ Encoder : Self-Attention



❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- 실제 연산은 matrix 단위에서 수행

$$X \times W_E^Q = Q$$

나는 x_1
학생 x_2
이다 x_3

(3×4) × (4×2) = (3×2)

$$X \times W_E^K = K$$

나는 x_1
학생 x_2
이다 x_3

$$X \times W_E^V = V$$

나는 x_1
학생 x_2
이다 x_3

$$\text{Softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) = \text{Attention score}$$

Q: q_1, q_2, q_3 × K^T: k_1 k_2 k_3

$$\text{Attention score} \times V = Z$$

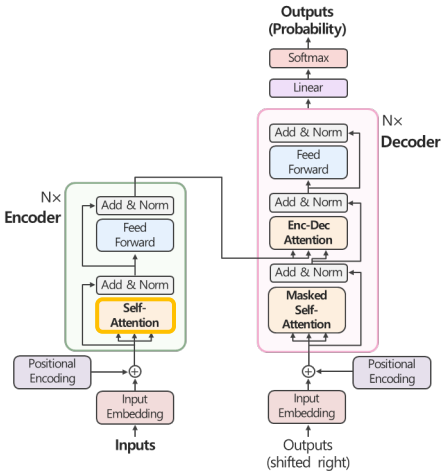
Attention score: s_1, s_2, s_3 × V: x_1, x_2, x_3 = Z

Transformer

③ Encoder : Self-Attention

❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- 서로 다른 weight를 사용하는 multi-head구조로 확장



X (나는 x_1, 학생 x_2, 이다 x_3) × W_E^Q = Q (# of heads)

X (나는 x_1, 학생 x_2, 이다 x_3) × W_E^K = K

X (나는 x_1, 학생 x_2, 이다 x_3) × W_E^V = V

$$\text{Softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) = \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix}$$

Attention score w/ multi-head

$$\begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} \times V = Z$$

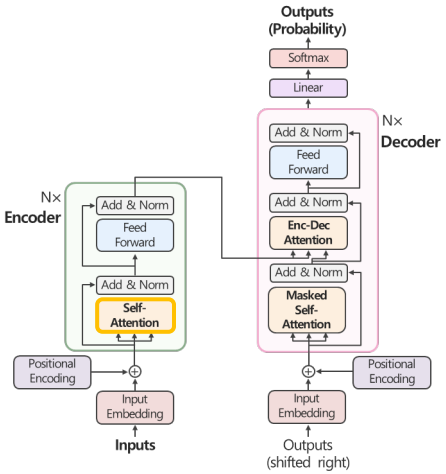
Attention score w/ multi-head × V = Z (# of heads)

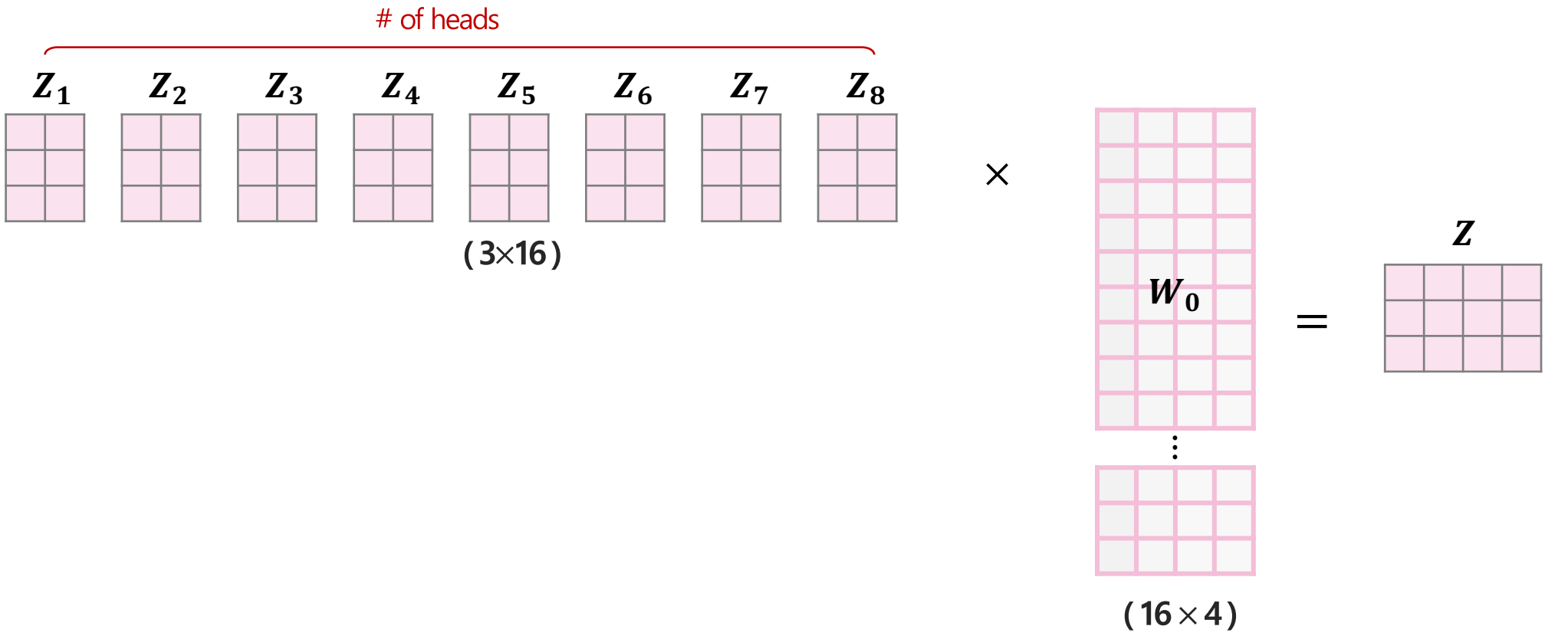
Transformer

③ Encoder : Self-Attention

❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

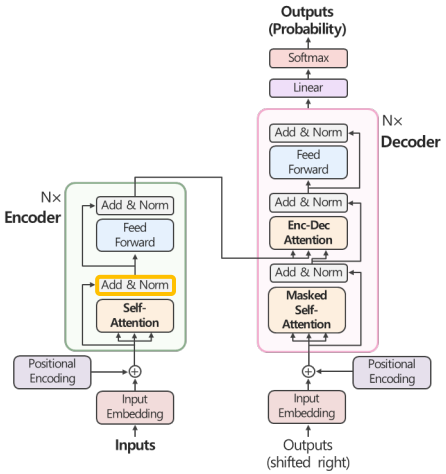
- 서로 다른 weight를 사용하는 multi-head구조로 확장





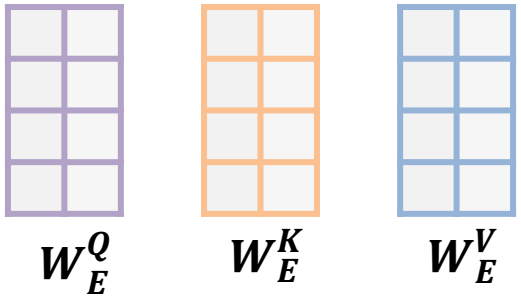
Transformer

③ Encoder : Self-Attention

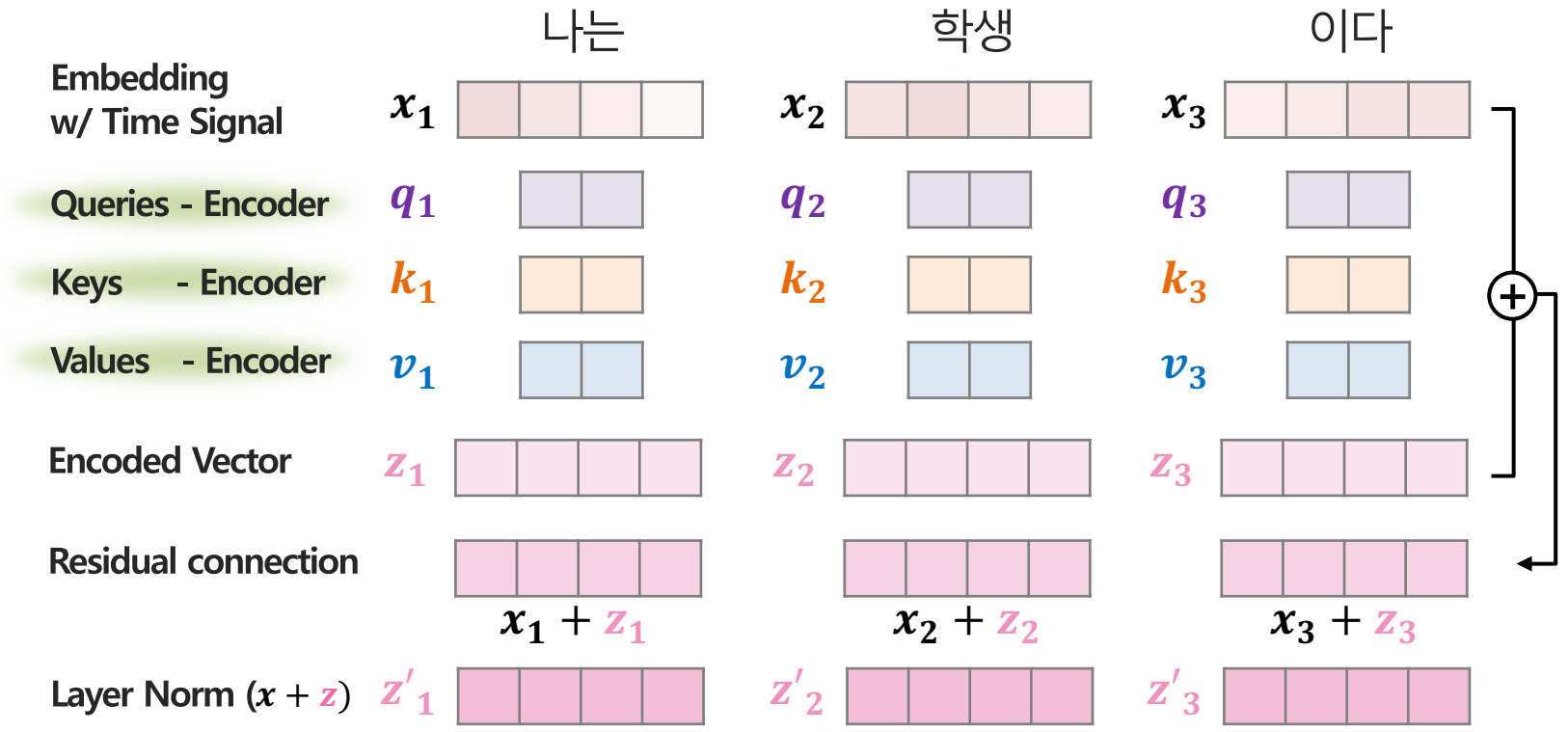


❖ **입력 값을 구성하는 모든 단어사이 관계를 비교하고 특징을 추출하여 z 도출**

- Residual connection을 통해 입력값의 정보를 직접적으로 반영

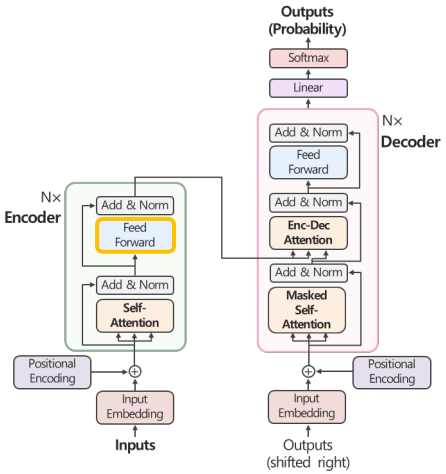


$$d_k = \frac{d_{model}\,(512)}{num\ heads\ (8)} = 64$$



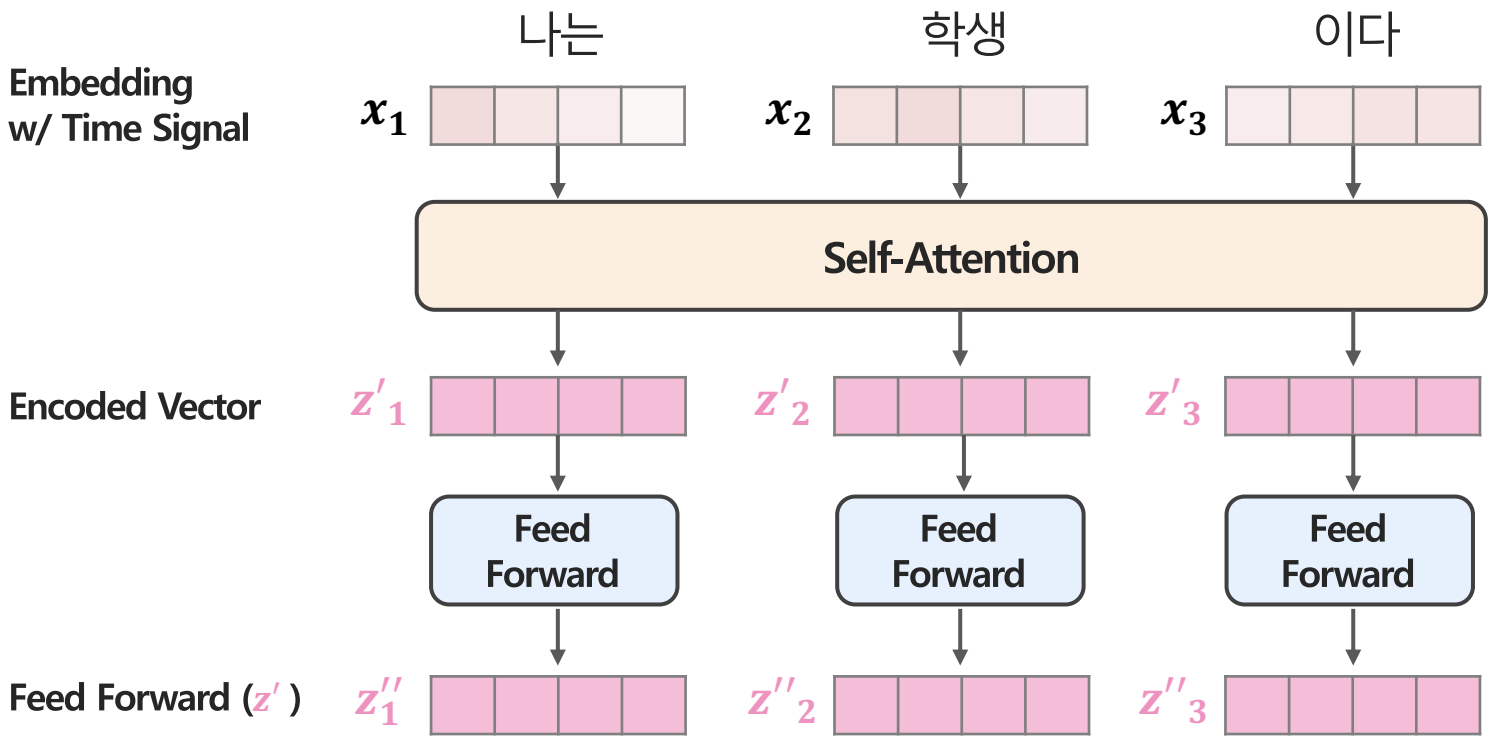
Transformer

③ Encoder : Feed Forward Network



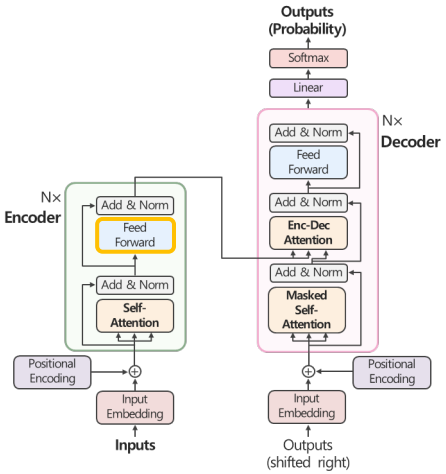
❖ **각 단어에 대응되는 값들에 개별적으로 FFN를 적용하여 비선형성 부여**

