Channel Independent Strategy for Time Series Forecasting

2024.10.11

Data Mining & Quality Analytics Lab.

조광은

발표자 소개

❖ **조광은 (Kwangeun Cho)**

- 고려대학교 산업경영공학과 Data Mining & Quality Analytics Lab.
- M.S. Student (2024.03 ~ Present)
- 지도교수: 김성범 교수님

❖ **Research Interest**

- Multivariate Time Series Analysis

❖ **Contact**

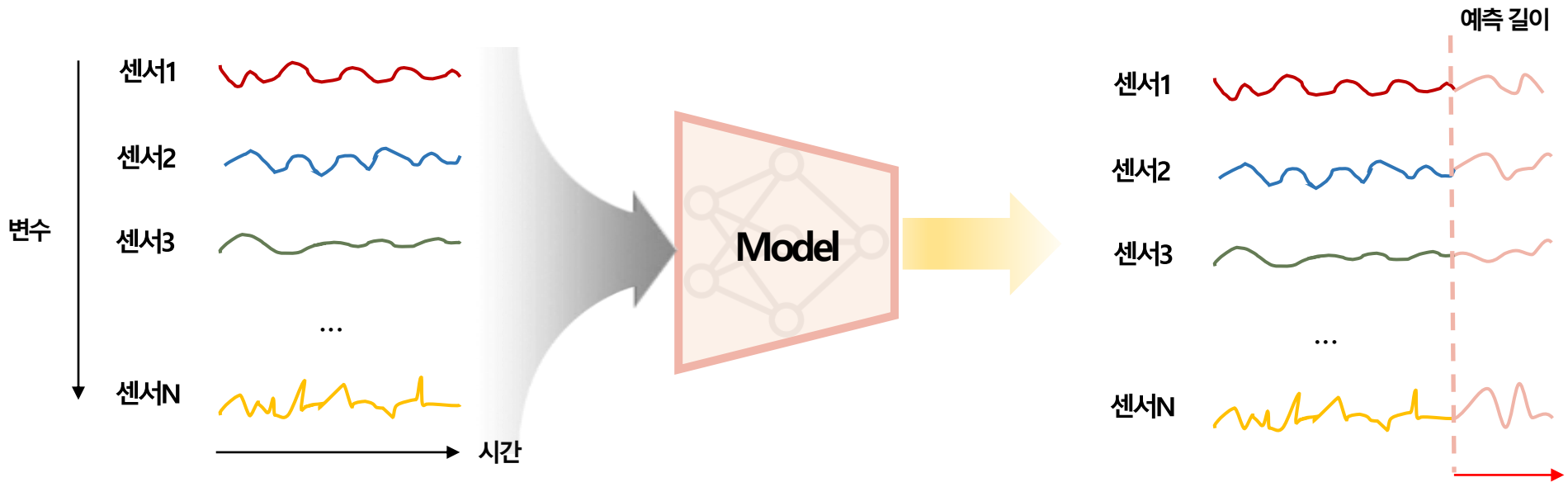
- E-mail: chopol98@korea.ac.kr

목차

Background
Multivariate Time Series Forecasting

❖ **Multivariate Time Series Forecasting**

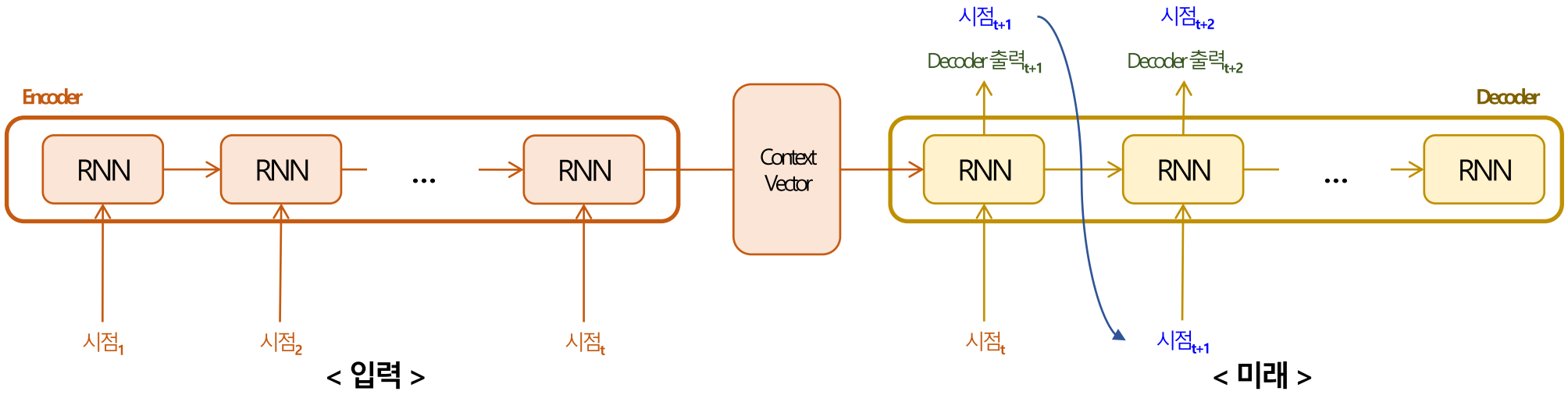
- 다변량 시계열 데이터: 여러 개의 변수가 시간의 흐름에 따라 **순서대로 관측**되는 데이터
- **다변량 시계열 예측:** 다변량 시계열 데이터의 **과거 패턴을 기반**으로 Target의 미래 흐름을 예측하는 문제