- 동일한 encoder에서는 parameter sharing을 통해 연산 효율성 확보



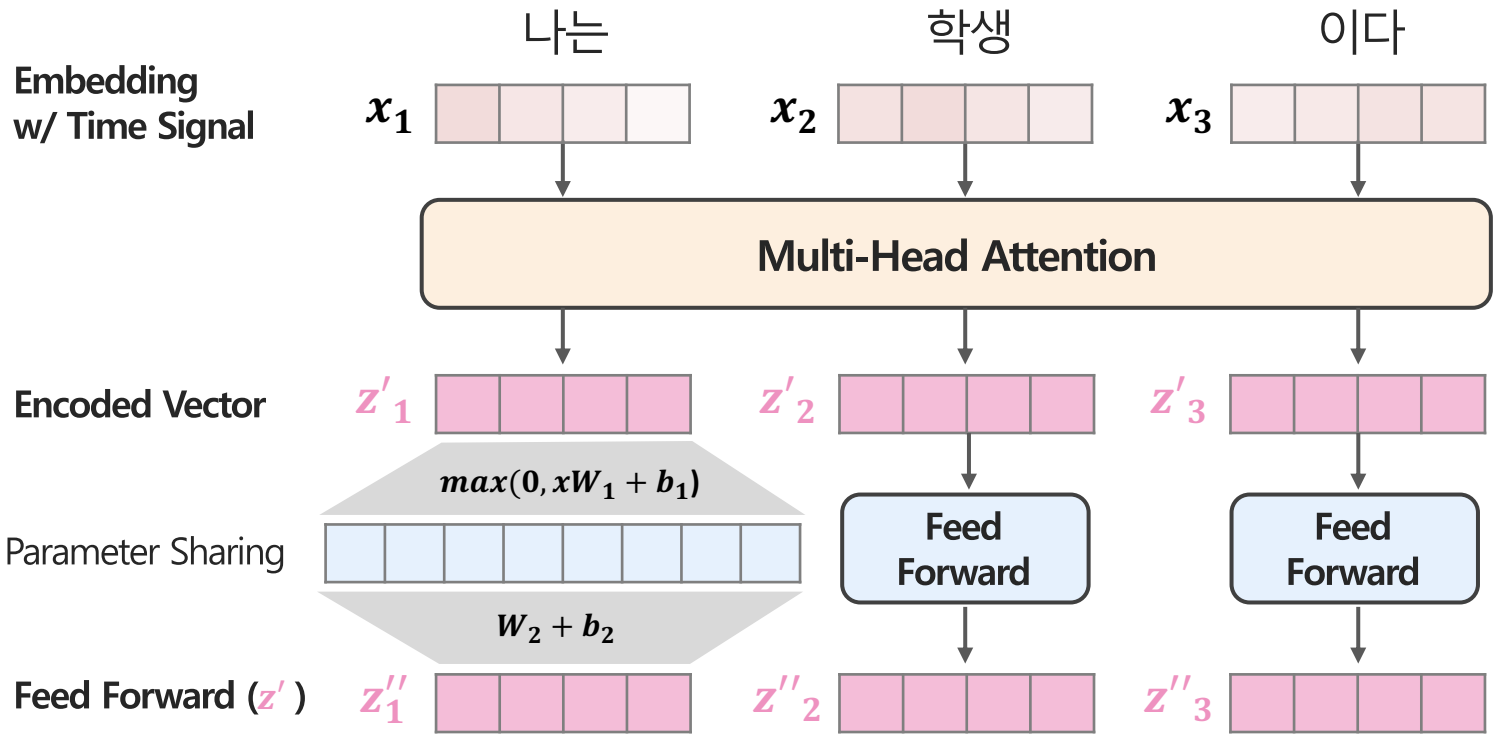
Transformer

③ Encoder : Feed Forward Network



❖ **각 단어에 대응되는 값들에 개별적으로 FFN를 적용하여 비선형성 부여**

- 동일한 encoder에서는 parameter sharing을 통해 연산 효율성 확보

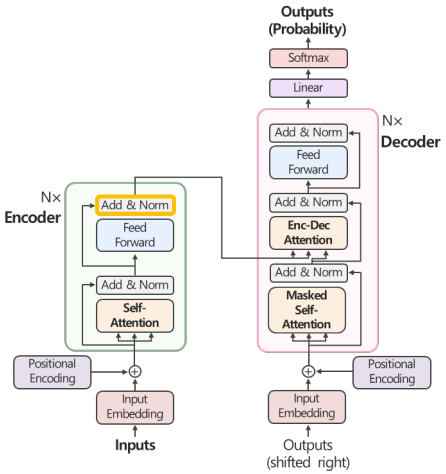


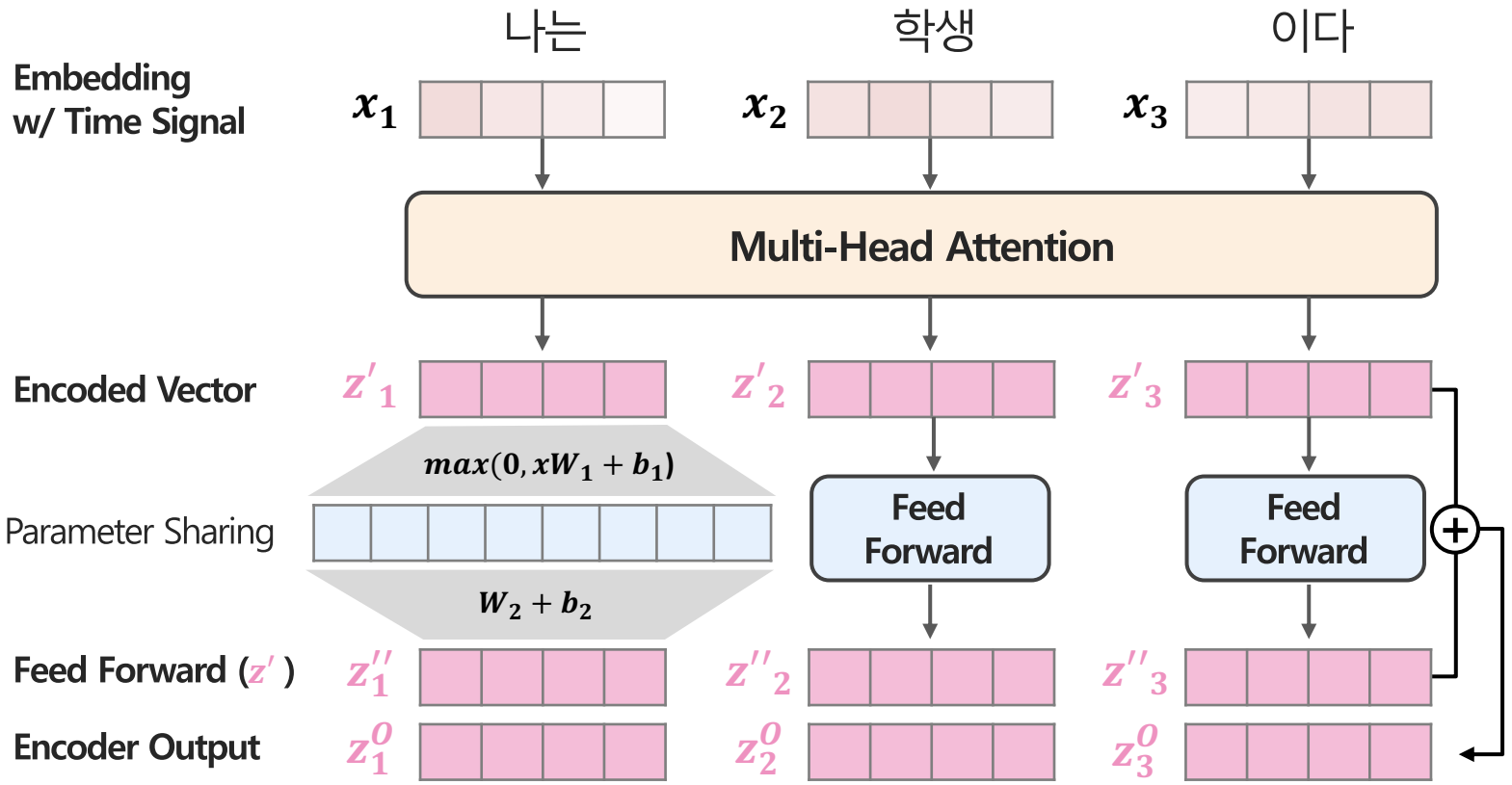
Transformer

③ Encoder : Feed Forward Network

❖ **각 단어에 대응되는 값들에 개별적으로 FFN를 적용하여 비선형성 부여**

- 동일한 encoder에서는 parameter sharing을 통해 연산 효율성 확보

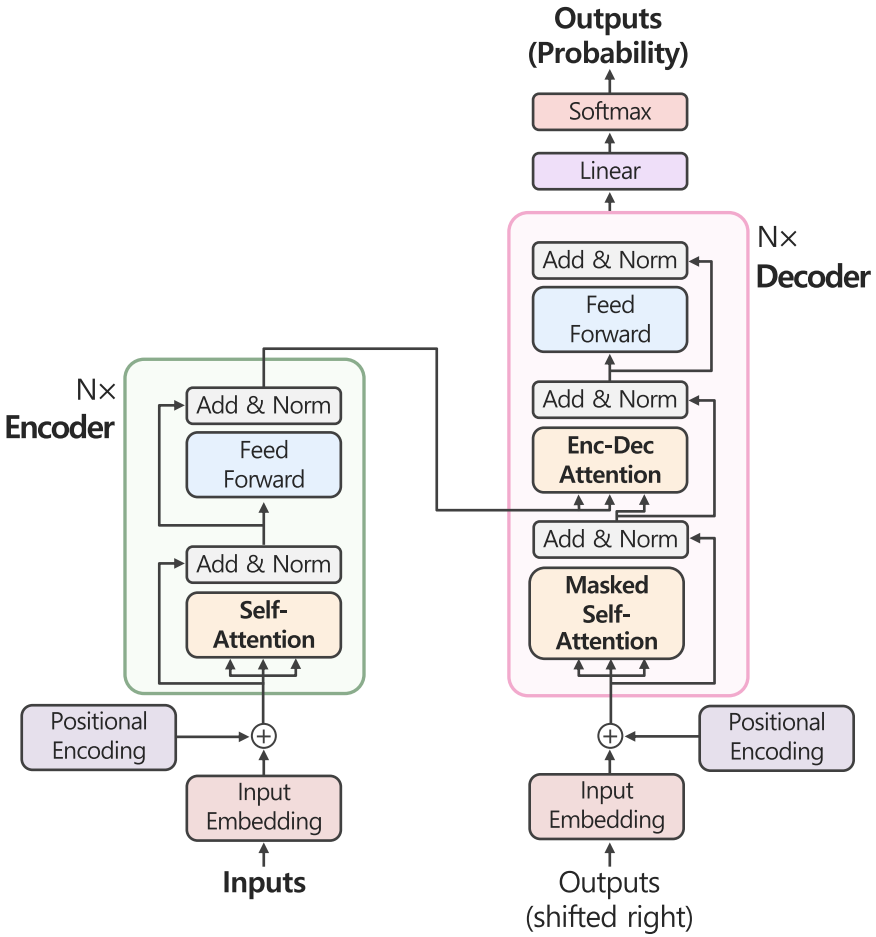




Transformer

Model Architecture

❖ Transformer는 Encoder-Decoder 구조로 구성



① Embedding

② Positional Encoding

③ Encoder

- Self-Attention
- Feed Forward

④ Decoder

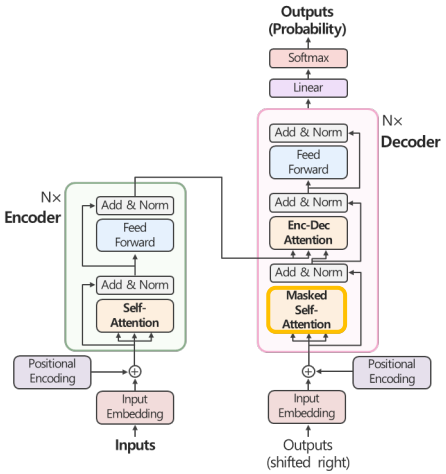
- Masked Self-Attention
- Encoder-Decoder Attention
- Feed Forward

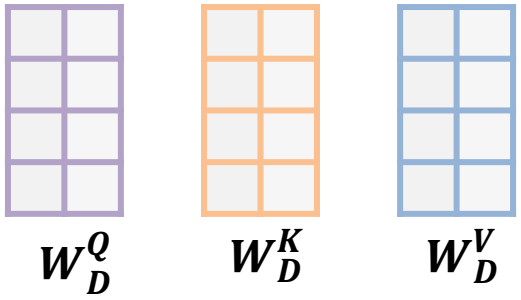
⑤ Prediction

Transformer

④ Decoder : Masked Self-Attention

❖ 뒤에 나오는 단어에 대해 반영하지 않고 현재 주어진 단어사이 관계 고려





W_D^Q W_D^K W_D^V

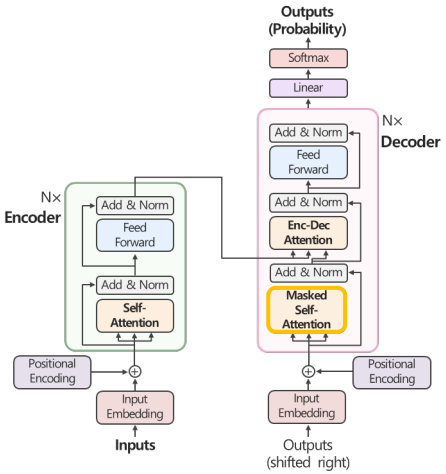
$$d_k = \frac{d_{model}\,(512)}{num\ heads\ (8)} = 64$$

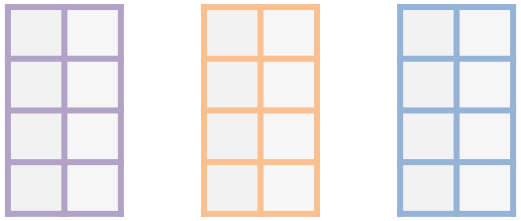
	I	am	a	student
Embedding w/ Time Signal	x_1	x_2	x_3	x_4
Queries - Decoder	q_1	q_2	q_3	q_4
Keys - Decoder	k_1	k_2	k_3	k_4
Values - Decoder	v_1	v_2	v_3	v_4
Score	$q_1 \cdot k_1 = 152$	$q_1 \cdot k_2 = 103$	$q_1 \cdot k_3 = 48$	$q_1 \cdot k_4 = 136$
Divide by 8 ($\sqrt{d_k}$)	152 / 8 = 19	103 / 8 = 13	48 / 8 = 6	136 / 8 = 17
Softmax	0.879	0.002	0.000	0.119
Softmax × Value	v'_1	v'_2	v'_3	v'_4
Summation	z_1			

Transformer

④ Decoder : Masked Self-Attention

❖ 뒤에 나오는 단어에 대해 반영하지 않고 현재 주어진 단어사이 관계 고려





W_D^Q W_D^K W_D^V

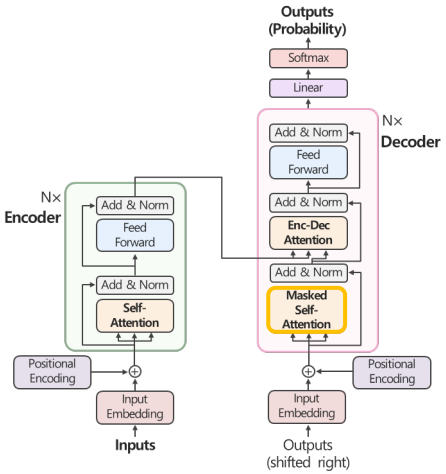
$$d_k = \frac{d_{model}\,(512)}{num\ heads\,(8)} = 64$$

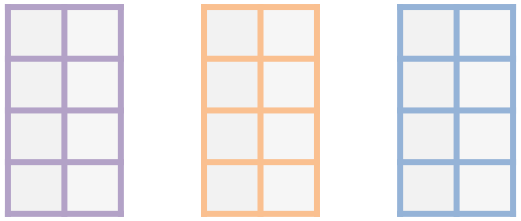
	I	am	a	student
Embedding w/ Time Signal	x_1	x_2	x_3	x_4
Queries - Decoder	q_1	q_2	q_3	q_4
Keys - Decoder	k_1	k_2	k_3	k_4
Values - Decoder	v_1	v_2	v_3	v_4
Score	$q_1 \cdot k_1 = 152$	$q_1 \cdot k_2 = -\infty$	$q_1 \cdot k_3 = -\infty$	$q_1 \cdot k_4 = -\infty$
Divide by 8 ($\sqrt{d_k}$)	152 / 8 = 19	$-\infty$ / 8 = $-\infty$	$-\infty$ / 8 = $-\infty$	$-\infty$ / 8 = $-\infty$
Softmax	1	0	0	0
Softmax × Value	v'_1	v'_2	v'_3	v'_4
Summation	z_1			

Transformer

④ Decoder : Masked Self-Attention

❖ 뒤에 나오는 단어에 대해 반영하지 않고 현재 주어진 단어사이 관계 고려





W_D^Q W_D^K W_D^V

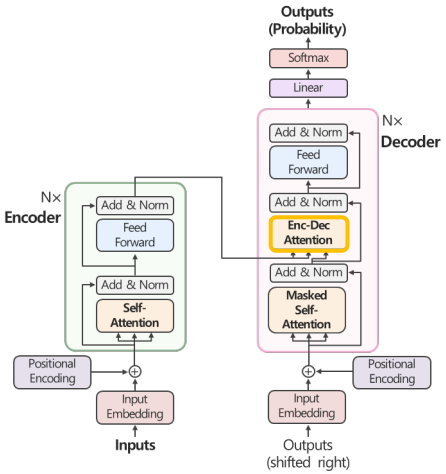
$$d_k = \frac{d_{model}\,(512)}{num\ heads\,(8)} = 64$$

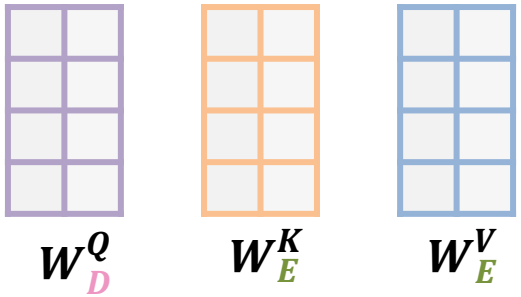
	I	am	a	student
Embedding w/ Time Signal	x_1	x_2	x_3	x_4
Queries - Decoder	q_1	q_2	q_3	q_4
Keys - Decoder	k_1	k_2	k_3	k_4
Values - Decoder	v_1	v_2	v_3	v_4
Score	$q_2 \cdot k_1 = 102$	$q_2 \cdot k_2 = 136$	$q_2 \cdot k_3 = -\infty$	$q_2 \cdot k_4 = -\infty$
Divide by 8 ($\sqrt{d_k}$)	102 / 8 = 13	136 / 8 = 17	$-\infty$ / 8 = $-\infty$	$-\infty$ / 8 = $-\infty$
Softmax	0.018	0.982	0	0
Softmax × Value	v'_1	v'_2	v'_3	v'_4
Summation		z_2		

Transformer

④ Decoder : Encoder-Decoder Attention

❖ Encoder의 Key, Value values를 활용하여 encoding 정보 고려





W_D^Q W_E^K W_E^V

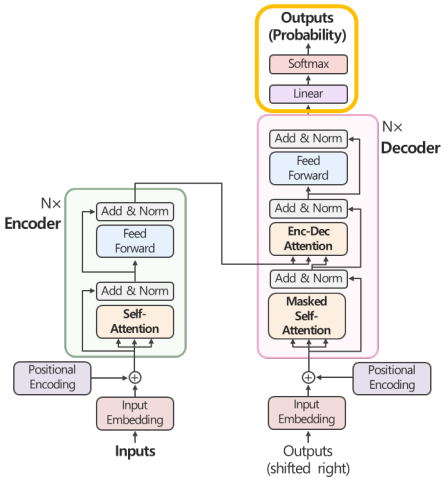
$$d_k = \frac{d_{model}\,(512)}{num\ heads\,(8)} = 64$$

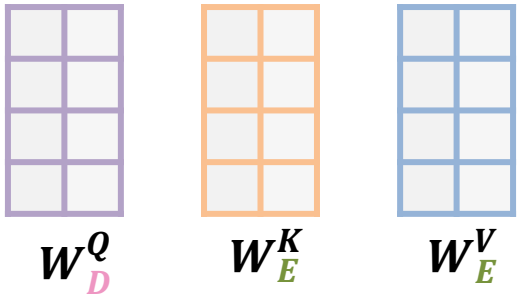
	I	am	a	student
Decoded Vector	z'_1	z'_2	z'_3	z'_4
Queries - Decoder	q_1	q_2	q_3	q_4
Keys - Encoder	k_1 나는	k_2 학생	k_3 이다	
Values - Encoder	v_1	v_2	v_3	
Score		$q_2 \cdot k_1 = 98$	$q_2 \cdot k_2 = 67$	$q_2 \cdot k_3 = 102$
Divide by 8 ($\sqrt{d_k}$)		98 / 8 = 12	67 / 8 = 8	102 / 8 = 13
Softmax		0.268	0.005	0.727
Softmax × Value	v'_1	v'_2	v'_3	
Summation		z_2		

Transformer

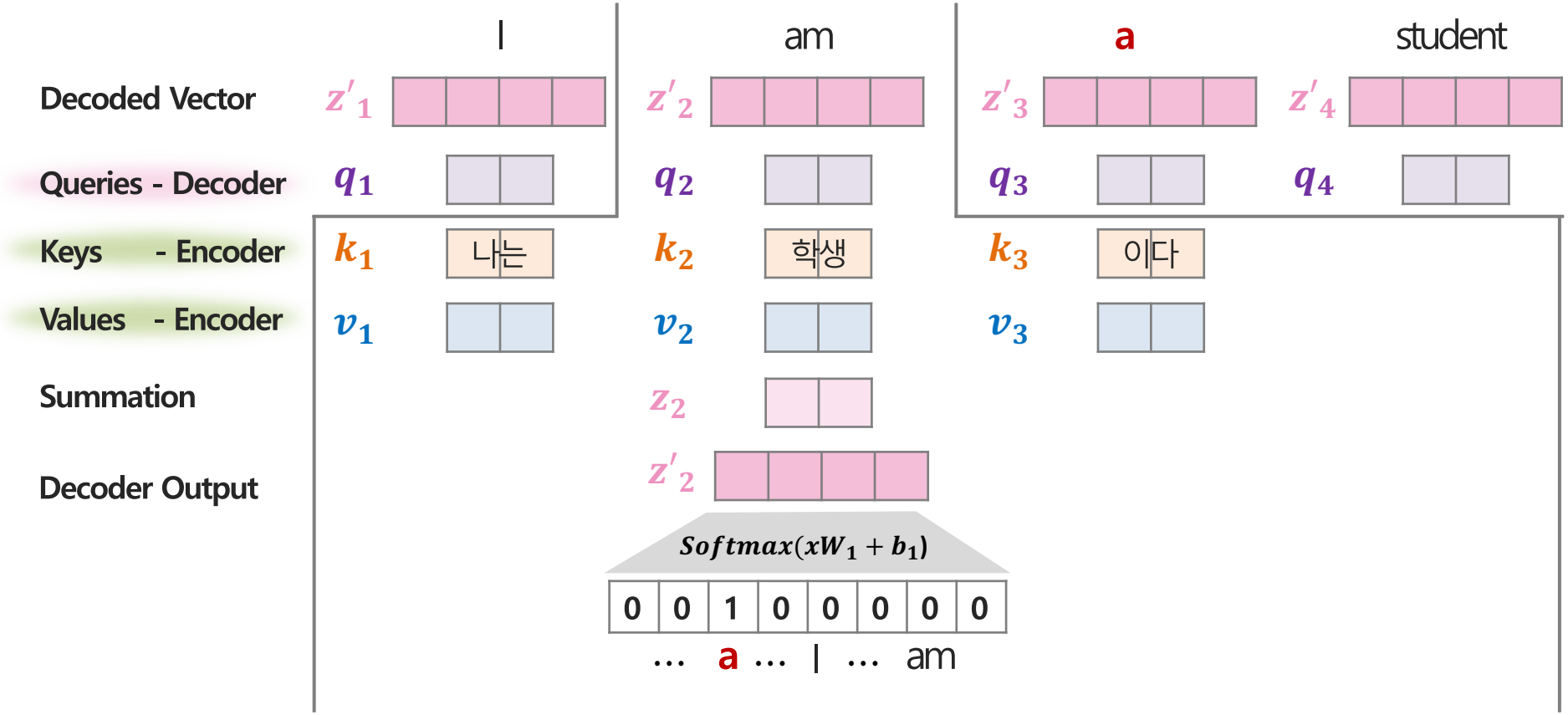
④ Decoder : Prediction

❖ 해당 모듈은 수행하고자 하는 task에 따라 변형 가능





$$d_k = \frac{d_{model}\,(512)}{num\ heads\,(8)} = 64$$

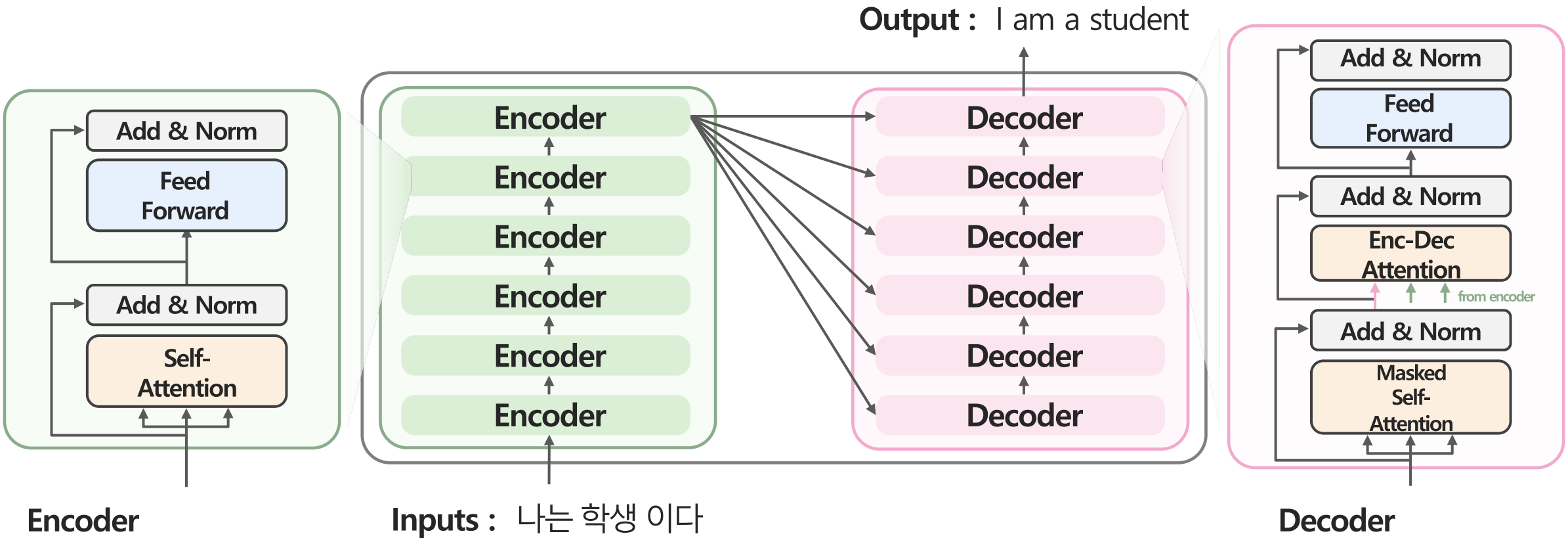


Transformer

Model Architecture

❖ **Transformer는 Encoder-Decoder 구조로 구성**

- 각 N개의 모듈로 구성되며 구조는 동일

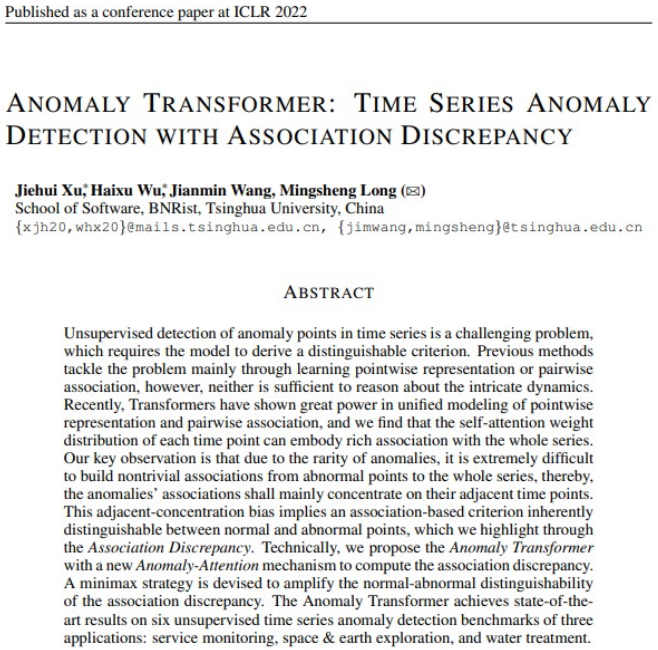


Anomaly Transformer

Introduction

❖ Anomaly Transformer: Time series anomaly detection with association discrepancy (2022, ICLR)

- 2023년 1월 기준 45회 인용
- Transformer 구조를 다변량 시계열 데이터에 접목시킨 사례로 현재 SOTA로 알려짐
- https://github.com/thuml/Anomaly-Transformer



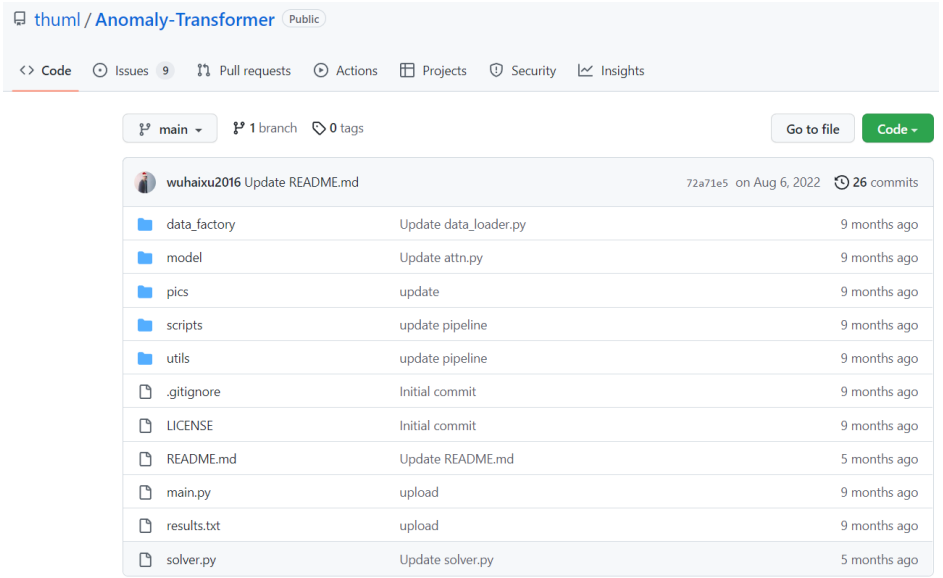
Published as a conference paper at ICLR 2022

ANOMALY TRANSFORMER: TIME SERIES ANOMALY DETECTION WITH ASSOCIATION DISCREPANCY

Jiehui Xu,* Haixu Wu,* Jianmin Wang, Mingsheng Long (✉)
School of Software, BNRist, Tsinghua University, China
{xjh20,whx20}@mails.tsinghua.edu.cn, {jimwang,mingsheng}@tsinghua.edu.cn

ABSTRACT

Unsupervised detection of anomaly points in time series is a challenging problem, which requires the model to derive a distinguishable criterion. Previous methods tackle the problem mainly through learning pointwise representation or pairwise association, however, neither is sufficient to reason about the intricate dynamics. Recently, Transformers have shown great power in unified modeling of pointwise representation and pairwise association, and we find that the self-attention weight distribution of each time point can embody rich association with the whole series. Our key observation is that due to the rarity of anomalies, it is extremely difficult to build nontrivial associations from abnormal points to the whole series, thereby, the anomalies' associations shall mainly concentrate on their adjacent time points. This adjacent-concentration bias implies an association-based criterion inherently distinguishable between normal and abnormal points, which we highlight through the *Association Discrepancy*. Technically, we propose the *Anomaly Transformer* with a new *Anomaly-Attention* mechanism to compute the association discrepancy. A minimax strategy is devised to amplify the normal-abnormal distinguishability of the association discrepancy. The Anomaly Transformer achieves state-of-the-art results on six unsupervised time series anomaly detection benchmarks of three applications: service monitoring, space & earth exploration, and water treatment.



thuml / Anomaly-Transformer Public

Code | Issues 9 | Pull requests | Actions | Projects | Security | Insights

main | 1 branch | 0 tags | Go to file | Code

wuhaixu2016 Update README.md — 72a71e5 on Aug 6, 2022 — 26 commits

data_factory	Update data_loader.py	9 months ago
model	Update attn.py	9 months ago
pics	update	9 months ago
scripts	update pipeline	9 months ago
utils	update pipeline	9 months ago
.gitignore	Initial commit	9 months ago
LICENSE	Initial commit	9 months ago
README.md	Update README.md	5 months ago
main.py	upload	9 months ago
results.txt	upload	9 months ago
solver.py	Update solver.py	5 months ago

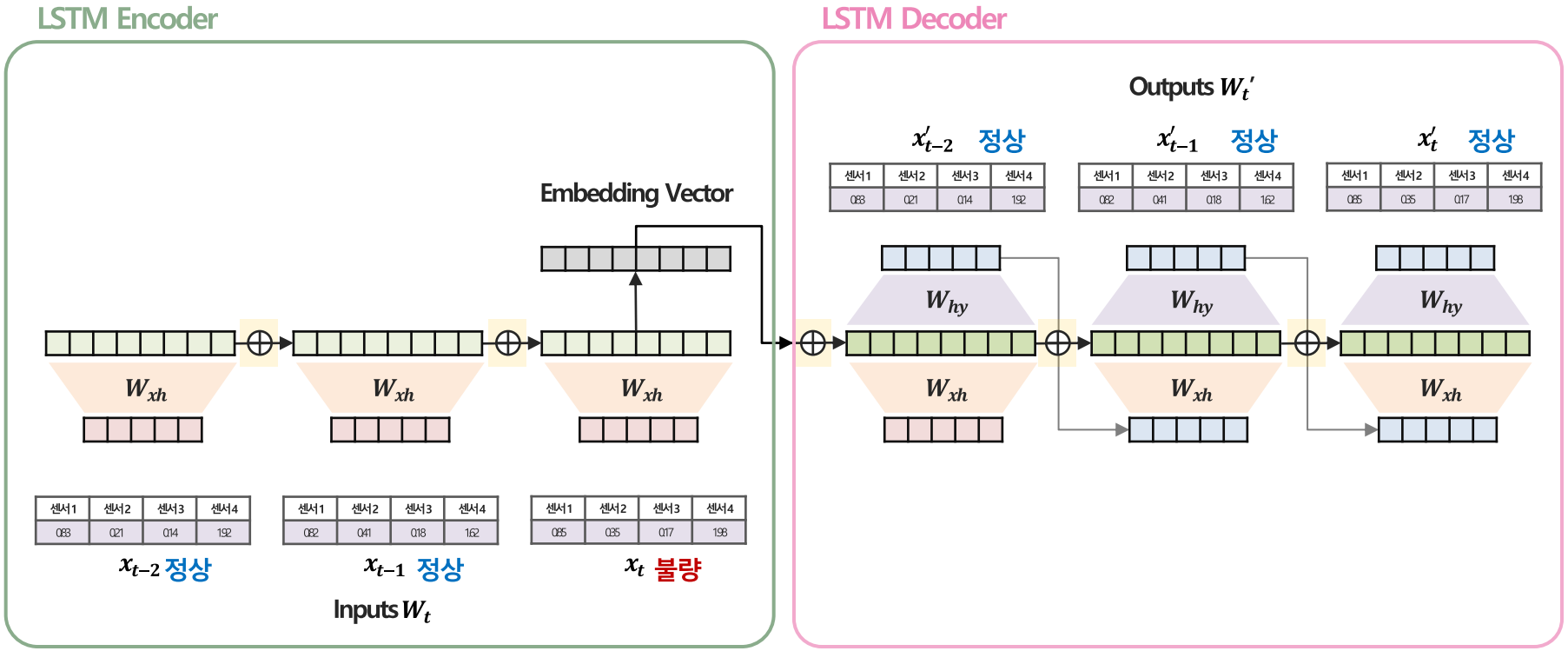
Xu, J., Wu, H., Wang, J., & Long, M. (2021, September). Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *International Conference on Learning Representations*.

Anomaly Transformer

Introduction

❖ 기존 다변량 시계열 데이터에서의 anomaly detection 한계

- 불량에 대한 시점은 정상에 대한 시점 대비 매우 적게 빈출되므로 모델이 정상에 과적합될 우려가 있음
- Hidden state가 각 순차적으로 연산되므로 전반적인 시계열 정보에 대한 반영이 어려울 수 있음

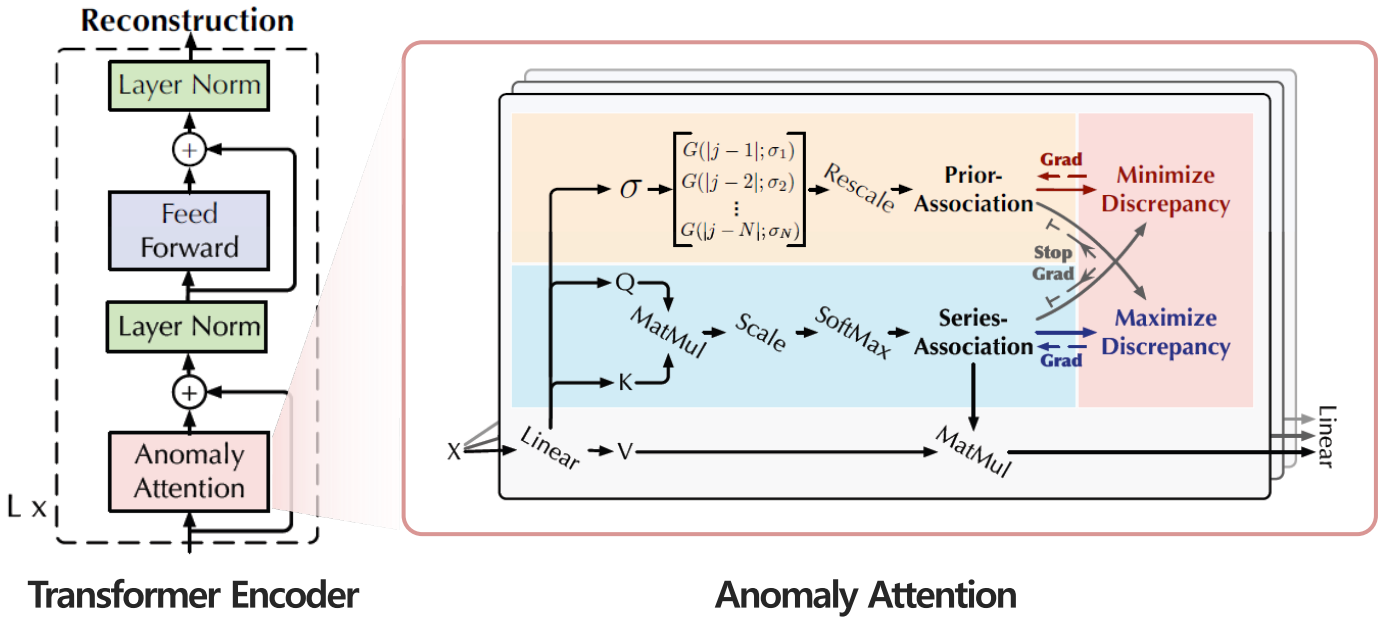


Anomaly Transformer

Introduction

❖ **Transformer구조를 통해 기존 다변량 시계열 데이터에서의 anomaly detection 개선**

- 불량에 대한 시점은 정상에 대한 시점 대비 매우 적게 빈출되므로 모델이 정상에 과적합될 우려가 있음
  - 시계열 데이터 특성상 **불량 시점은 인접한 시점과 관계**가 있으며 이러한 **지역적 시계열 특징**을 반영
- Anomaly score / Hidden state가 각 시점별로 연산되므로 전반적인 시계열 정보에 대한 반영이 어려울 수 있음
  - **모든 시점 사이 관계를 활용**하여 **전반적인 시계열 특징**을 반영



Anomaly Transformer

Introduction

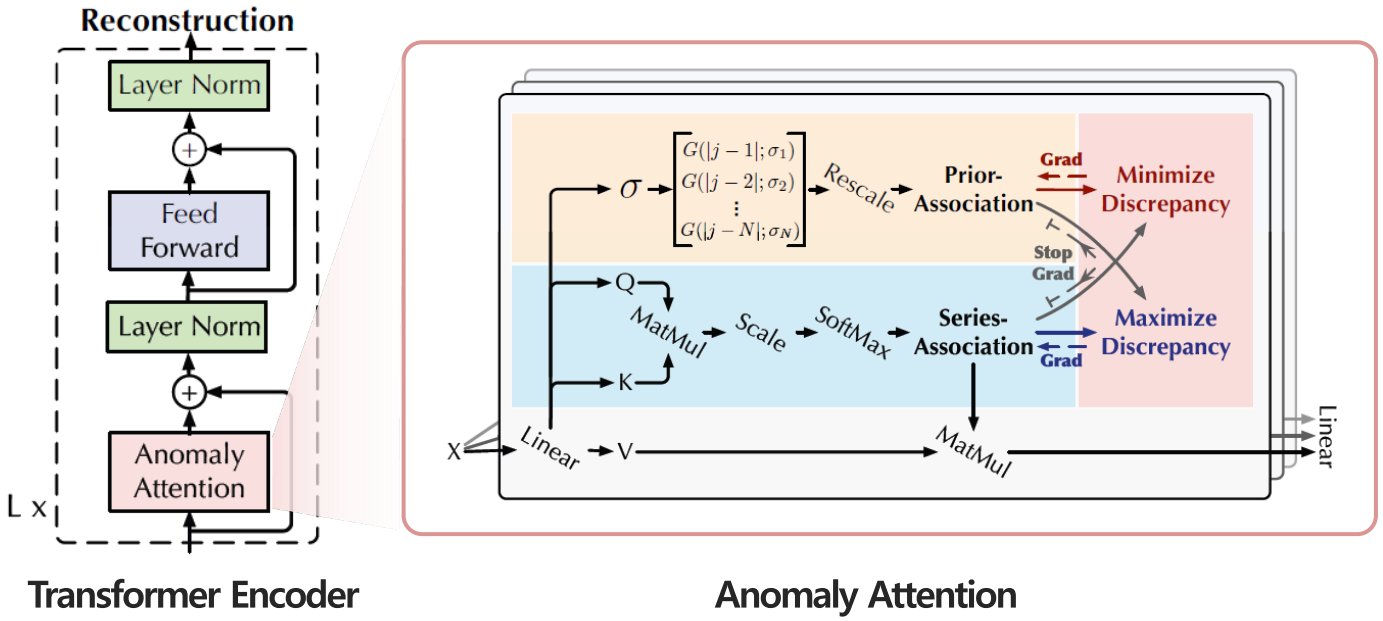
❖ **Transformer구조를 통해 기존 다변량 시계열 데이터에서의 anomaly detection 개선**

- 불량에 대한 시점은 정상에 대한 시점 대비 매우 적게 빈출되므로 모델이 정상에 과적합될 우려가 있음

Prior Association - 시계열 데이터 특성상 **불량 시점은 인접한 시점과 관계**가 있으며 이러한 **지역적 시계열 특징**을 반영

- Anomaly score / Hidden state가 각 시점별로 연산되므로 전반적인 시계열 정보에 대한 반영이 어려울 수 있음

Series Association - **모든 시점 사이 관계를 활용**하여 **전반적인 시계열 특징**을 반영



① Prior Association
② Series Association
} Attention

③ Association Discrepancy
= Prior 와 Series 사이 차이

정상과 불량의 P&S특징이 다름
이를 활용하여 이상치 탐지

Anomaly Transformer

Introduction

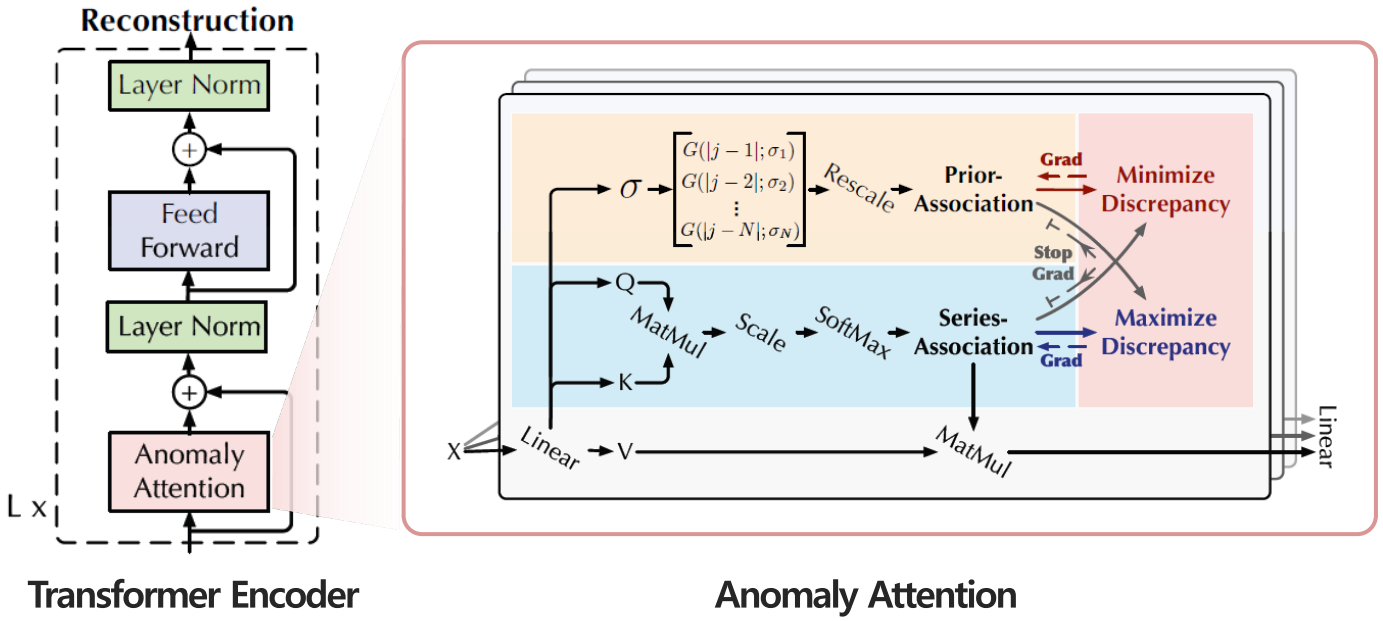
❖ **Transformer구조를 통해 기존 다변량 시계열 데이터에서의 anomaly detection 개선**

- 불량에 대한 시점은 정상에 대한 시점 대비 매우 적게 빈출되므로 모델이 정상에 과적합될 우려가 있음

Prior Association - 시계열 데이터 특성상 **불량 시점은 인접한 시점과 관계**가 있으며 이러한 **지역적 시계열 특징**을 반영

- Anomaly score / Hidden state가 각 시점별로 연산되므로 전반적인 시계열 정보에 대한 반영이 어려울 수 있음

Series Association - **모든 시점 사이 관계를 활용**하여 **전반적인 시계열 특징**을 반영



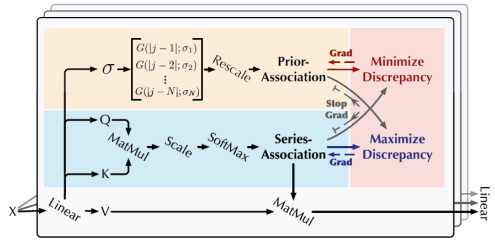
① Prior Association
② Series Association
} Attention

③ Association Discrepancy
= Prior 와 Series 사이 차이

정상과 불량의 P&S특징이 다름
이를 활용하여 이상치 탐지

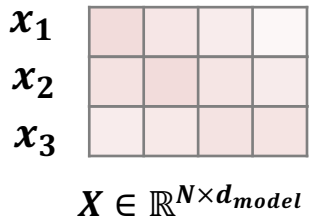
Anomaly Transformer

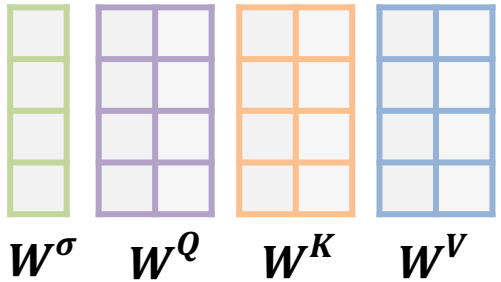
Architecture

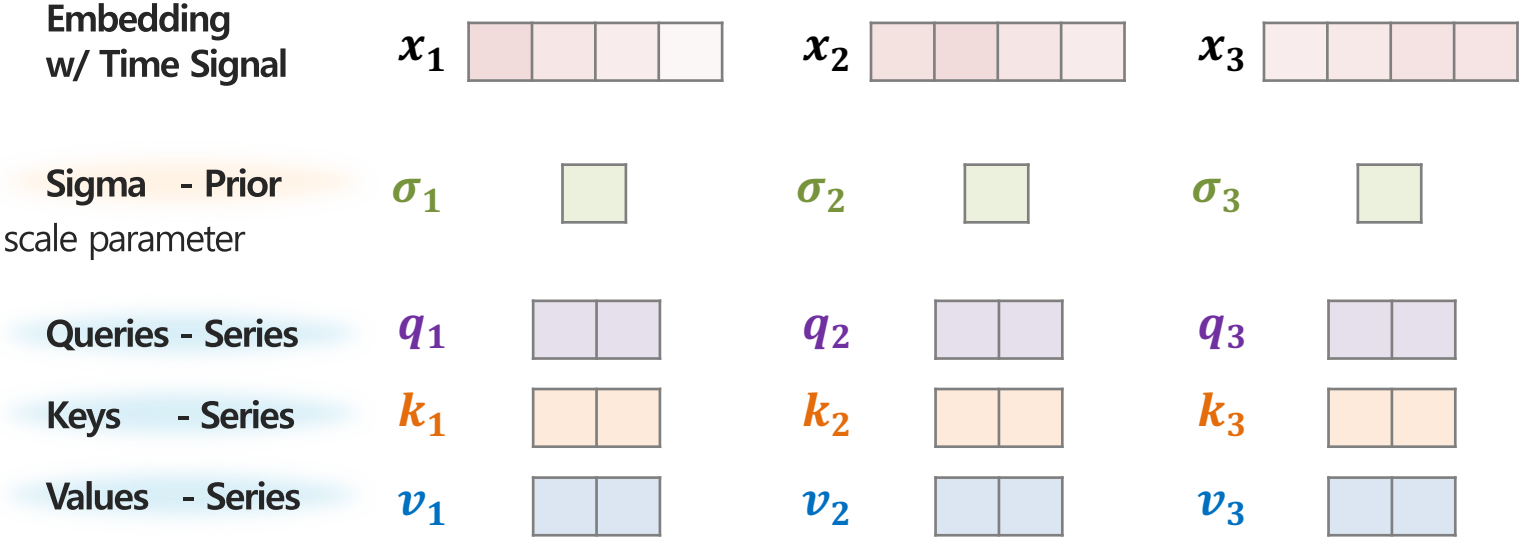


❖ 학습 파라미터 $W^{\sigma}, W^{Q}, W^{K}, W^{V}$ 를 통해 각 σ, Q, K, V 값 도출

- 기존 Transformer에서 정규분포의 scale parameter인 σ 가 추가로 도출되는 형태





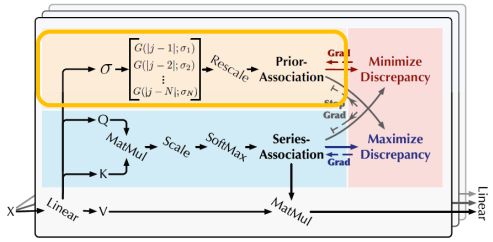


$W^{\sigma} \in \mathbb{R}^{d_{model} \times 1}$

$W^{Q}, W^{K}, W^{V} \in \mathbb{R}^{d_{model} \times d_{model}/h}$

Anomaly Transformer

① Prior Association



❖ **Prior Association: 인접한 시점에 큰 가중치를 부여하기 위한 attention score**

- Gaussian Distribution은 unimodal구조로 인접한 시점에 큰 가중치를 부여하기에 적합

x_1	
x_2	
x_3	

$X \in \mathbb{R}^{N \times d_{model}}$

σ_1	5
σ_2	2
σ_3	1

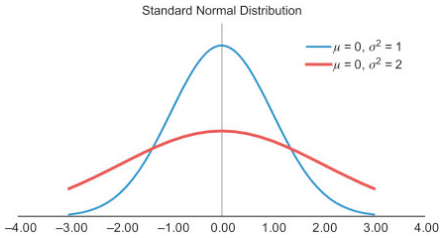
$\sigma \in \mathbb{R}^{N \times 1}$

	0	1	2
0	0	1	4
1	1	0	1
2	4	1	0

$|j-i|^2$

0.08	0.07	0.06
0.18	0.20	0.18
0.05	0.24	0.40

$\frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)$



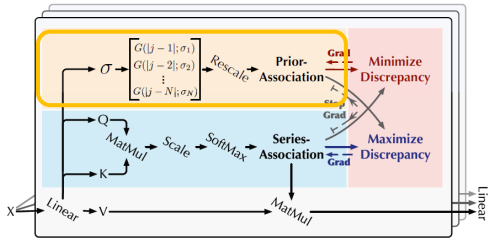
$$P = Rescale\left(\frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)\right)$$