Background
Multivariate Time Series Forecasting

❖ **Multivariate Time Series Forecasting**

- 시계열 데이터는 **순서를 가지고 있는 데이터**
- 초기 딥러닝 기반 시계열 예측은 sequential data를 처리하는 데 뛰어난 성능을 보이는 RNN을 사용



Background
Multivariate Time Series Forecasting

❖ **The Problems of Multivariate Time Series Forecasting using RNN**

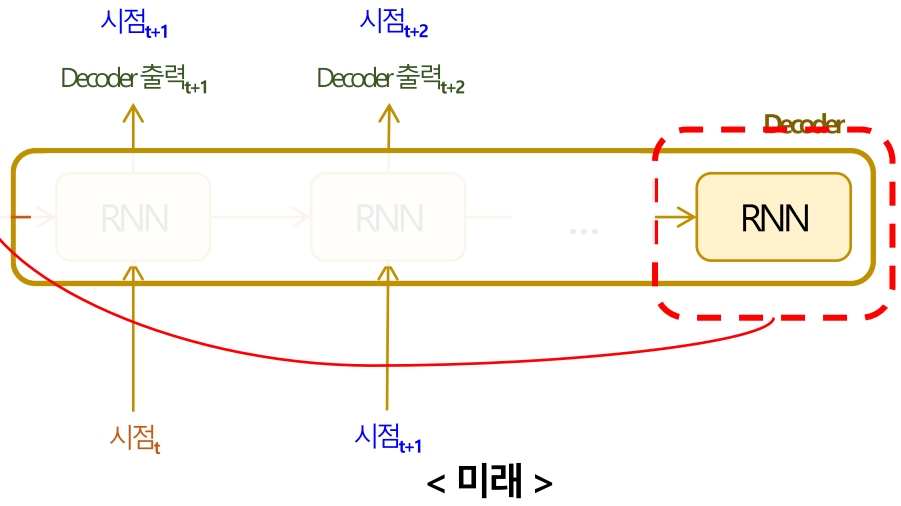
- RNN 기반 시계열 예측의 두 가지 문제: **Gradient Vanishing/Explosion**, Fixed Context Vector
- Gradient Vanishing/Explosion: 특정 시점의 hidden state를 업데이트하는 과정에서 RNN 구조의 한계로 기울기 소실/폭발 문제 발생

특정 시점의 hidden state의 gradient

$$\partial_{w_h} h_t = \sum_{i=1}^{t-1} \left(\prod_{j=i+1}^{t} (\partial_{h_{j-1}} f_{w_h}(x_j, h_{j-1})) \right) \partial_{w_h} f_{w_h}(x_i, h_{i-1}) + \partial_{w_h} f_{w_h}(x_t, h_{t-1})$$

이전 모든 시점의 gradient를 곱해주는 형태로 정의

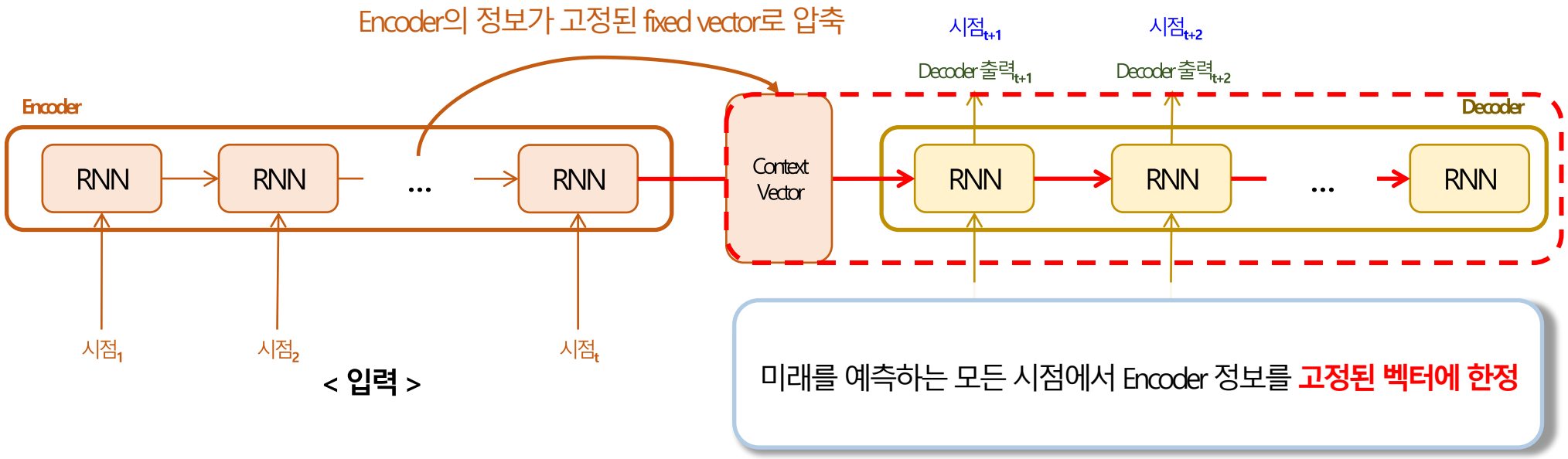
특정 state의 기울기가 0에 가까울 경우: **기울기 소실**
특정 state의 기울기가 1을 초과할 경우: **기울기 폭발**



Background
Multivariate Time Series Forecasting

❖ **The Problems of Multivariate Time Series Forecasting using RNN**

- RNN 기반 시계열 예측의 두 가지 문제: Gradient Vanishing/Explosion, **Fixed Context Vector**
- Fixed Context Vector: 예측하고자 하는 모든 timestep에서 **고정된 context vector 이용**



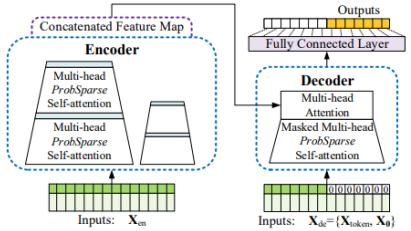
Background
Multivariate Time Series Forecasting

❖ **Multivariate Time Series Forecasting using Transformer**

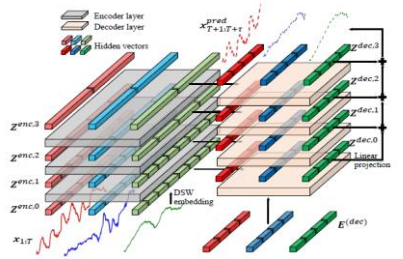
- 이와 같은 이유로 RNN based model이 아닌 다른 모델을 활용하여 시계열 예측을 시도
- 당시 NLP, CV 도메인에서 뛰어난 성능을 보이던 **Transformer를 활용한 방법론**들이 활발하게 연구

Model Type	Short Description	Date	Reference
LogTrans	Local and LogSparse Attention	2019	[61]
Reformer	Only Similar Queries and Keys Are Compared	2020	[57]
Informer	Uses Selected Query Prototypes	2021	[150]
Autoformer	Replaces Self-Attention with Auto-Correlation	2021	[130]
Pyraformer	Hierarchical/Pyramidal Attention	2021	[70]
FEDformer	Series Decomposition and Use of Frequency Domain	2022	[152]
Non-stationary TRF	Series Stationarization and Use of De-stationary Attention	2022	[72]
Triformer	Triangular Structure for Layer Shrinking	2022	[18]
CrossFormer	Cross-channel Modeling	2023	[145]

Informer(2021)



Crossformer(2023)



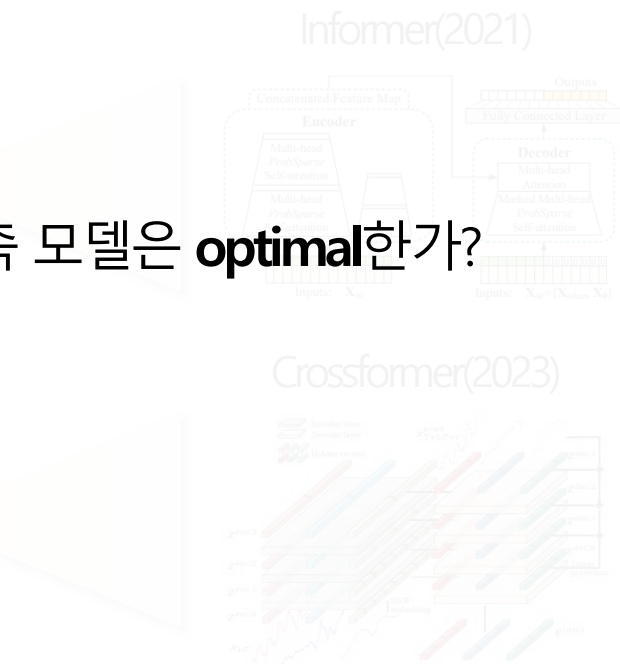
Miller, J. A., Aldosari, M., Saeed, F., Barna, N. H., Rana, S., Arpinar, I. B., & Liu, N. (2024). A survey of deep learning and foundation models for time series forecasting. *arXiv preprint arXiv:2401.13912*

Background
Multivariate Time Series Forecasting

❖ **Multivariate Time Series Forecasting using Transformer**

- 이와 같은 이유로 RNN based model이 아닌 다른 모델을 활용하여 시계열 예측을 시도
- 당시 NLP, CV 도메인에서 뛰어난 성능을 보이던 **Transformer를 활용한 방법론**들이 활발하게 연구

Model Type	Short Description	Date	Reference
LogTrans	Local and LogSparse Attention	2019	[61]
Reformer	Only Similar Queries and Keys Are Compared	2020	[57]
Informer	Uses Selected Query Prototypes	2021	[150]
Autoformer	Replaces Self-Attention	2021	[140]
Pyraformer	Hierarchical/Pyramidal Attention	2021	[70]
FEDformer	Series Decomposition and Use of Frequency Domain	2022	[152]
Non-stationary TRF	Series Stationarization and Use of De-stationary Attention	2022	[72]
Triformer	Triangular Structure for Layer Shrinking	2022	[18]
CrossFormer	Cross-channel Modeling	2023	[145]



Q. Transformer를 기반으로 한 시계열 예측 모델은 **optimal**한가?

Miller, J. A., Aldosari, M., Saeed, F., Barna, N. H., Rana, S., Arpinar, I. B., & Liu, N. (2024). A survey of deep learning and foundation models for time series forecasting. *arXiv preprint arXiv:2401.13912*

Background
Multivariate Time Series Forecasting

LTSF-Linear	PatchTST	The Capacity and Robustness Trade-off
문제 제기 Transformer는 시계열 예측 문제에서 최적의 모델이 아니다.	**반박** Channel Indepenent Strategy와 결합하면 시계열 예측에서 Transformer는 최적의 모델이다	**원인** 왜 Channel independent Strategy 는 시계열 예측에서 뛰어난 성능을 발휘하는가?