$|j-i|^2$: 순차 인덱스 값 차이

σ_i : 학습하여 도출된 해당 인덱스에서의 정규분포 표준편차

Anomaly Transformer

① Prior Association



❖ Prior Association: 인접한 시점에 큰 가중치를 부여하기 위한 attention score

- Gaussian Distribution은 unimodal구조로 인접한 시점에 큰 가중치를 부여하기에 적합

x_1
x_2
x_3

$X \in \mathbb{R}^{N \times d_{model}}$

σ_1	5
σ_2	2
σ_3	1

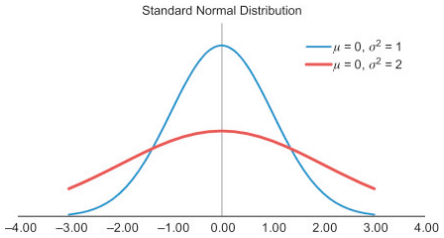
$\sigma \in \mathbb{R}^{N \times 1}$

	0	1	2
0	0	1	4
1	1	0	1
2	4	1	0

$|j-i|^2$

0.08	0.07	0.06
0.18	0.20	0.18
0.05	0.24	0.40

$\frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)$



$$P = Rescale\left(\frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)\right)$$

$|j-i|^2$: 순차 인덱스 값 차이

σ_i : 학습하여 도출된 해당 인덱스에서의 정규분포 표준편차

0.21
0.56
0.69

$$\sum_{j=0}^{2} \frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)$$

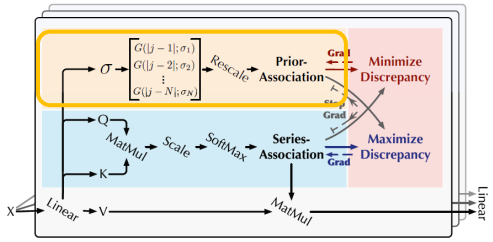
0.38	0.33	0.29
0.32	0.36	0.32
0.07	0.34	0.58

Prior Association

$$P = Rescale\left(\frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)\right)$$

Anomaly Transformer

① Prior Association



❖ Prior Association: 인접한 시점에 큰 가중치를 부여하기 위한 attention score

- Gaussian Distribution은 unimodal구조로 인접한 시점에 큰 가중치를 부여하기에 적합
- σ가 작을수록 인접한 시점 정보를 반영

x_1
x_2
x_3

$X \in \mathbb{R}^{N \times d_{model}}$

σ_1	5
σ_2	2
σ_3	1

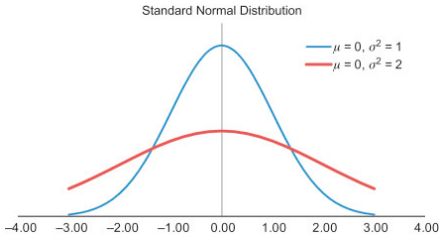
$\sigma \in \mathbb{R}^{N \times 1}$

	0	1	2
0	0	1	4
1	1	0	1
2	4	1	0

$|j-i|^2$

0.08	0.07	0.06
0.18	0.20	0.18
0.05	0.24	0.40

$$\frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)$$



$$P = Rescale\left(\frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)\right)$$

$|j-i|^2$: 순차 인덱스 값 차이

σ_i : 학습하여 도출된 해당 인덱스에서의 정규분포 표준편차

0.21
0.56
0.69

$$\sum_{j=0}^{2} \frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)$$

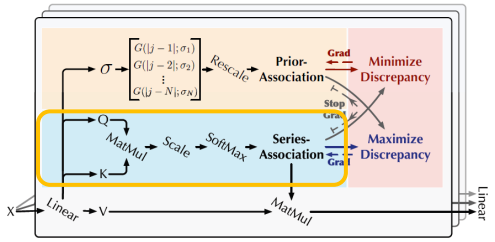
0.38	0.33	0.29
0.32	0.36	0.32
0.07	0.34	0.58

Prior Association

$$P = Rescale\left(\frac{1}{\sqrt{2\pi}\sigma_i} exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)\right)$$

Anomaly Transformer

② Series Association



❖ **Series Association: 모든 시점 사이 유사도에 따른 가중치를 부여하기 위한 attention score**

- Transformer에서 self-attention구조와 동일하며 복원에 직접적으로 영향을 미침

$$X \times W^{Q} = Q \quad (3\times4)\times(4\times2)=(3\times2)$$

$$X \times W^{K} = K$$

$$X \times W^{V} = V$$

$$\text{Softmax}\left(\frac{Q K^{T}}{\sqrt{d_k}}\right) = S$$

0.64	0.21	0.15
0.05	0.74	0.21
0.08	0.10	0.82

S Series Association

$$S \times V = \hat{Z}$$

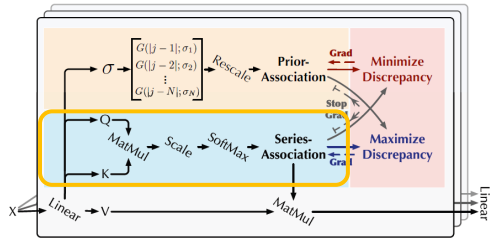
Series Association

$$S = Softmax\left(\frac{QK^{T}}{\sqrt{d_{model}}}\right)$$

$$\hat{Z} = SV$$

Anomaly Transformer

② Series Association



❖ **Series Association: 모든 시점 사이 유사도에 따른 가중치를 부여하기 위한 attention score**

- Transformer에서 self-attention구조와 동일하며 복원에 직접적으로 영향을 미침
- 최종적으로 복원 값인 reconstructed vector값을 도출

x_1
x_2
x_3

$X \in \mathbb{R}^{N \times d_{model}}$

W^{σ} W^{Q} W^{K} W^{V}

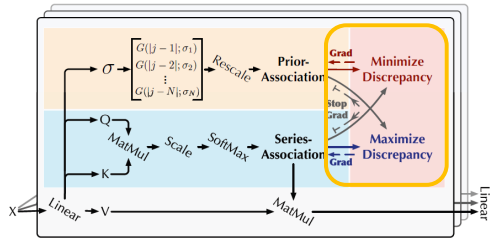
$W^{\sigma} \in \mathbb{R}^{d_{model} \times 1}$

$W^{Q}, W^{K}, W^{V} \in \mathbb{R}^{d_{model} \times d_{model}/h}$

Embedding w/ Time Signal	x_1	x_2	x_3
Sigma - Prior scale parameter	σ_1	σ_2	σ_3
Queries - Series	q_1	q_2	q_3
Keys - Series	k_1	k_2	k_3
Values - Series	v_1	v_2	v_3
Reconstructed vector	$\hat{z}_1$	$\hat{z}_2$	$\hat{z}_3$
	$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$

Anomaly Transformer

③ Association Discrepancy



❖ **Association Discrepancy: 두 association사이 차이를 의미하며, KL Divergence 값으로 연산**

- Prior Association: 인접한 시점에 큰 가중치를 부여하기 위한 attention score
- Series Association: 모든 시점 사이 유사도에 따른 가중치를 부여하기 위한 attention score

$$AssDis(P,S;X) = \left[\frac{1}{L}\sum_{l=1}^{L}\left(KL(P_{i,:}^{l}||S_{i,:}^{l}) + KL(S_{i,:}^{l}||P_{i,:}^{l})\right)\right]_{i=1,..,N}$$

$$KL(P_{i,:}^{l}||S_{i,:}^{l}) = \sum_{j=1}^{N} P_{i,j}^{l}\, ln\left(\frac{P_{i,j}^{l}}{S_{i,j}^{l}}\right)$$

$$KL(S_{i,:}^{l}||P_{i,:}^{l}) = \sum_{j=1}^{N} S_{i,j}^{l}\, ln\left(\frac{S_{i,j}^{l}}{P_{i,j}^{l}}\right)$$

l: 대응되는 layer 번호

0.38	0.33	0.29
0.32	0.36	0.32
0.07	0.34	0.58

Prior Association

$P \in \mathbb{R}^{N\times N}$

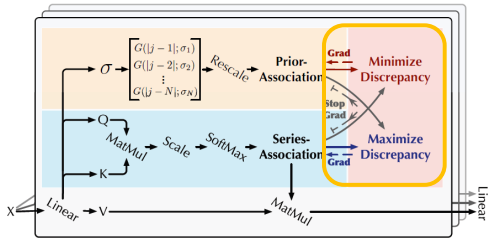
0.64	0.21	0.15
0.05	0.74	0.21
0.08	0.10	0.82

Series Association

$S \in \mathbb{R}^{N\times N}$

Anomaly Transformer

③ Association Discrepancy



❖ **Association Discrepancy: 두 association사이 차이를 의미하며, KL Divergence 값으로 연산**

- Prior Association: 인접한 시점에 큰 가중치를 부여하기 위한 attention score
- Series Association: 모든 시점 사이 유사도에 따른 가중치를 부여하기 위한 attention score

$$AssDis(P,S;X) = \left[\frac{1}{L}\sum_{l=1}^{L}\left(KL(P_{i,:}^{l}||S_{i,:}^{l}) + KL(S_{i,:}^{l}||P_{i,:}^{l})\right)\right]_{i=1,..,N}$$

$$KL(P_{i,:}^{l}||S_{i,:}^{l}) = \sum_{j=1}^{N} P_{i,j}^{l}\ ln\left(\frac{P_{i,j}^{l}}{S_{i,j}^{l}}\right)$$

$$KL(S_{i,:}^{l}||P_{i,:}^{l}) = \sum_{j=1}^{N} S_{i,j}^{l}\ ln\left(\frac{S_{i,j}^{l}}{P_{i,j}^{l}}\right)$$

l: 대응되는 layer 번호

0.38	0.33	0.29
0.32	0.36	0.32
0.07	0.34	0.58

Prior Association

$P \in \mathbb{R}^{N\times N}$

0.64	0.21	0.15
0.05	0.74	0.21
0.08	0.10	0.82

Series Association

$S \in \mathbb{R}^{N\times N}$

0.14
0.47
0.20

$KL(P_{i,:}^{1}||S_{i,:}^{1}) \in \mathbb{R}^{N\times 1}$

0.13
0.35
0.17

$KL(S_{i,:}^{1}||P_{i,:}^{1}) \in \mathbb{R}^{N\times 1}$

0.27
0.82
0.37

$KL(P_{i,:}^{1}||S_{i,:}^{1}) + KL(S_{i,:}^{1}||P_{i,:}^{1}) \in \mathbb{R}^{N\times 1}$

0.27
0.82
0.37

# of layers

$KL(P_{i,:}^{l}||S_{i,:}^{l}) + KL(S_{i,:}^{l}||P_{i,:}^{l}) \in \mathbb{R}^{N\times L}$

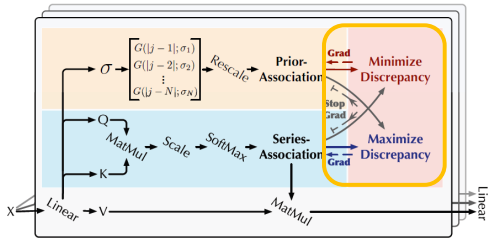
0.27
0.82
0.37

$AssDis(P,S;X) \in \mathbb{R}^{N\times 1}$

두 association의 차이를 함축

Anomaly Transformer

③ Association Discrepancy



❖ **Association Discrepancy: 두 association사이 차이를 의미하며, KL Divergence 값으로 연산**

- 불량에 대한 series association과 prior association이 유사함
- 불량에 대한 association discrepancy가 정상보다 작은 값을 지님

$$AssDis(P,S;X) = \left[\frac{1}{L}\sum_{l=1}^{L}\left(KL(P_{i,:}^{l}\|S_{i,:}^{l}) + KL(S_{i,:}^{l}\|P_{i,:}^{l})\right)\right]_{i=1,\ldots,N}$$

0.38	0.33	0.29
0.32	0.36	0.32
0.07	0.34	0.58

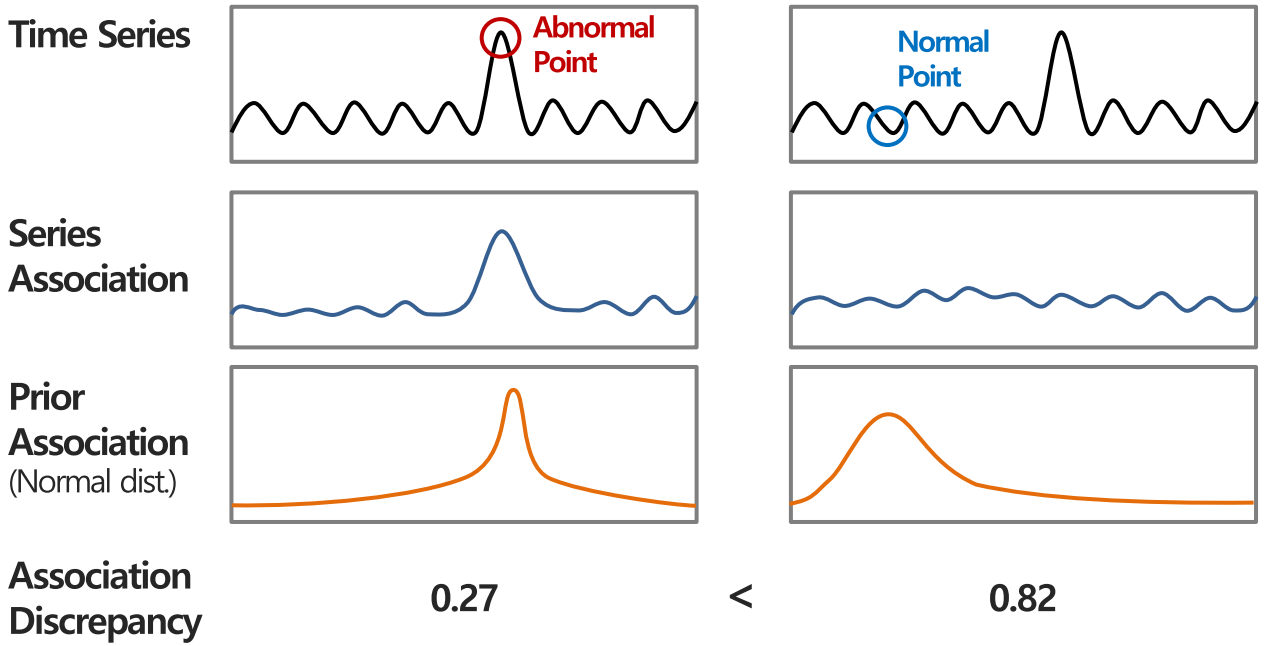
Prior Association

$P \in \mathbb{R}^{N\times N}$

0.64	0.21	0.15
0.05	0.74	0.21
0.08	0.10	0.82

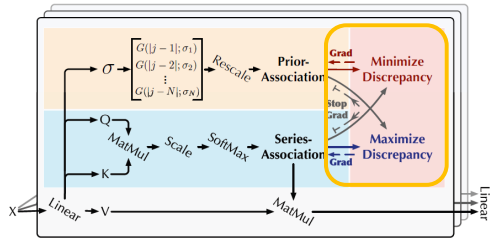
Series Association

$S \in \mathbb{R}^{N\times N}$



Anomaly Transformer

MinMax Association Learning



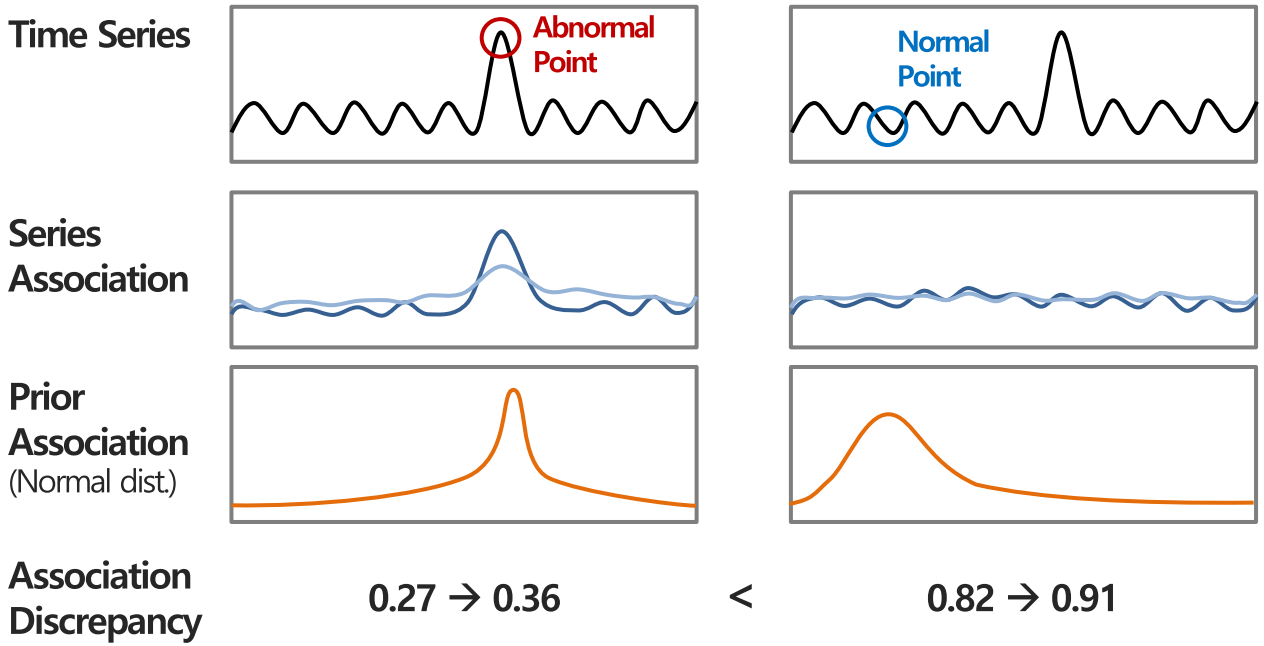
❖ **정상과 불량에 대한 차이를 극명하게 만들기 위해 association discrepancy에 대한 손실함수 추가**

- Reconstruction Loss: 각 시점별로 실제 값과 복원값이 유사해지도록 학습

$$Loss_{total}(\hat{X}, P, S, \lambda; X) = \|X - \hat{X}\|_F^2 - \lambda \times \|AssDis(P, S; X)\|_1$$

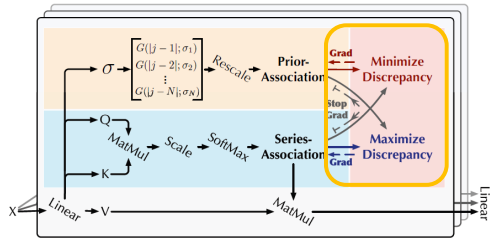
Minimization ↓ Maximization ↑

$$AssDis(P, S; X) = \left[\frac{1}{L}\sum_{l=1}^{L}(KL(P_{i,:}^{l} \| S_{i,:}^{l}) + KL(S_{i,:}^{l} \| P_{i,:}^{l})\right]_{i=1,\ldots,N}$$



Anomaly Transformer

MinMax Association Learning



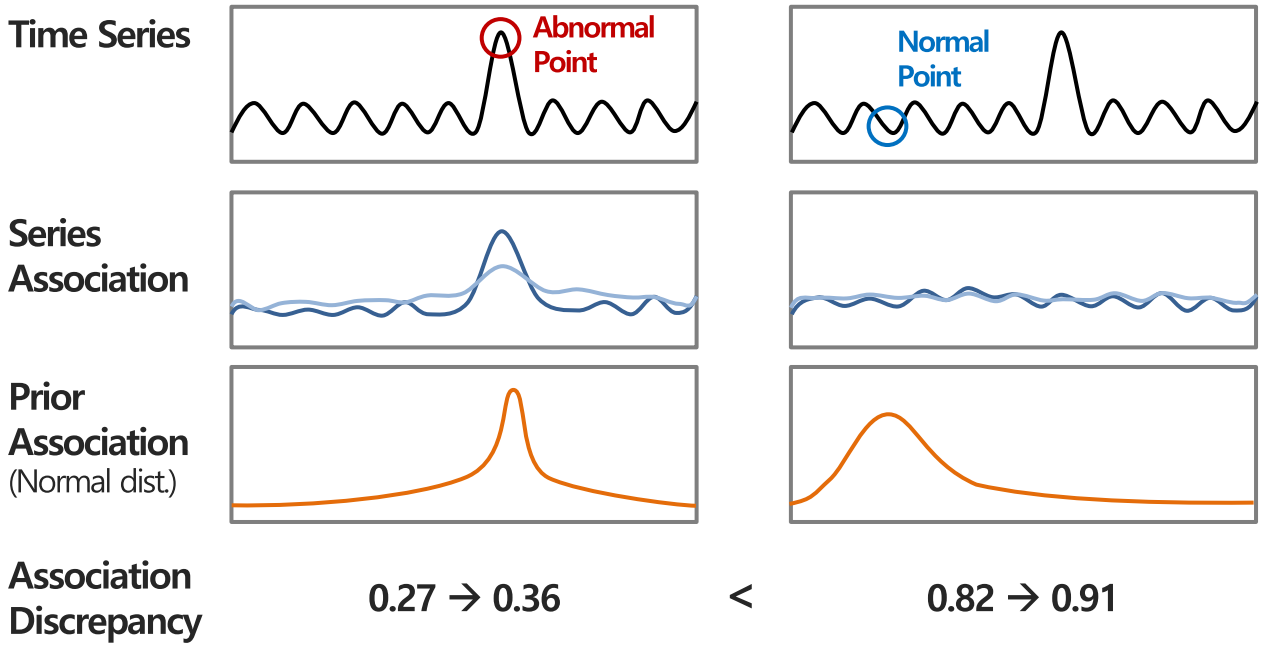
❖ **정상과 불량에 대한 차이를 극명하게 만들기 위해 association discrepancy에 대한 손실함수 추가**