Background
Multivariate Time Series Forecasting

LTSF-Linear

문제 제기

Transformer는
시계열 예측 문제에서 최적의
모델이 아니다.

PatchTST

반박

Channel Indepenent Strategy와
결합하면
시계열 예측에서 Transformer는
최적의 모델이다

The Capacity and Robustness Trade-off

원인

왜 Channel independent Strategy
는
시계열 예측에서 뛰어난
성능을 발휘하는가?

Method

Are Transformers Effective for Time Series Forecasting?

❖ **Are Transformers Effective for Time Series Forecasting? (2023, AAAI)**

- 2024년 10월 기준 1,127회 인용
- 간단한 Linear baseline model을 통해 Transformer가 시계열 예측에 있어 최적의 모델이 아님을 주장

The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Are Transformers Effective for Time Series Forecasting?

Ailing Zeng[1,2*], **Muxi Chen**[1*], **Lei Zhang**[2], **Qiang Xu**[1]

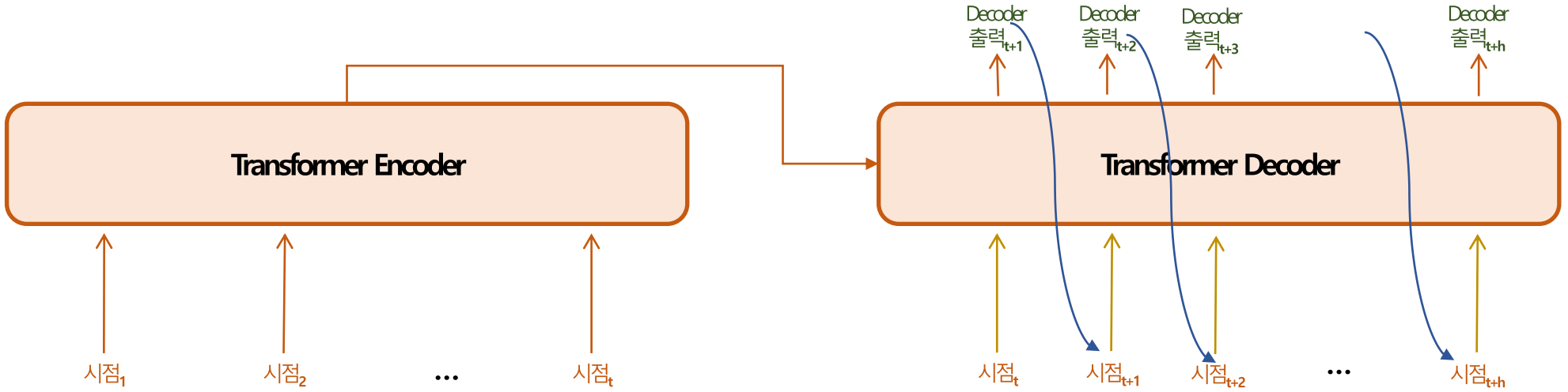
[1]The Chinese University of Hong Kong
[2]International Digital Economy Academy
{zengailing, leizhang}@idea.edu.cn,{mxchen21, qxu}@cse.cuhk.edu.hk

Method

Are Transformers Effective for Time Series Forecasting?

❖ **Why Transformers is Ineffective in Time Series Forecasting?**

- IMS Design: Transformer Decoder의 Autoregressive한 성질 때문에 **error accumulation**을 불러일으킨다.
- Vanilla Transformer의 Inference 단계에서 발생

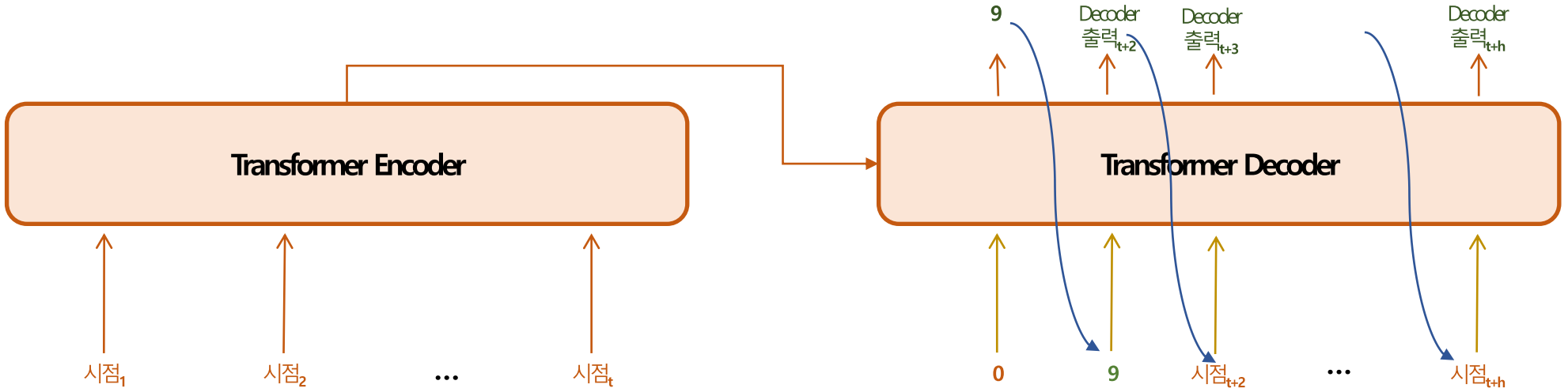


Method

Are Transformers Effective for Time Series Forecasting?

❖ **Why Transformers is Ineffective in Time Series Forecasting?**

- IMS Design: Transformer Decoder의 Autoregressive한 성질 때문에 **error accumulation**을 불러일으킨다.
- Vanilla Transformer의 Inference 단계에서 발생



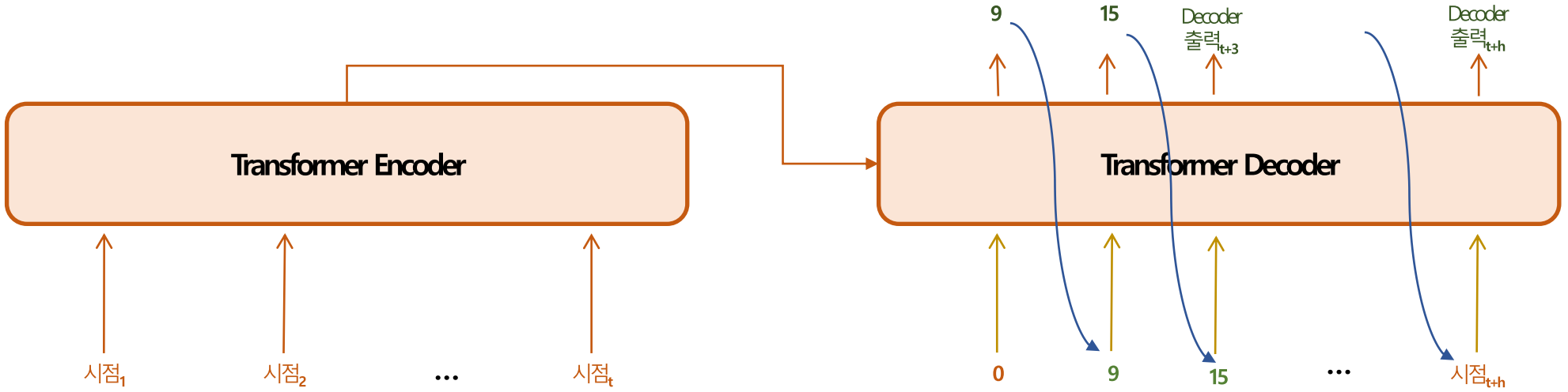
시점	t+1	t+2		t+h
실제값	1	3	...	9
예측값	9	15	...	50

Method

Are Transformers Effective for Time Series Forecasting?

❖ **Why Transformers is Ineffective in Time Series Forecasting?**

- IMS Design: Transformer Decoder의 Autoregressive한 성질 때문에 **error accumulation**을 불러일으킨다.
- Vanilla Transformer의 Inference 단계에서 발생



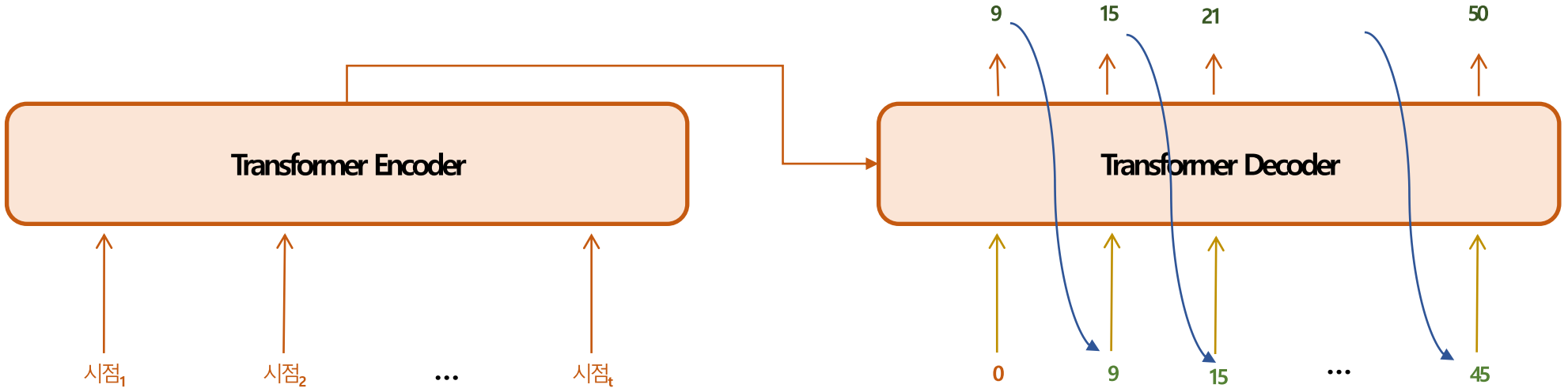
시점	t+1	t+2		t+h
실제값	1	3	...	9
예측값	9	15	...	50

Method

Are Transformers Effective for Time Series Forecasting?