- Association discrepancy: Series association이 **인접하지 않은 영역에 가중치를 크게 부여**하도록 학습
  - 복원에 직접적으로 연계되는 series association이 인접한 값에 적게 영향을 받으므로 불량에 대한 복원이 어려움
  - 결과적으로 정상과 불량에 대한 차이를 극명하게 만들어 이상치 탐지를 적절히 수행

$$Loss_{total}(\hat{X}, P, S, \lambda; X) = \|X - \hat{X}\|_F^2 - \lambda \times \|AssDis(P, S; X)\|_1$$

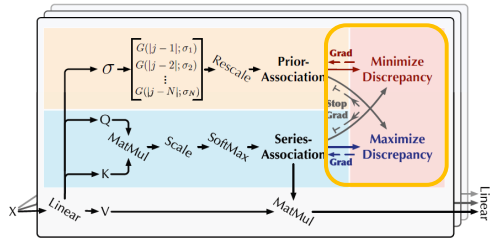
Minimization ↓ Maximization ↑

$$AssDis(P, S; X) = \left[\frac{1}{L}\sum_{l=1}^{L}(KL(P_{i,:}^{l} \| S_{i,:}^{l}) + KL(S_{i,:}^{l} \| P_{i,:}^{l})\right]_{i=1,\ldots,N}$$



Anomaly Transformer

MinMax Association Learning



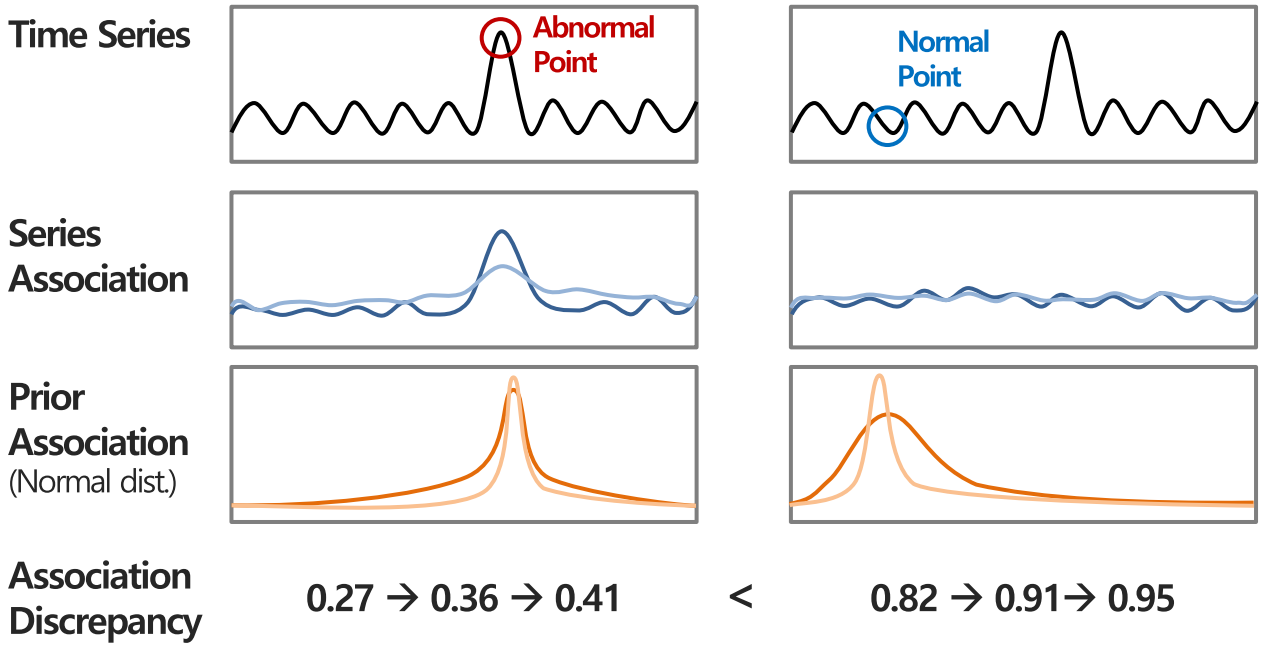
❖ **정상과 불량에 대한 차이를 극명하게 만들기 위해 association discrepancy에 대한 손실함수 추가**

- Association discrepancy: Series association이 인접하지 않은 영역에 가중치를 크게 부여하도록 학습
- 단순히 maximization을 진행하면 σ가 작아지므로 prior association의 의미가 퇴색될 수 있음
  - Minmax strategy 학습전략을 통해 완화하고자 함

$$Loss_{total}(\hat{X}, P, S, \lambda; X) = \|X - \hat{X}\|_F^2 - \lambda \times \|AssDis(P, S; X)\|_1$$

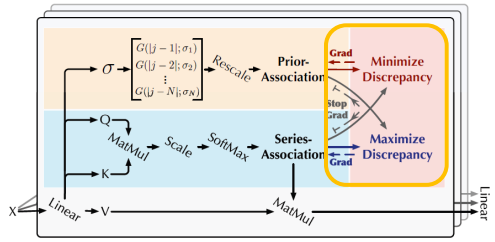
Minimization ↓ Maximization ↑

$$AssDis(P, S; X) = \left[\frac{1}{L}\sum_{l=1}^{L}(KL(P_{i,:}^l \| S_{i,:}^l) + KL(S_{i,:}^l \| P_{i,:}^l)\right]_{i=1,\ldots,N}$$



Anomaly Transformer

MinMax Association Learning



❖ Minimize Phase

- Prior association이 series association에 근사해지도록 학습
- σ가 너무 작아지는 것을 완화하며 prior association이 전체적인 시계열 패턴을 반영할 수 있도록 개선

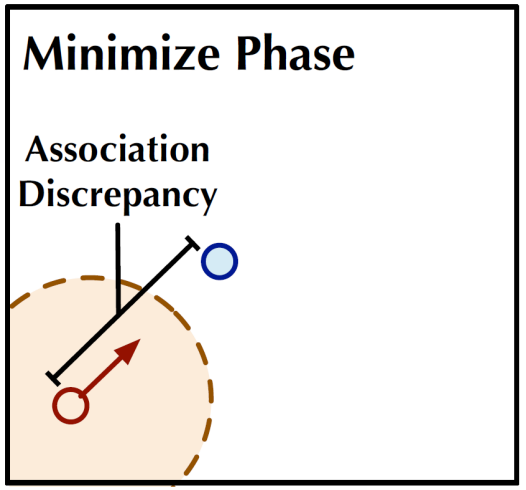
$$Loss_{total}(\hat{X}, P, S, \lambda; X) = \|X - \hat{X}\|_F^2 - \lambda \times \|AssDis(P, S; X)\|_1$$

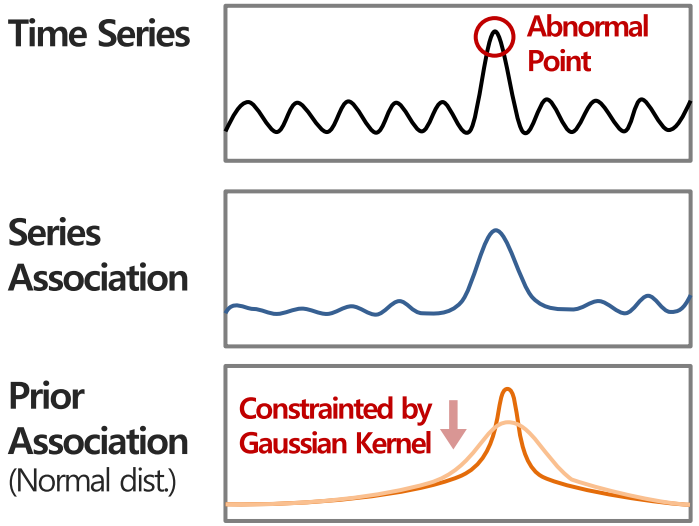
Minimization ↓ Maximization ↑

$$Minimize\ Phase: L_{total}(\hat{X}, P, S_{detach}, \lambda; X)$$

$$Maximize\ Phase: L_{total}(\hat{X}, P_{detach}, S, \lambda; X)$$

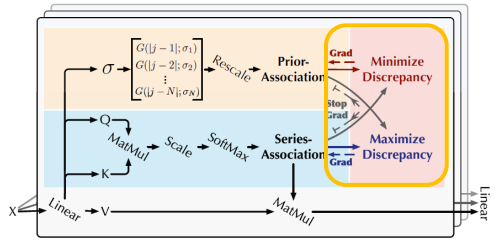
$detach$ = gradient backpropagation이 되지 않도록 고정





Anomaly Transformer

MinMax Association Learning



❖ Maximize Phase

- Series association을 보정하여association discrepancy를 최대화하도록 변형
- Series association이 인접하지 않은 시점에도 가중치가 부여될 수 있도록 학습

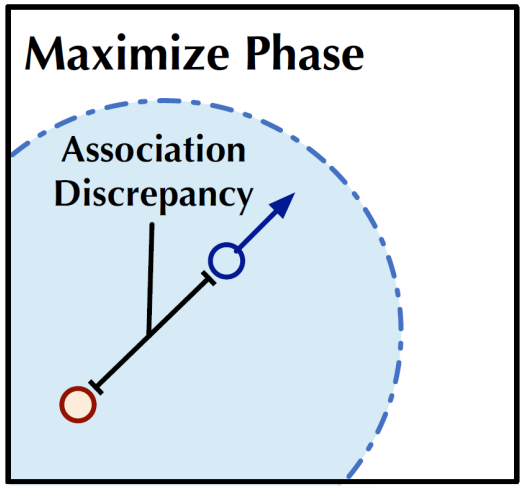
$$Loss_{total}(\hat{X}, P, S, \lambda; X) = \|X - \hat{X}\|_F^2 - \lambda \times \|AssDis(P, S; X)\|_1$$

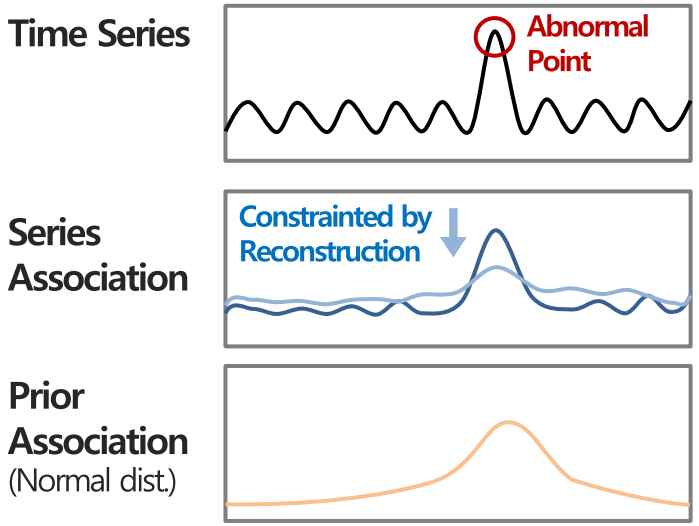
Minimization ↓ Maximization ↑

$$Minimize\ Phase: L_{total}(\hat{X}, P, S_{detach}, \lambda; X)$$

$$Maximize\ Phase: L_{total}(\hat{X}, P_{detach}, S, \lambda; X)$$

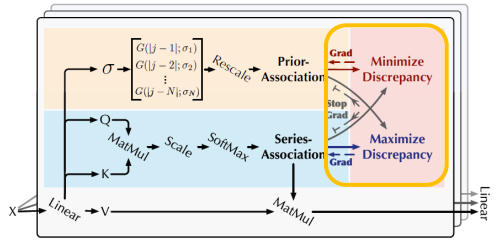
$detach$ = gradient backpropagation이 되지 않도록 고정





Anomaly Transformer

MinMax Association Learning



❖ 학습에 Reconstruction Loss 와 Association Discrepancy를 동시에 활용하는 효과

- Minimize phase를 통해 prior association을 series association와 유사하게 보정하여 고정
- Maximize phase를 통해 series association과 prior association의 차이가 커지도록 series association을 보정
- 복원에 직접적으로 연계되는 series association이 인접한 값에 적게 영향을 받으므로 불량에 대한 복원이 어려움

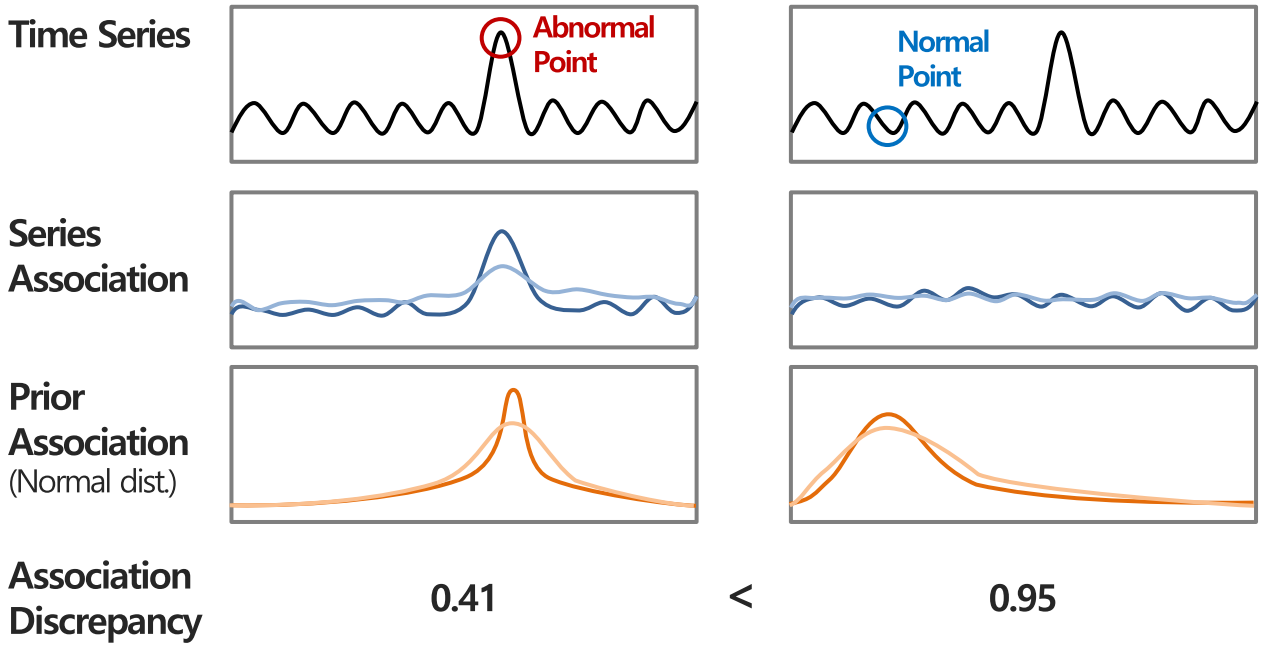
$$Loss_{total}(\hat{X}, P, S, \lambda; X) = \|X - \hat{X}\|_F^2 - \lambda \times \|AssDis(P, S; X)\|_1$$

Minimization ↓ Maximization ↑

$$Minimize\ Phase: L_{total}(\hat{X}, P, S_{detach}, \lambda; X)$$

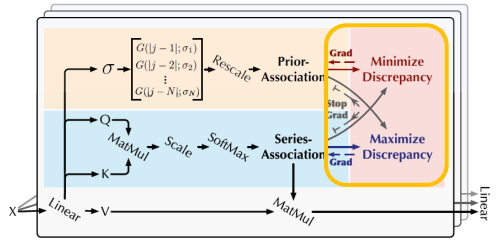
$$Maximize\ Phase: L_{total}(\hat{X}, P_{detach}, S, \lambda; X)$$

$detach$ = gradient backpropagation이 되지 않도록 고정



Anomaly Transformer

Association-based Anomaly Criterion



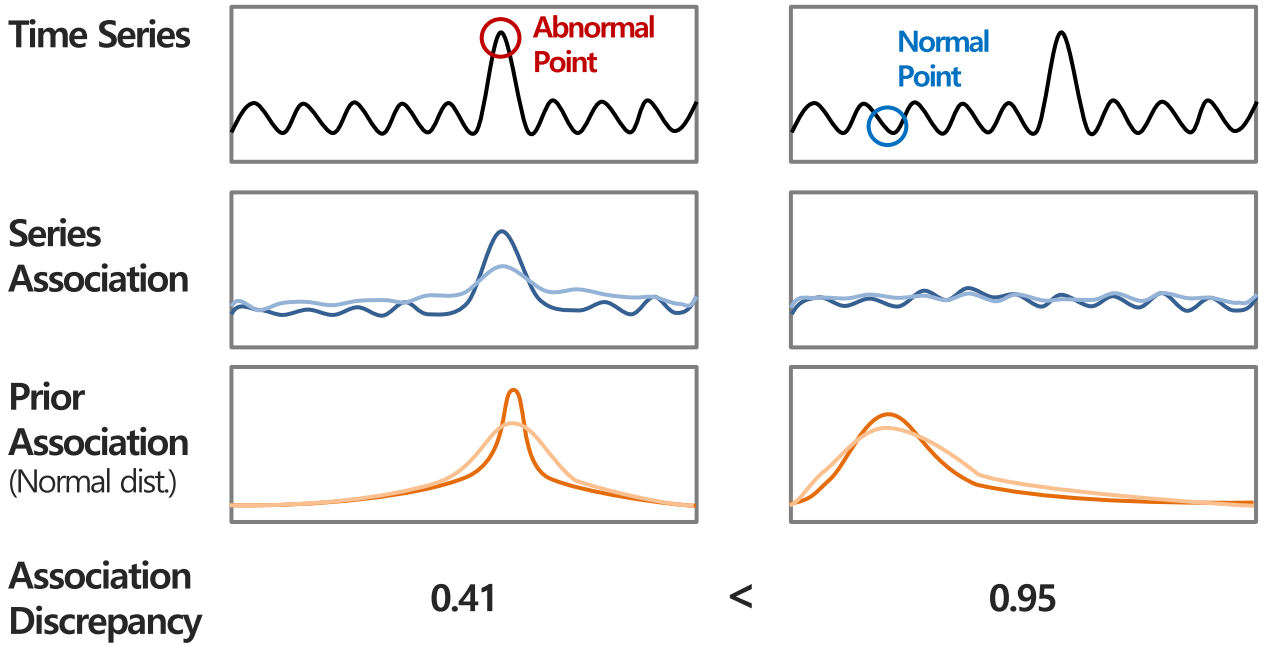
❖ **Anomaly score에 Reconstruction Loss 와 Association Discrepancy를 모두 활용**

- Series association이 인접한 값에 적게 영향을 받더라도 불량에 대한 association discrepancy는 정상대비 작음
- Series association이 인접한 값에 적게 영향을 받으므로 불량에 대한 복원이 어려움

$$Anomaly\,Score(X) = Softmax(-AssDis(P,S;X)) \times \left\|X - \hat{X}\right\|_2^2$$

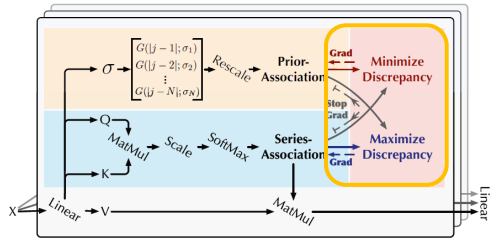
불량 : $AssDis(P,S;X)\downarrow$ & $\left\|X - \hat{X}\right\|_2^2\uparrow$

정상 : $AssDis(P,S;X)\uparrow$ & $\left\|X - \hat{X}\right\|_2^2\downarrow$



Anomaly Transformer

Association-based Anomaly Criterion



❖ **Anomaly score에 Reconstruction Loss 와 Association Discrepancy를 모두 활용**

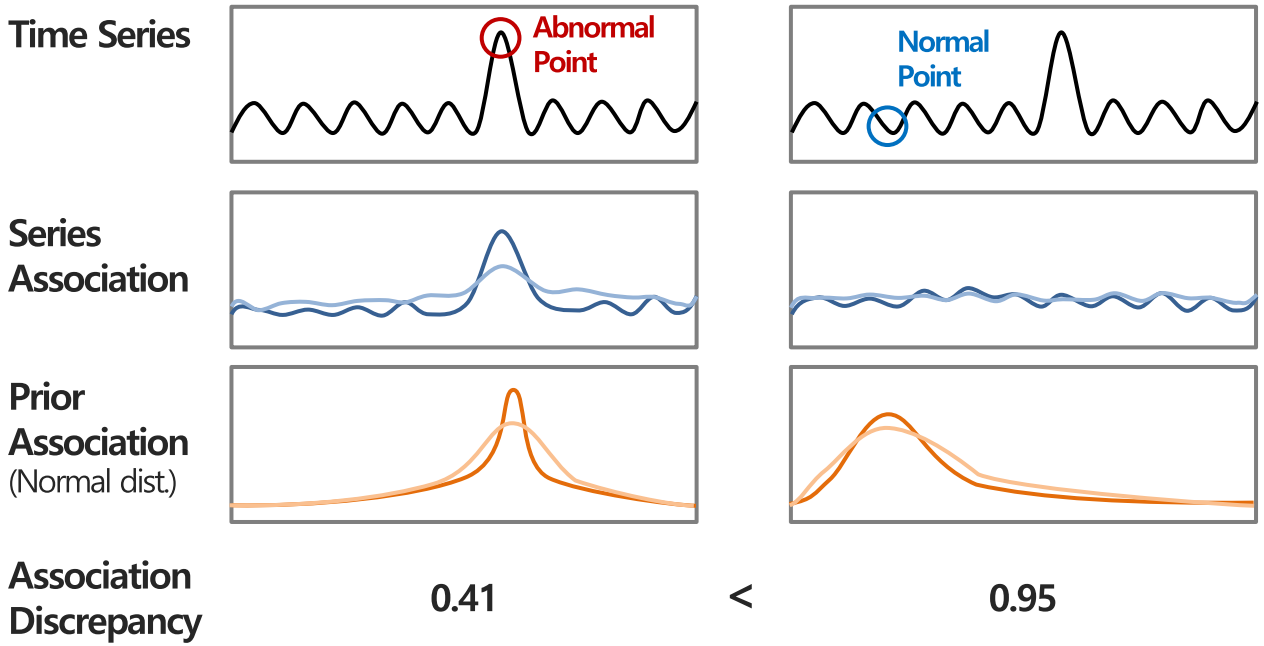
- Series association이 인접한 값에 적게 영향을 받더라도 불량에 대한 association discrepancy는 정상대비 작음
- Series association이 인접한 값에 적게 영향을 받으므로 불량에 대한 복원이 어려움

$$Anomaly\,Score(X) = Softmax(-AssDis(P,S;X)) \times \|X - \hat{X}\|_2^2$$

불량

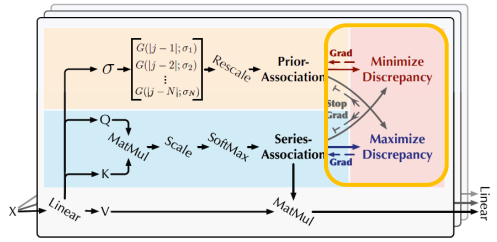
불량 : $AssDis(P,S;X)$↓ & $\|X - \hat{X}\|_2^2$↑