❖ **Why Transformers is Ineffective in Time Series Forecasting?**

- IMS Design: Transformer Decoder의 Autoregressive한 성질 때문에 **error accumulation**을 불러일으킨다.
- Vanilla Transformer의 Inference 단계에서 발생



시점	t+1	t+2		t+h
실제값	1	3	...	9
예측값	9	15	...	50

Method

Are Transformers Effective for Time Series Forecasting?

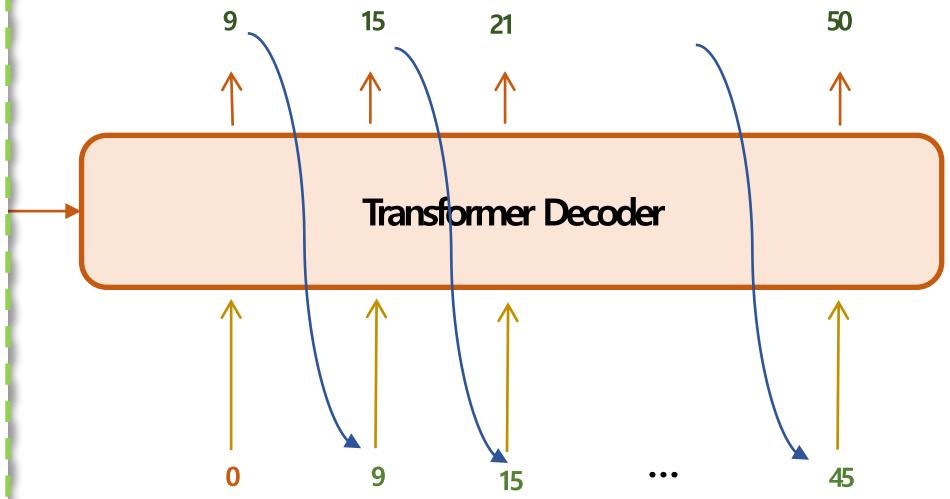
❖ Why Transformers is Ineffective in Time Series Forecasting?

- IMS Design: Transformer Decoder의 Autoregressive한 성질 때문에 **error accumulation**을 불러일으킨다.
- Transformer의 Inference 단계에서 발생

Error Accumulation

초기 시점에서 예측값을 잘못 예측했을 경우,
잘못된 값을 참조하기 때문에
뒤 예측값은 더 부정확해진다.

긴 기간을 예측해야 할 경우 예측 성능에 악영향



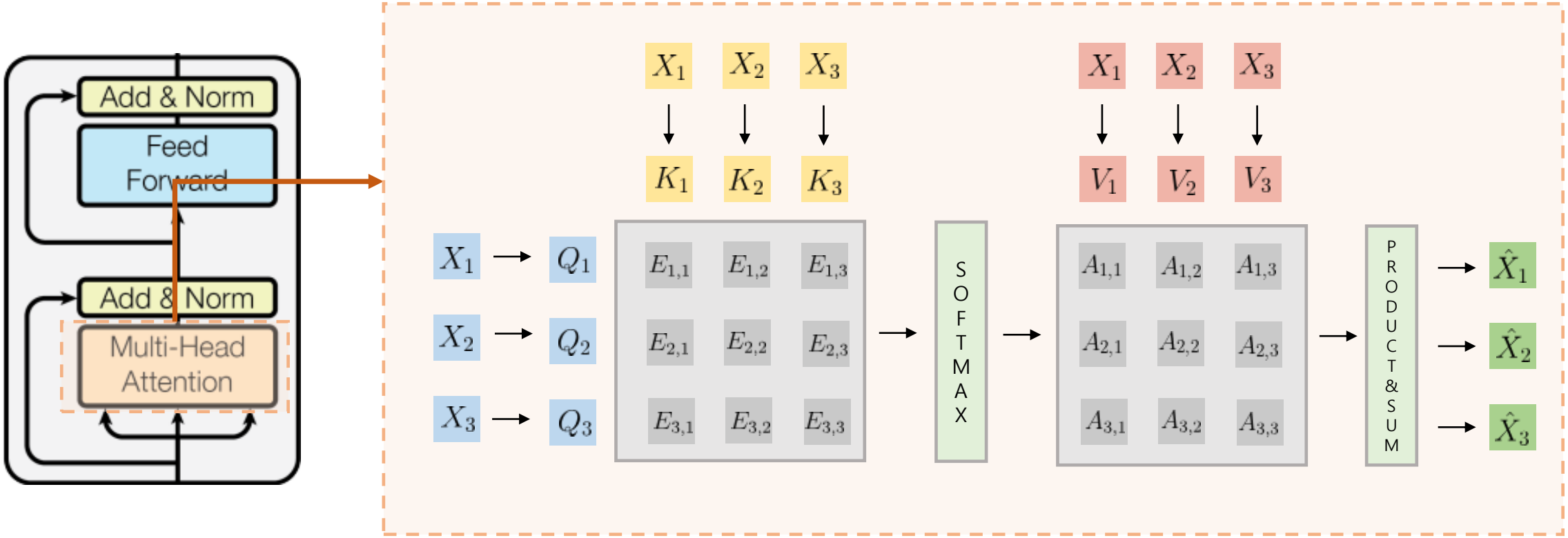
시점	t+1	t+2		t+h
실제값	1	3	...	9
예측값	9	15	...	50

Method

Are Transformers Effective for Time Series Forecasting?

❖ **Why Transformers is Ineffective in Time Series Forecasting?**

- Permutation Equivariant: 어떠한 인덱스에 대해 인풋의 순서를 바꾸게 되면 **아웃풋 또한 순서만 바뀐다**
- Transformer의 Self-Attention에서 발생



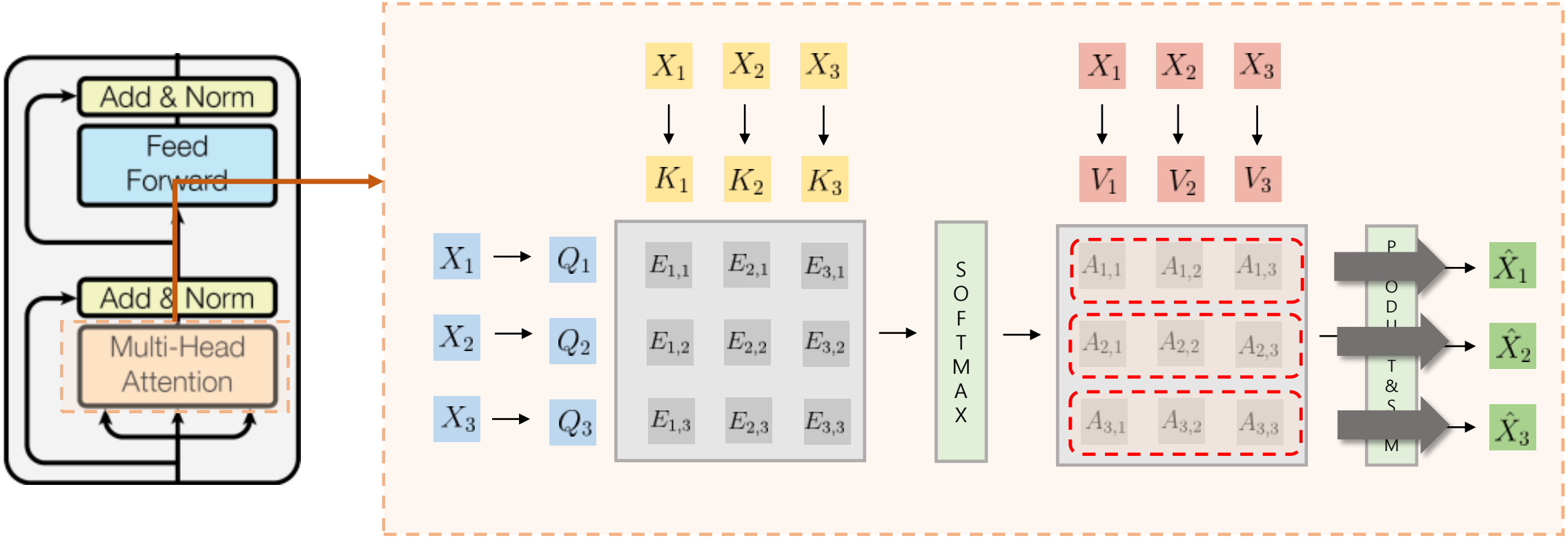
Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Method

Are Transformers Effective for Time Series Forecasting?

❖ **Why Transformers is Ineffective in Time Series Forecasting?**

- Permutation Equivariant: 어떠한 인덱스에 대해 인풋의 순서를 바꾸게 되면 **아웃풋 또한 순서만 바뀐다**
- Transformer의 Self-Attention에서 발생