정상 : $AssDis(P,S;X)$↑ & $\|X - \hat{X}\|_2^2$↓



Anomaly Transformer

Association-based Anomaly Criterion



❖ **Anomaly score에 Reconstruction Loss 와 Association Discrepancy를 모두 활용**

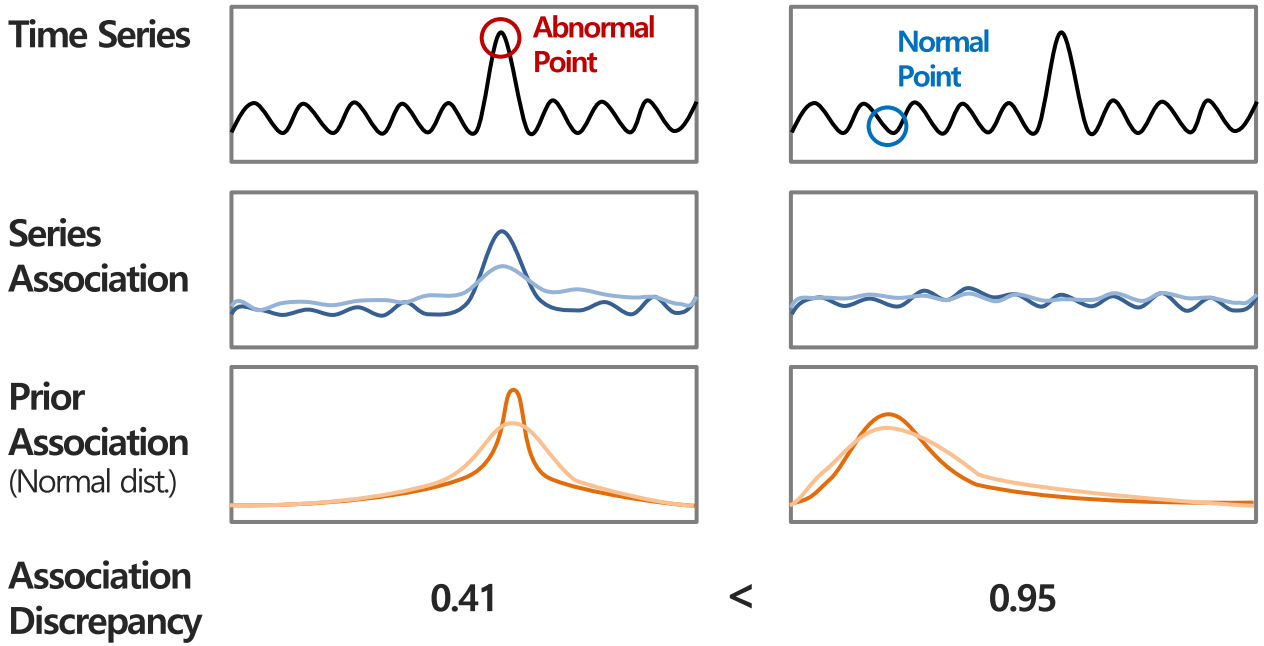
- Series association이 인접한 값에 적게 영향을 받더라도 불량에 대한 association discrepancy는 정상대비 작음
- Series association이 인접한 값에 적게 영향을 받으므로 불량에 대한 복원이 어려움

$$Anomaly\,Score(X) = Softmax(-AssDis(P,S;X)) \times \|X - \hat{X}\|_2^2$$

정상 ↓ ↓

불량 : $AssDis(P,S;X)$↓ & $\|X - \hat{X}\|_2^2$↑

정상 : $AssDis(P,S;X)$↑ & $\|X - \hat{X}\|_2^2$↓



Anomaly Transformer

Experiments

❖ **5개의 공용데이터와 다양한 시나리오를 다루는 NeurIPS-TS에 대해 적용**

- 입력: windowing process를 수행하여 sub-series형태로 입력 [Batch, Seq_len, Var]
- 출력: 각 시점별로 anomaly score / 상태 여부 출력 [Batch, Seq_len, 1]

Table 13: Details of benchmarks. AR represents the truth abnormal proportion of the whole dataset.

Benchmarks	Applications	Dimension	Window	#Training	#Validation	#Test (labeled)	AR (Truth)
SMD	Server	38	100	566,724	141,681	708,420	0.042
PSM	Server	25	100	105,984	26,497	87,841	0.278
MSL	Space	55	100	46,653	11,664	73,729	0.105
SMAP	Space	25	100	108,146	27,037	427,617	0.128
SWaT	Water	51	100	396,000	99,000	449,919	0.121
NeurIPS-TS	Various Anomalies	1	100	20,000	10,000	20,000	0.018

Anomaly Transformer

Experiments

❖ **다양한 비교실험 수행결과 월등히 우수한 성능 도출**

- NeurIPS-TS 를 구성하는 다양한 시나리오에서 모두 적절하게 anomaly score를 산출

Table 1: Quantitative results for Anomaly Transformer (*Ours*) in five real-world datasets. The *P*, *R* and *F1* represent the precision, recall and F1-score (as %) respectively. F1-score is the harmonic mean of precision and recall. For these three metrics, a higher value indicates a better performance.

Dataset	SMD			MSL			SMAP			SWaT			PSM		
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
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
IsolationForest	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
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
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
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
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
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
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
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
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
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
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
Ours	89.40	95.45	**92.33**	92.09	95.15	**93.59**	94.13	99.40	**96.69**	91.55	96.73	**94.07**	96.91	98.90	**97.89**

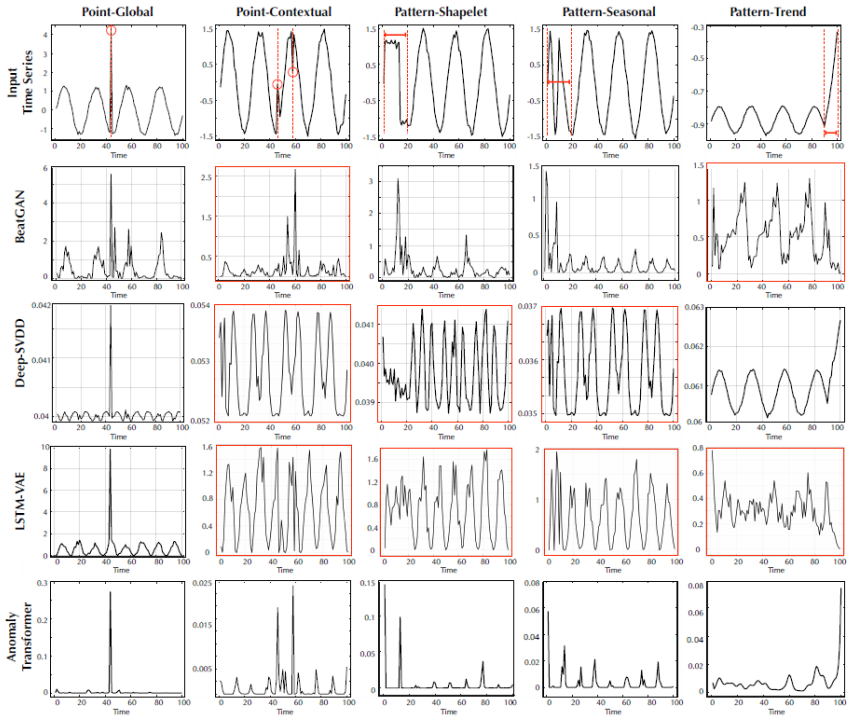


Figure 8: Visualization of learned criterion for the NeurIPS-TS dataset. Anomalies are labeled by red circles and red segments (first row). The failure cases of the baselines are bounded by red boxes.

TranAD

Introduction

- **TranAD: Deep transformer networks for anomaly detection in multivariate time series data (2022, VLDB)**
  - 2023년 1월 기준 31회 인용
  - Transformer 구조와 adversarial training을 다변량 시계열 데이터에 접목시킨 연구
  - https://github.com/imperial-qore/TranAD

TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data

Shreshth Tuli
Imperial College London
London, UK
s.tuli20@imperial.ac.uk

Giuliano Casale
Imperial College London
London, UK
g.casale@imperial.ac.uk

Nicholas R. Jennings
Loughborough University
London, UK
n.r.jennings@lboro.ac.uk

ABSTRACT

Efficient anomaly detection and diagnosis in multivariate time-series data is of great importance for modern industrial applications. However, building a system that is able to quickly and accurately pinpoint anomalous observations is a challenging problem. This is due to the lack of anomaly labels, high data volatility and the demands of ultra-low inference times in modern applications. Despite the recent developments of deep learning approaches for anomaly detection, only a few of them can address all of these challenges. In this paper, we propose TranAD, a deep transformer network based anomaly detection and diagnosis model which uses attention-based sequence encoders to swiftly perform inference with the knowledge of the broader temporal trends in the data. TranAD uses focus score-based self-conditioning to enable robust multi-modal feature extraction and adversarial training to gain stability. Additionally, model-agnostic meta learning (MAML) allows us to train the model using limited data. Extensive empirical studies on six publicly available datasets demonstrate that TranAD can outperform state-of-the-art baseline methods in detection and diagnosis performance with data and time-efficient training. Specifically, TranAD increases F1 scores by up to 17%, reducing training times by up to 99% compared to the baselines.

PVLDB Reference Format:
Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. PVLDB, 15(6): 1201 - 1214, 2022.
doi:10.14778/3514061.3514067

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/imperial-qore/TranAD.

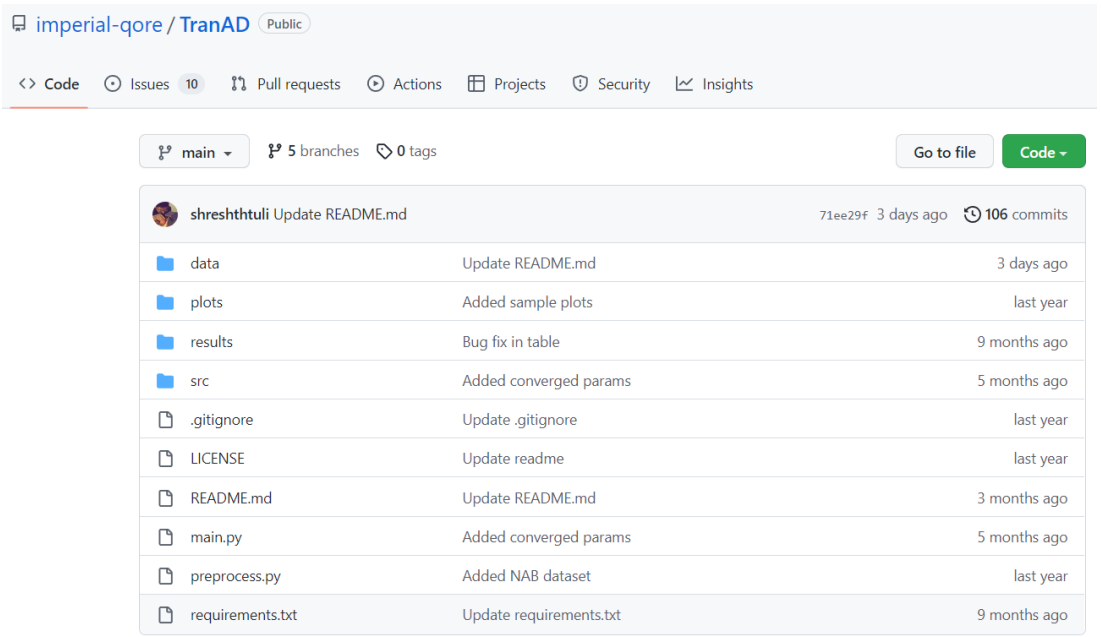
1 INTRODUCTION

Modern IT operations generate enormous amounts of high dimensional sensor data used for continuous monitoring and proper functioning of large-scale datasets. Traditionally, data mining experts have studied and highlighted data that do not follow usual trends to report faults. Such reports have been crucial for system management models for reactive fault tolerance and robust database design [47]. However, with the advent of big-data analytics and deep learning, this problem has become of interest to data mining researchers and to aid experts in handling increasing amounts of data. One particular use case is in artificial intelligence for Industry-4.0 databases, with a specific focus on service reliability [38] that has automated fault detection, recovery and management of modern systems. Detecting data-faults, or any type of behavior not conforming to the expected trends, is an active research discipline referred to as anomaly detection in multivariate time series [11]. Many data-driven industries, including ones related to distributed computing, Internet of Things (IoT), robotics and urban resource management [4, 46] are now adopting machine learning based unsupervised methods for anomaly detection.

Challenges. The problem of anomaly detection is becoming increasingly challenging in large-scale databases due to the increasing data modality [18, 28, 54]. In particular, the increasing number of sensors and devices in contemporary IoT platforms with increasing data volatility creates the requirement for significant amounts of data for accurate inference. However, due to the rising federated learning paradigm with geographically distant clusters, synchronizing databases across devices is expensive, causing limited data availability for training [48, 57]. Further, next-generation applications need ultra-fast inference speeds for quick recovery and optimal Quality of Service (QoS) [6, 50]. Time-series databases are generated using several engineering artifacts (servers, robots, etc.) that interact with the environment, humans or other systems. As a result, the data often displays both stochastic and temporal trends [45]. It thus becomes crucial to distinguish outliers due to stochasticity and only pinpoint observations that do not adhere to the observed temporal trends. Moreover, the lack of labeled data and anomaly diversity makes the problem challenging as we cannot use supervised learning models, which have shown to be effective in other areas of data mining [12]. Finally, it is not only important to detect anomalies but also the root causes, i.e., the specific data sources leading to abnormal behavior [23]. This complicates the problem further as we need to perform multi-class prediction (whether there is an anomaly and from which source if so) [60].

Existing solutions. The above discussed challenges have led to the development of a myriad of unsupervised learning solutions for automated anomaly detection. Researchers have developed reconstruction-based methods that predominantly aim to encapsulate the temporal trends and predict the time-series data in an unsupervised fashion, then use the deviation of the prediction with the ground-truth data as anomaly scores. Based on various extreme value analysis methods, such approaches classify timestamps with high anomaly scores as abnormal [4, 10, 14, 20, 28, 29, 45, 60, 62]. The way prior works generate a predicted time-series from a given one varies from one work to another. Traditional approaches, like SAND [10], use clustering and statistical analysis to

Proceedings of the VLDB Endowment, Vol. 15, No. 6 ISSN 2150-8097.
doi:10.14778/3514061.3514067



Tuli, S., Casale, G., & Jennings, N. R. (2022). Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*.

TranAD

Introduction

❖ **Transformer구조를 통해 기존 다변량 시계열 데이터에서의 anomaly detection 개선**

- 전체적인 시점정보와 지역적 시점정보 모두 반영하여 시계열 데이터가 지닌 **장단기 특징을 반영**
- 두개의 decoder를 지닌 구조로 **adversarial training**을 통해 안정적인 학습 및 불량 탐지 효과 개선
  - 정상에 대해 좀더 강건하고 일반화된 특징을 적절히 학습

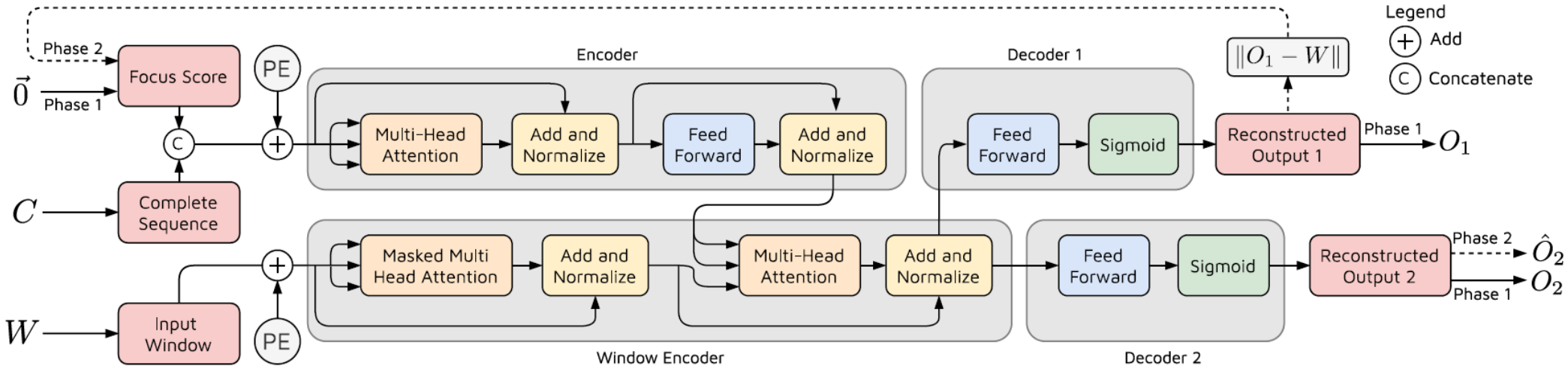


Figure 1: The TranAD Model.

TranAD

Introduction

❖ **Transformer구조를 통해 기존 다변량 시계열 데이터에서의 anomaly detection 개선**

- 전체적인 시점정보와 지역적 시점정보 모두 반영하여 시계열 데이터가 지닌 **장단기 특징을 반영**
- 두개의 decoder를 지닌 구조로 **adversarial training**을 통해 안정적인 학습 및 불량 탐지 효과 개선
  - 정상에 대해 좀더 강건하고 일반화된 특징을 적절히 학습

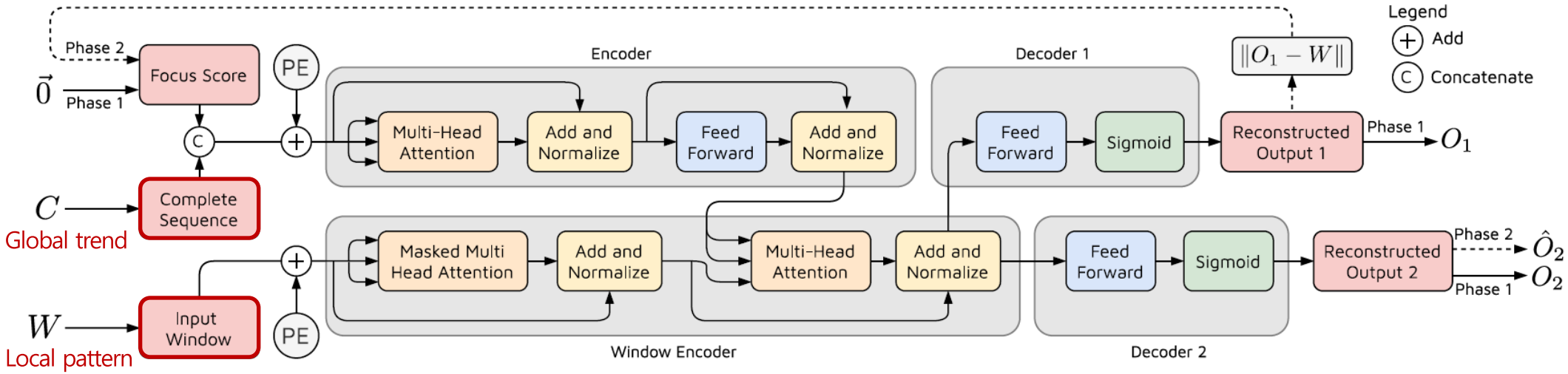


Figure 1: The TranAD Model.

TranAD

Introduction

❖ **Transformer구조를 통해 기존 다변량 시계열 데이터에서의 anomaly detection 개선**

- 전체적인 시점정보와 지역적 시점정보 모두 반영하여 시계열 데이터가 지닌 **장단기 특징을 반영**
- 두개의 decoder를 지닌 구조로 **adversarial training**을 통해 안정적인 학습 및 불량 탐지 효과 개선
  - 정상에 대해 좀더 강건하고 일반화된 특징을 적절히 학습

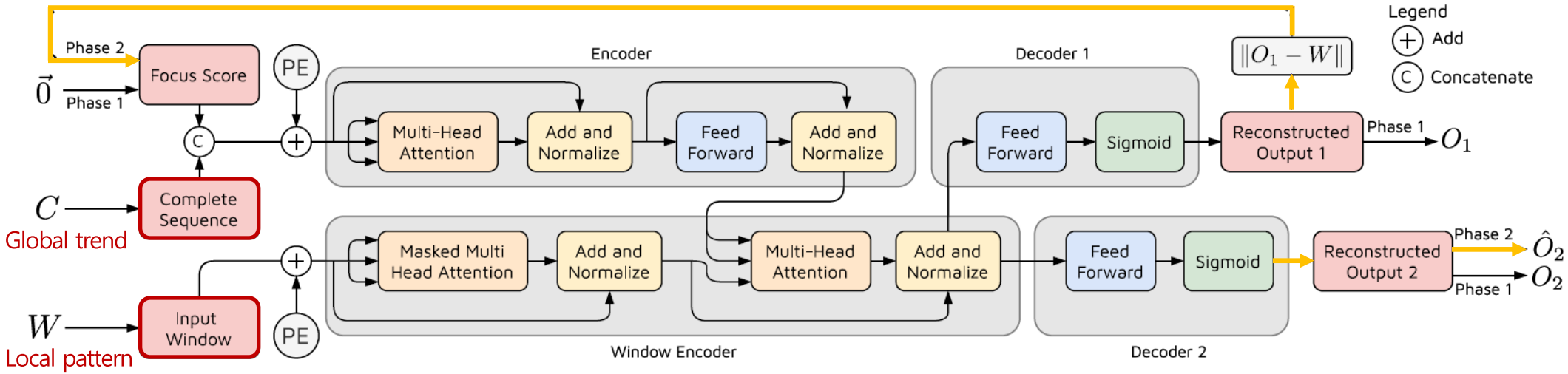


Figure 1: The TranAD Model.

TranAD

Architecture

❖ **TranAD는 encoder와 두개의 decoder로 구성**

- Encoder는 기존의 Transformer구조와 동일하며, Transformer에서의 encoder, decoder를 모두 포함한 형태
- Decoder는 두개로 구성되어 각각에 대한 reconstruction loss와 adversarial training을 적용 (USAD와 유사)

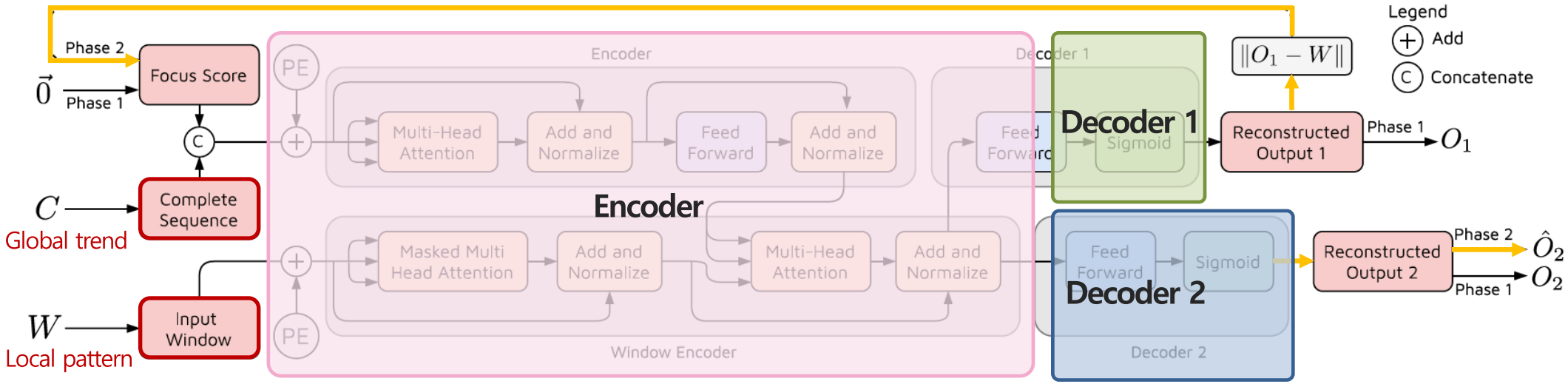


Figure 1: The TranAD Model.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2020, August). Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 3395-3404).

TranAD

① Encoder

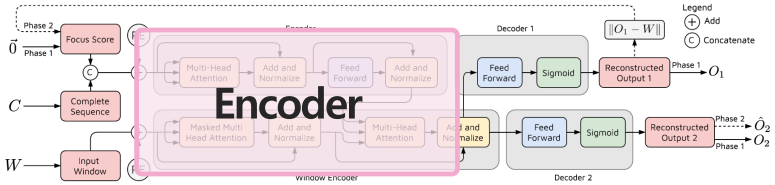
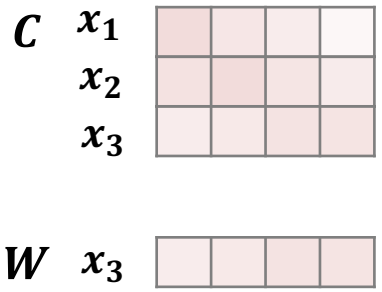
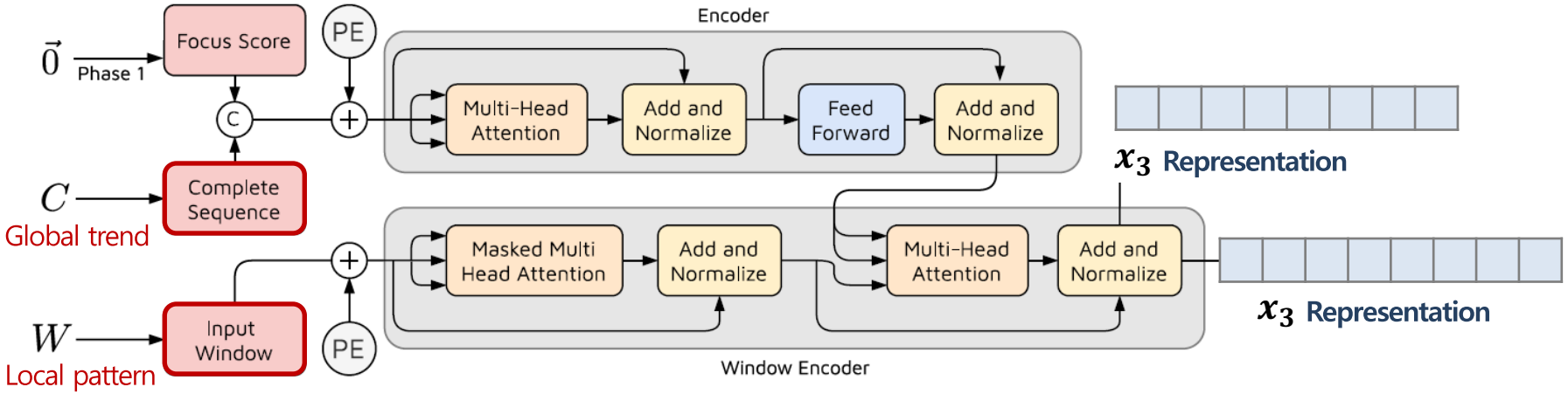


Figure 1: The TranAD Model.

❖ **Encoder는 장단기 특징을 모두 반영하여 해당 window에 대한 특징을 추출**

- 배포 코드상 일반적인 windowing이 적용된 데이터를 complete sequence로 현 시점을 input window로 정의
- 초기에는 window input과 같은 차원의 0으로 구성된 focus score를 활용하고, 차원을 증폭 시킴





TranAD

① Encoder

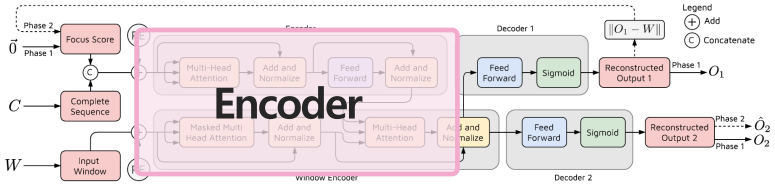
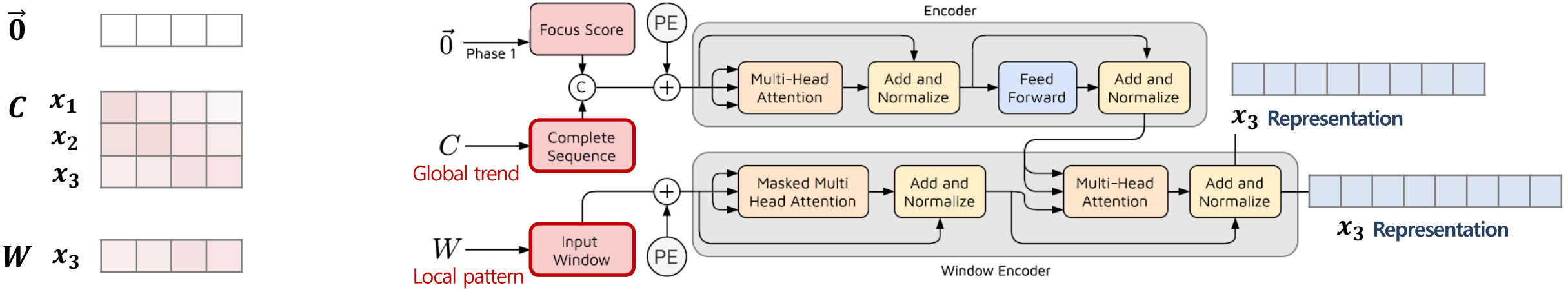


Figure 1: The TranAD Model.

❖ **Encoder는 장단기 특징을 모두 반영하여 해당 window에 대한 특징을 추출**

- 배포 코드상 일반적인 windowing이 적용된 데이터를 complete sequence로 현 시점을 input window로 정의
- 초기에는 window input과 같은 차원의 0으로 구성된 focus score를 활용하고, 차원을 증폭 시킴



TranAD

① Encoder

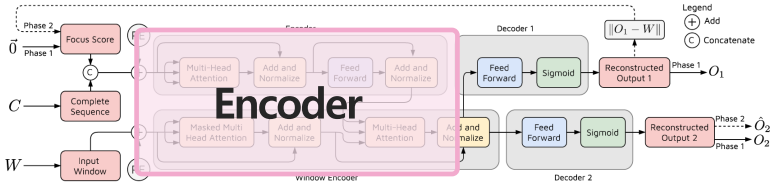
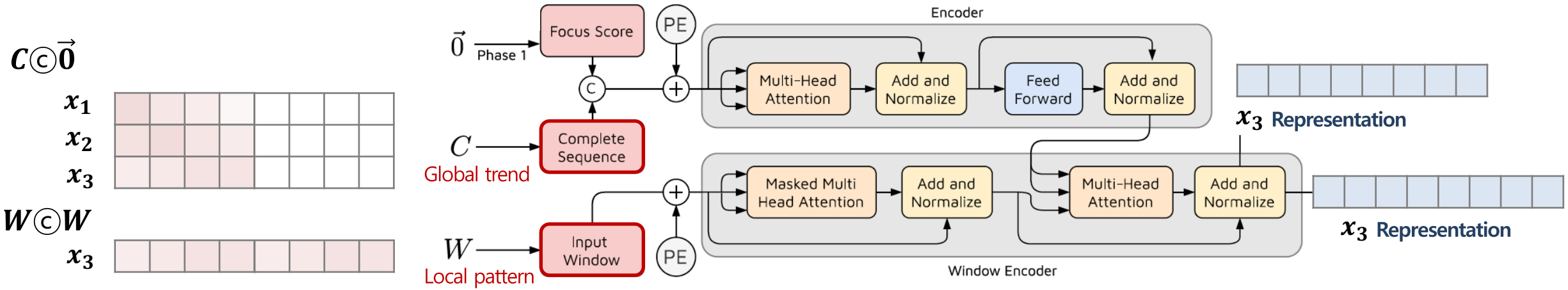


Figure 1: The TranAD Model.

❖ **Encoder는 장단기 특징을 모두 반영하여 해당 window에 대한 특징을 추출**

- 배포 코드상 일반적인 windowing이 적용된 데이터를 complete sequence로 현 시점을 input window로 정의
- 초기에는 window input과 같은 차원의 0으로 구성된 focus score를 활용하고, 차원을 증폭 시킴
- 결과적으로 encoder를 통해 과거시점을 반영하여 현 window에 대해 적절히 특징을 추출



TranAD

② Decoder

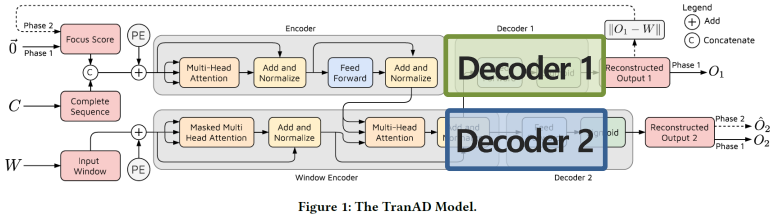
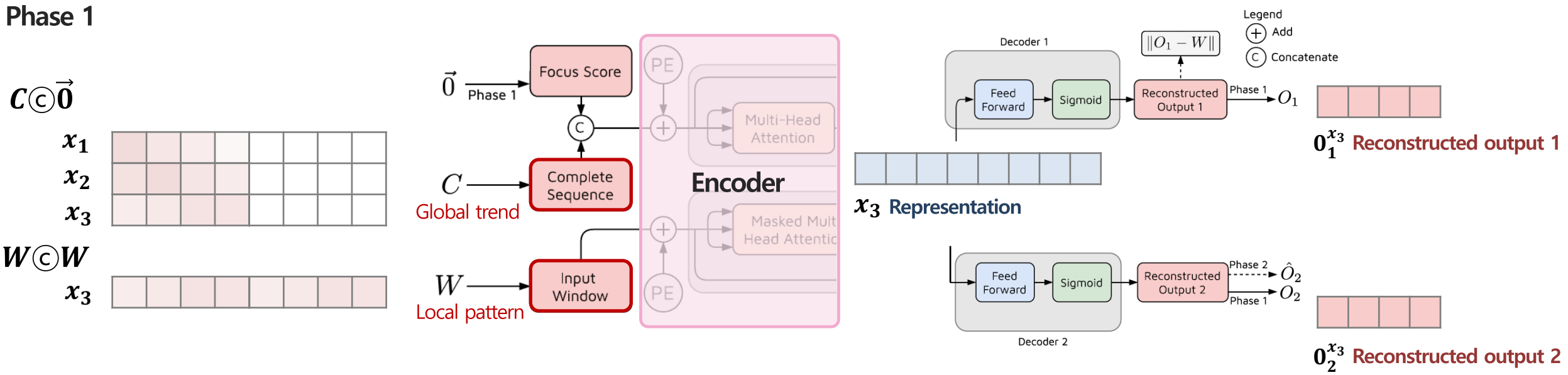


Figure 1: The TranAD Model.

❖ **Decoder는 두가지로 구성되며 모두 reconstruction loss를 연산하지만 궁극적으로 각각의 역할이 다름**

- Decoder 1 (reconstruction decoder) : 진짜 데이터는 적절하게 재구축 (진짜 같은 가짜 데이터 생성)
- Decoder 2 (prediction decoder) : 진짜 데이터는 적절하게 재구축하고, 가짜 데이터는 재구축하지 못함



TranAD

② Decoder

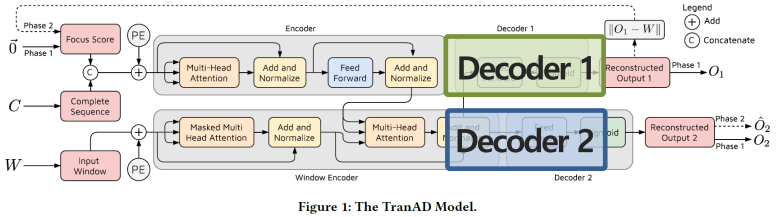
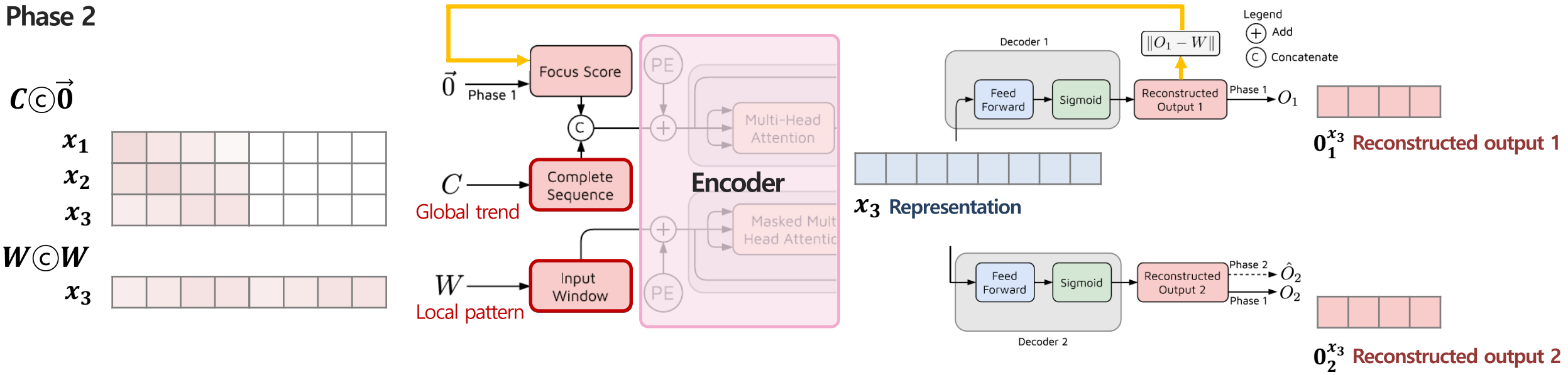


Figure 1: The TranAD Model.

❖ **Decoder는 두가지로 구성되며 모두 reconstruction loss를 연산하지만 궁극적으로 각각의 역할이 다름**

- Decoder 1 (reconstruction decoder) : 진짜 데이터는 적절하게 재구축 (진짜 같은 가짜 데이터 생성)
- Decoder 2 (prediction decoder) : 진짜 데이터는 적절하게 재구축하고, 가짜 데이터는 재구축하지 못함
- Decoder 1의 결과 값을 adversarial training의 입력 값으로 활용



TranAD

② Decoder

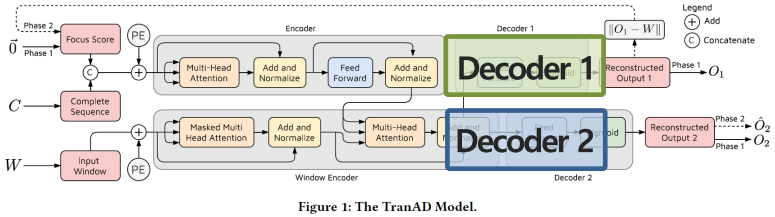
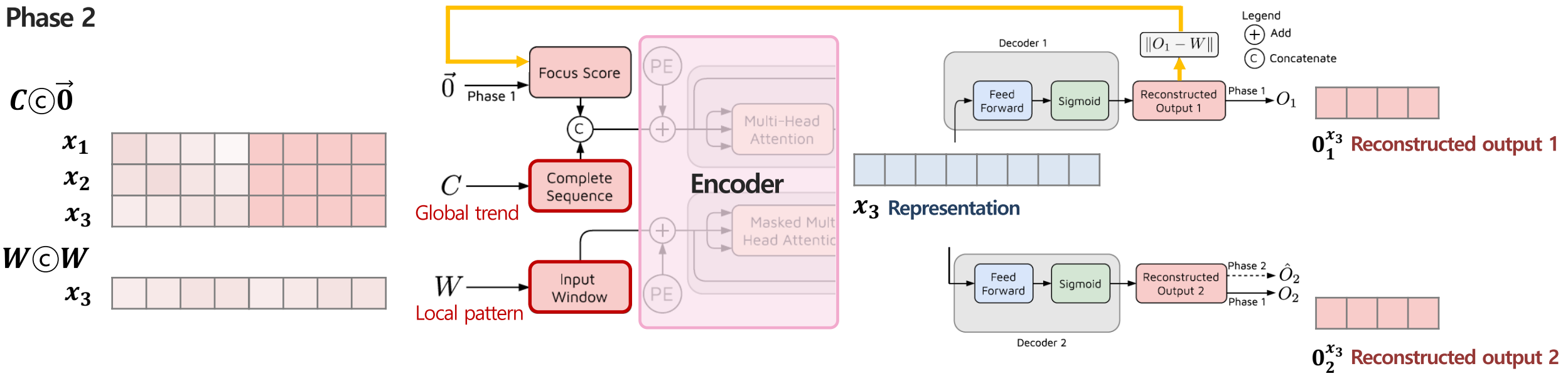


Figure 1: The TranAD Model.

❖ **Decoder는 두가지로 구성되며 모두 reconstruction loss를 연산하지만 궁극적으로 각각의 역할이 다름**

- Decoder 1 (reconstruction decoder) : 진짜 데이터는 적절하게 재구축 (진짜 같은 가짜 데이터 생성)
- Decoder 2 (prediction decoder) : 진짜 데이터는 적절하게 재구축하고, 가짜 데이터는 재구축하지 못함
- Decoder 1의 결과 값을 adversarial training의 입력 값으로 활용



TranAD

② Decoder

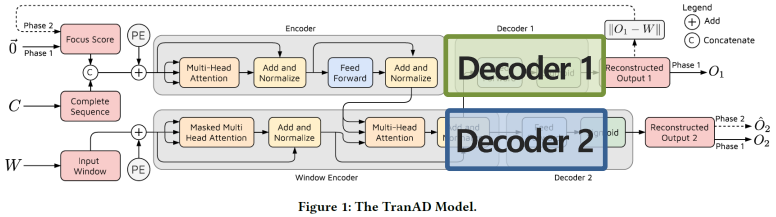
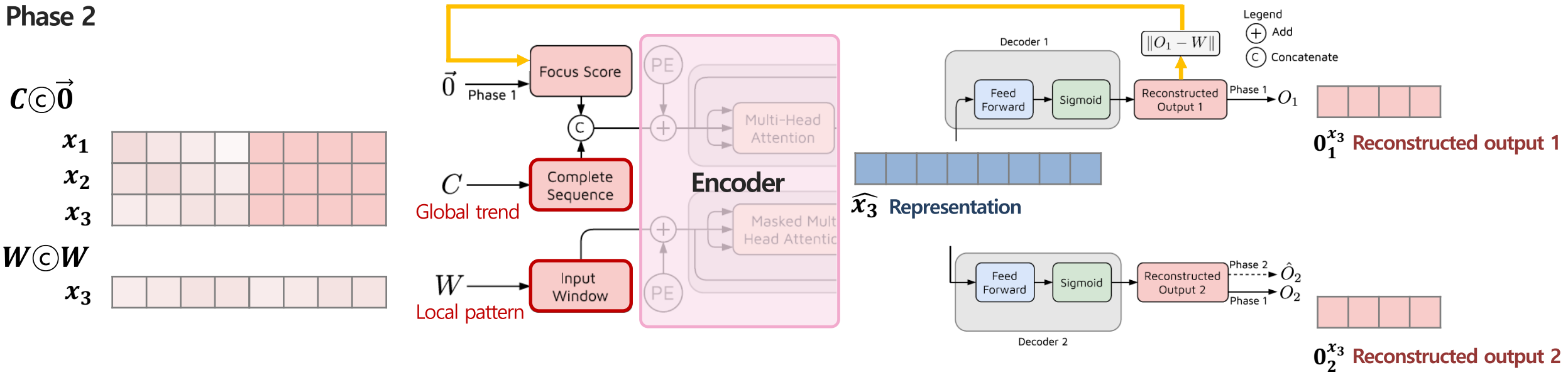


Figure 1: The TranAD Model.

❖ **Decoder는 두가지로 구성되며 모두 reconstruction loss를 연산하지만 궁극적으로 각각의 역할이 다름**

- Decoder 1 (reconstruction decoder) : 진짜 데이터는 적절하게 재구축 (진짜 같은 가짜 데이터 생성)
- Decoder 2 (prediction decoder) : 진짜 데이터는 적절하게 재구축하고, 가짜 데이터는 재구축하지 못함
- Decoder 1의 결과 값을 adversarial training의 입력 값으로 활용



TranAD

② Decoder

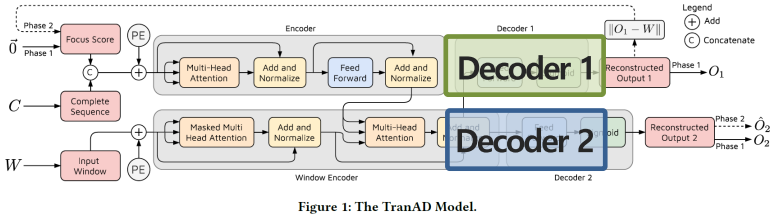
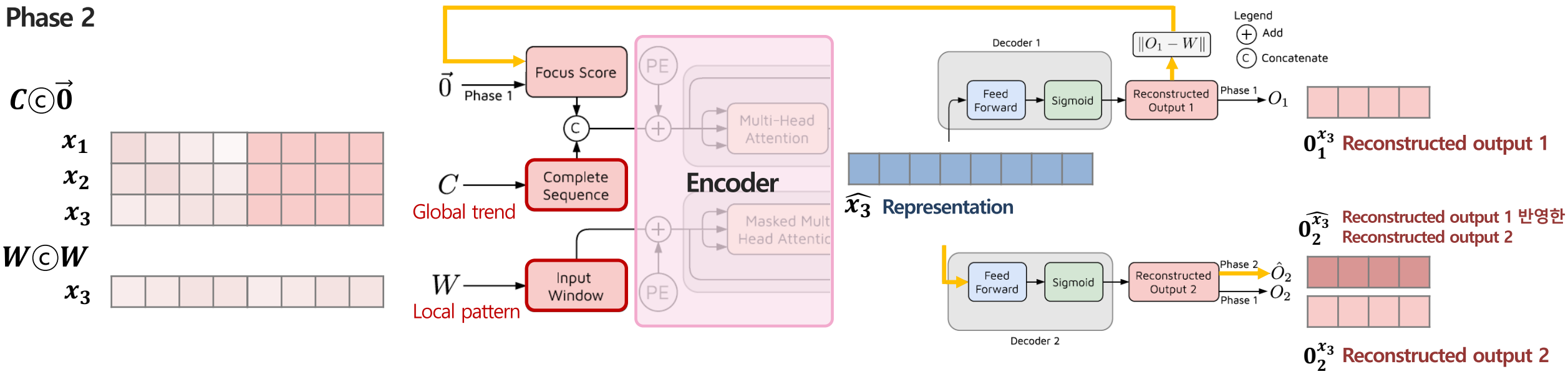


Figure 1: The TranAD Model.

❖ **Decoder는 두가지로 구성되며 모두 reconstruction loss를 연산하지만 궁극적으로 각각의 역할이 다름**

- Decoder 1 (reconstruction decoder) : 진짜 데이터는 적절하게 재구축 (진짜 같은 가짜 데이터 생성)
- Decoder 2 (prediction decoder) : 진짜 데이터는 적절하게 재구축하고, 가짜 데이터는 재구축하지 못함
- Decoder 1의 결과 값을 adversarial training의 입력 값으로 활용

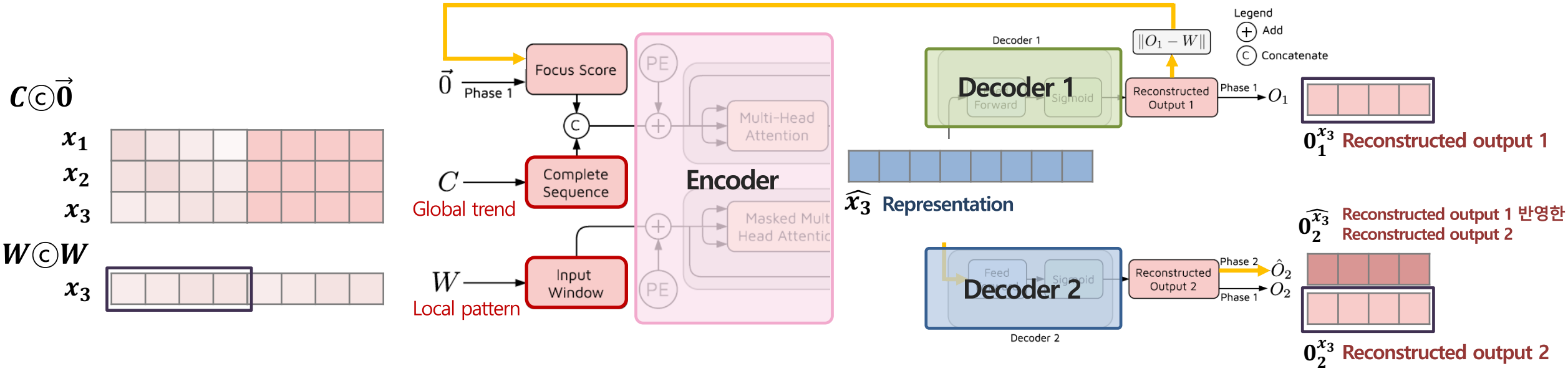


TranAD

Loss function

❖ **Reconstruction loss와 adversarial loss가 병합된 형태로 최종 손실함수 정의**

- Reconstruction loss : 입력 데이터와 재구축 데이터가 '각각' 유사해지도록 학습
- $L_1^{recon} = \|O_1 - W\|_2$
- $L_2^{recon} = \|O_2 - W\|_2$

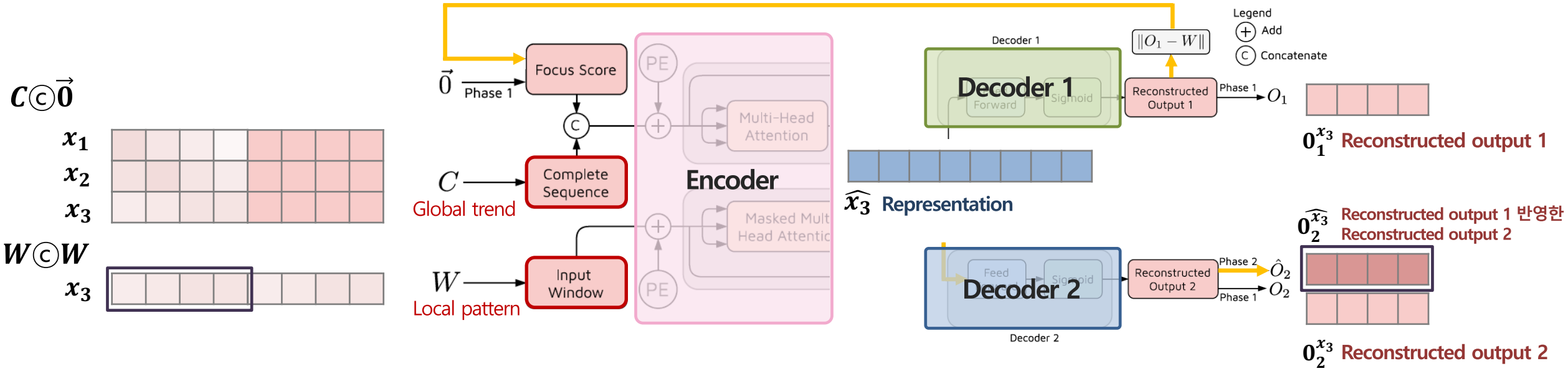


TranAD

Loss function

❖ **Reconstruction loss와 adversarial loss가 병합된 형태로 최종 손실함수 정의**

- Adversarial loss : Decoder 1은 진짜 같은 가짜 데이터를 생성하고, decoder 2는 진짜 데이터와 가짜데이터를 구별
- $L_1^{adversarial} = +\left\|\widehat{O_2} - W\right\|_2$ Minimize (차이가 작아지도록, 유사하도록)
- $L_2^{adversarial} = -\left\|\widehat{O_2} - W\right\|_2$ Maximize (차이가 크도록, 구별되도록)



TranAD

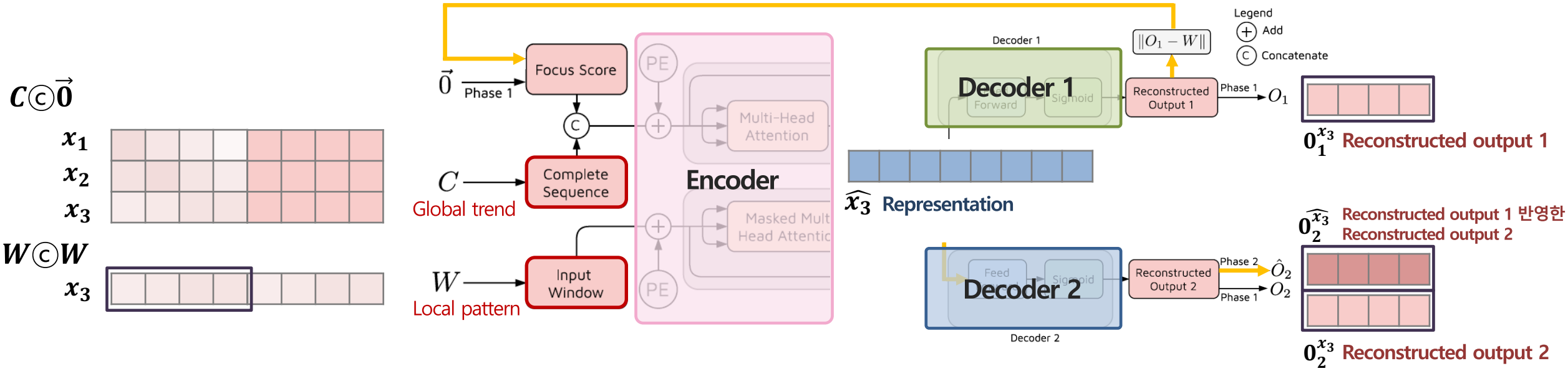
Loss function

❖ Reconstruction loss와 adversarial loss가 병합된 형태로 최종 손실함수 정의

- Ecoder – Decoder 1: $L_{Total_1} = \frac{1}{n}\|O_1 - W\|_2 + (1-\frac{1}{n})\|\widehat{O_2} - W\|_2$ **Minimize (차이가 작아지도록, 유사하도록)**
- Encoder – Decoder 2 :$L_{Total_2} = \frac{1}{n}\|O_2 - W\|_2 - (1-\frac{1}{n})\|\widehat{O_2} - W\|_2$ **Maximize (차이가 크도록, 구별되도록)**

각각 복원이 잘되도록

n: epoch
학습 초반에는 reconstruction loss비율 크게

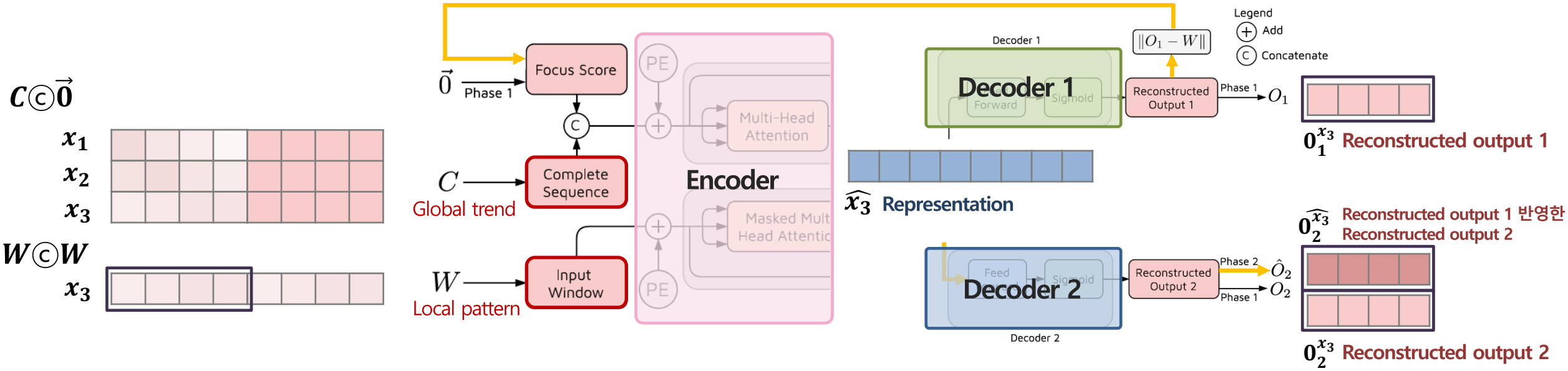


TranAD

Anomaly score

❖ 정상이라면 복원(O_1)이 잘 되며, 불량이라면 복원(O_1)이 잘 안됨

- Anomaly score: $S = \frac{1}{2}\|O_1 - \hat{W}\|_2 + \frac{1}{2}\|\hat{O_2} - \hat{W}\|_2$
- 정상의 경우, O_1 & $\hat{O_2}$ 모두 실제 값($\hat{W}$)과 차이 작음
- 불량의 경우, O_1 & $\hat{O_2}$ 모두 실제 값($\hat{W}$)과 차이 큼



TranAD

Experiments

❖ 9개의 공용데이터에 적용하여 우수성을 입증

- 입력: windowing process를 수행하여 sub-series형태로 입력 [Batch, Seq_len, Var]
- 출력: 각 시점별로 anomaly score / 상태 여부 출력 [Batch, Seq_len, 1]

Table 1: Dataset Statistics

Dataset	Train	Test	Dimensions	Anomalies (%)
NAB	4033	4033	1 (6)	0.92
UCR	1600	5900	1 (4)	1.88
MBA	100000	100000	2 (8)	0.14
SMAP	135183	427617	25 (55)	13.13
MSL	58317	73729	55 (3)	10.72
SWaT	496800	449919	51 (1)	11.98
WADI	1048571	172801	123 (1)	5.99
SMD	708405	708420	38 (4)	4.16
MSDS	146430	146430	10 (1)	5.37

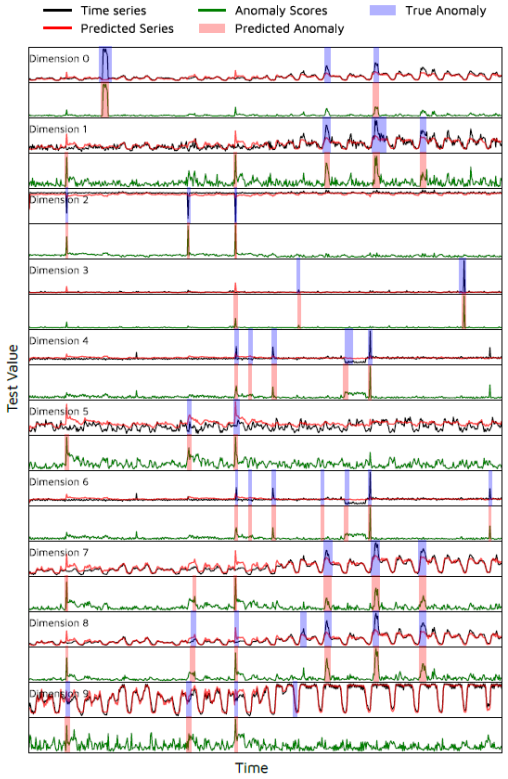
TranAD

Experiments

- ❖ 다양한 비교실험 수행결과 우수한 성능 도출
- ❖ Anomaly Transformer와 비슷한 시기에 수행된 연구로 서로 비교실험 결과는 없으나 유사한 성능 도출

Table 2: Performance comparison of TranAD with baseline methods on the complete dataset. P: Precision, R: Recall, AUC: Area under the ROC curve, F1: F1 score with complete training data. The best F1 and AUC scores are highlighted in bold.

Method	NAB				UCR				MBA			
	P	R	AUC	F1	P	R	AUC	F1	P	R	AUC	F1
MERLIN	0.8013	0.7262	0.8414	0.7619	0.7542	0.8018	0.8984	0.7542	0.9846	0.4913	0.7828	0.6555
LSTM-NDT	0.6400	0.6667	0.8322	0.6531	0.5231	0.8294	0.9781	0.5231	0.9207	0.9718	0.9780	0.9456
DAGMM	0.7622	0.7292	0.8572	0.7453	0.5337	0.9718	0.9916	0.5337	0.9475	0.9900	0.9858	0.9683
OmniAnomaly	0.8421	0.6667	0.8330	0.7442	0.8346	0.9999	0.9981	0.8346	0.8561	1.0000	0.9570	0.9225
MSCRED	0.8522	0.6700	0.8401	0.7502	0.5441	0.9718	0.9920	0.5441	0.9272	1.0000	0.9799	0.9623
MAD-GAN	0.8666	0.7012	0.8478	0.7752	0.8538	0.9891	0.9984	0.8538	0.9396	1.0000	0.9836	0.9689
USAD	0.8421	0.6667	0.8330	0.7442	0.8952	1.0000	0.9989	0.8952	0.8953	0.9989	0.9701	0.9443
MTAD-GAT	0.8421	0.7272	0.8221	0.7804	0.7812	0.9972	0.9978	0.7812	0.9018	1.0000	0.9721	0.9484
CAE-M	0.7918	0.8019	0.8019	0.7968	0.6981	1.0000	0.9957	0.6981	0.8442	0.9997	0.9661	0.9154
GDN	0.8129	0.7872	0.8542	0.7998	0.6894	0.9988	0.9959	0.6894	0.8832	0.9892	0.9528	0.9332
TranAD	0.8889	0.9892	**0.9541**	**0.9364**	0.9407	1.0000	**0.9994**	**0.9407**	0.9569	1.0000	**0.9885**	**0.9780**
Method	SMAP				MSL				SWaT			
	P	R	AUC	F1	P	R	AUC	F1	P	R	AUC	F1
MERLIN	0.1577	0.9999	0.7426	0.2725	0.2613	0.4645	0.6281	0.3345	0.6560	0.2547	0.6175	0.3669
LSTM-NDT	0.8523	0.7326	0.8602	0.7879	0.6288	1.0000	0.9532	0.7721	0.7778	0.5109	0.7140	0.6167
DAGMM	0.8069	0.9891	0.9885	0.8888	0.7363	1.0000	0.9716	0.8482	0.9933	0.6879	0.8436	0.8128
OmniAnomaly	0.8130	0.9419	0.9889	0.8728	0.7848	0.9924	0.9782	0.8765	0.9782	0.6957	0.8467	0.8131
MSCRED	0.8175	0.9216	0.9821	0.8664	0.8912	0.9862	0.9807	0.9363	0.9992	0.6770	0.8433	0.8072
MAD-GAN	0.8157	0.9216	0.9891	0.8654	0.8516	0.9930	0.9862	0.9169	0.9593	0.6957	0.8463	0.8065
USAD	0.7480	0.9627	0.9890	0.8419	0.7949	0.9912	0.9795	0.8822	0.9977	0.6879	0.8460	0.8143
MTAD-GAT	0.7991	0.9991	0.9844	0.8880	0.7917	0.9824	0.9899	0.8768	0.9718	0.6957	0.8464	0.8109
CAE-M	0.8193	0.9567	0.9901	0.8827	0.7751	1.0000	0.9903	0.8733	0.9697	0.6957	0.8464	0.8101
GDN	0.7480	0.9891	0.9864	0.8518	0.9308	0.9892	0.9814	**0.9591**	0.9697	0.6957	0.8462	0.8101
TranAD	0.8043	0.9999	**0.9921**	**0.8915**	0.9038	0.9999	**0.9916**	0.9494	0.9760	0.6997	**0.8491**	**0.8151**
Method	WADI				SMD				MSDS			
	P	R	AUC	F1	P	R	AUC	F1	P	R	AUC	F1
MERLIN	0.0636	0.7669	0.5912	0.1174	0.2871	0.5804	0.7158	0.3842	0.7254	0.3110	0.5022	0.4353
LSTM-NDT	0.0138	0.7823	0.6721	0.0271	0.9736	0.8440	0.9671	0.9042	0.9999	0.8012	0.8013	0.8896
DAGMM	0.0760	0.9981	0.8563	0.1412	0.9103	0.9914	0.9954	0.9491	0.9891	0.8026	0.9013	0.8861
OmniAnomaly	0.3158	0.6541	0.8198	0.4260	0.8881	0.9985	0.9946	0.9401	1.0000	0.7964	0.8982	0.8867
MSCRED	0.2513	0.7319	0.8412	0.3741	0.7276	0.9974	0.9921	0.8414	1.0000	0.7983	0.8943	0.8878
MAD-GAN	0.2233	0.9124	0.8026	0.3588	0.9991	0.8440	0.9933	0.9150	0.9982	0.6107	0.8054	0.7578
USAD	0.1873	0.8296	0.8723	0.3056	0.9060	0.9974	0.9933	0.9495	0.9912	0.7959	0.8979	0.8829
MTAD-GAT	0.2818	0.8012	0.8821	0.4169	0.8210	0.9215	0.9921	0.8683	0.9919	0.7964	0.8982	0.8835
CAE-M	0.2782	0.7918	0.8728	0.4117	0.9082	0.9671	0.9783	0.9367	0.9908	0.8439	0.9013	0.9115
GDN	0.2912	0.7931	0.8777	0.4260	0.7170	0.9974	0.9924	0.8342	0.9989	0.8026	**0.9105**	0.8900
TranAD	0.3529	0.8296	**0.8968**	**0.4951**	0.9262	0.9974	**0.9974**	**0.9605**	0.9999	0.8626	0.9013	**0.9262**



Conclusions

❖ **시계열성과 복잡한 변수사이 관계를 모두 반영하기 위해 Transformer기반 방법론이 활발히 제안됨**

❖ **Anomaly Transformer (Transformer encoder 구조만 활용)**

- 불량 데이터는 전체 시계열성과는 상관관계가 적지만 **인접 시점과는 상관관계가 크다**는 특징을 활용
- 전반적인 시계열성을 고려하는 **series association**, 인접시점과의 시계열성을 고려하는 **prior association** 제안
- **두 association의 유사성 지표인 association discrepancy를 제안**하고, minmax전략을 통해 학습
- 결과적으로 인접하지 않은 영역에도 높은 가중치(attention)을 부여하여 이상치 탐지 명확히 구분

❖ **TranAD (Transformer encoder-decoder 구조 모두 활용)**

- 전체 시계열성(global trend)을 반영하여 지역적 시계열성(local pattern)을 효과적으로 요약
- 두개의 decoder구조를 통해 **adversarial training**수행
- 결과적으로 정상에 대해 강건하고 일반화성능이 높게 학습되어 이상치 탐지 명확히 구분

References

❖ **Time Series Anomaly Detection**

- Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. IEEE Access, 9, 120043-120065.
- Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2022). Deep Learning for Time Series Anomaly Detection: A Survey. arXiv e-prints, arXiv-2211.

❖ **Transformer variants**

- Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. AI Open.

References

❖ **Transformer**

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
- https://jalammar.github.io/illustrated-transformer/
- https://github.com/pilsung-kang/Text-Analytics/blob/master/08%20Seq2Seq%20Learning%20and%20Pre-trained%20Models/08-2_Transformer.pdf
- https://wikidocs.net/31379
- https://blog.promedius.ai/transformer/
- https://moondol-ai.tistory.com/460

References

❖ **Anomaly Transformer**

- Xu, J., Wu, H., Wang, J., & Long, M. (2021). Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642.
- https://github.com/thuml/Anomaly-Transformer
- https://www.youtube.com/watch?v=C3dphckvyn0

❖ **TranAD**

- Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284.
- https://github.com/imperial-qore/TranAD
- https://www.youtube.com/watch?v=b2fSzneXPsg

Thank you