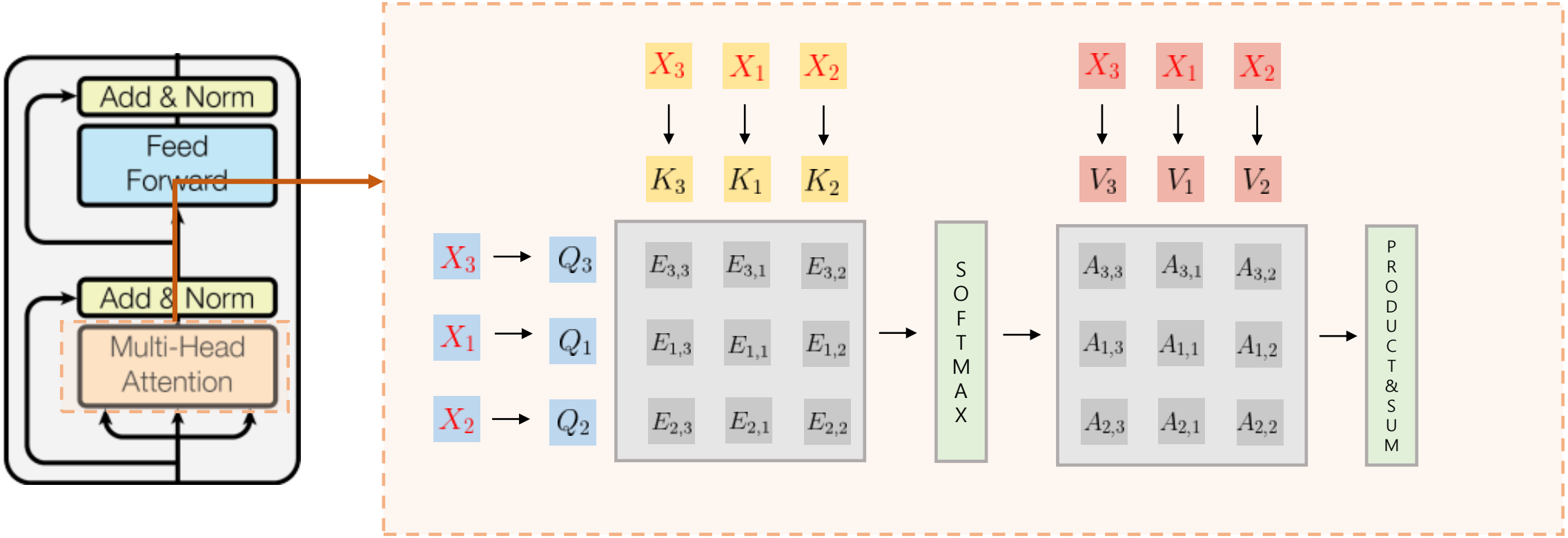
Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Method

Are Transformers Effective for Time Series Forecasting?

❖ **Why Transformers is Ineffective in Time Series Forecasting?**

- Permutation Equivariant: 어떠한 인덱스에 대해 인풋의 순서를 바꾸게 되면 **아웃풋 또한 순서만 바뀐다**
- Transformer의 Self-Attention에서 발생



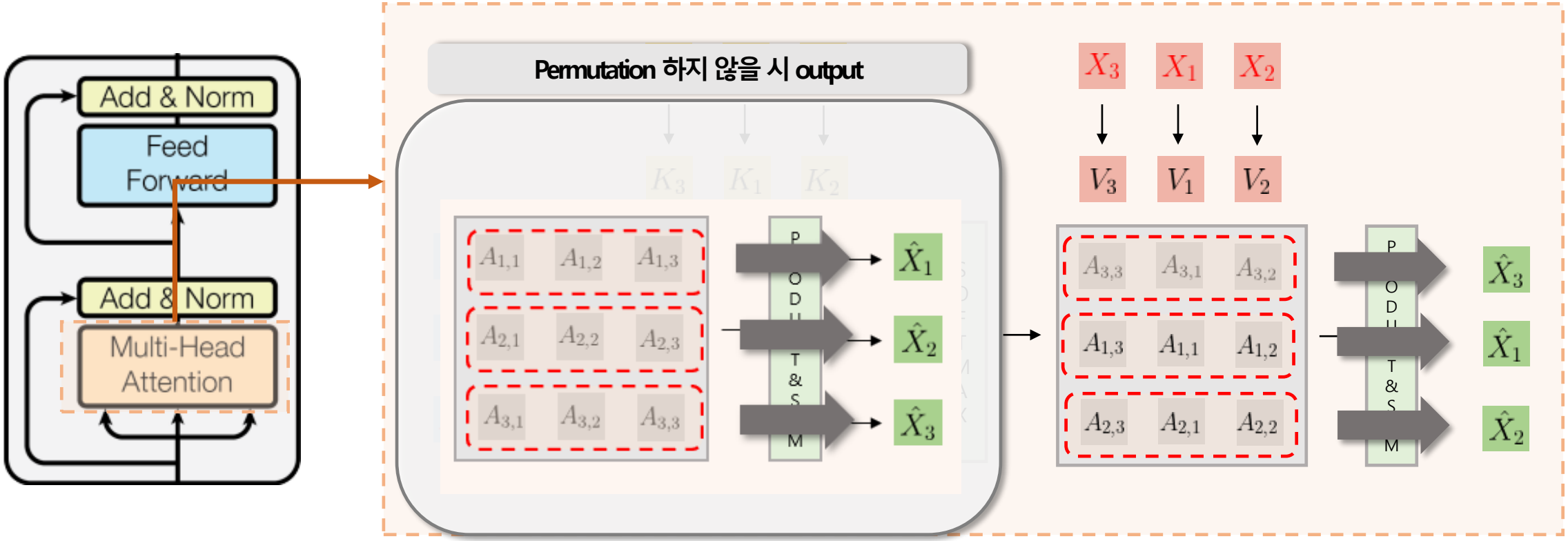
Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Method

Are Transformers Effective for Time Series Forecasting?

❖ **Why Transformers is Ineffective in Time Series Forecasting?**

- Permutation Equivariant: 어떠한 인덱스에 대해 인풋의 순서를 바꾸게 되면 **아웃풋 또한 순서만 바뀐다**
- Transformer의 Self-Attention에서 발생



Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Method

Are Transformers Effective for Time Series Forecasting?

❖ Why Transformers is Ineffective in Time Series Forecasting?

- Permutation Equivariant: 어떠한 인덱스에 대해 인풋의 순서를 바꾸게 되면 **아웃풋 또한 순서만 바뀐다**
- Transformer의 Self-Attention에서 발생

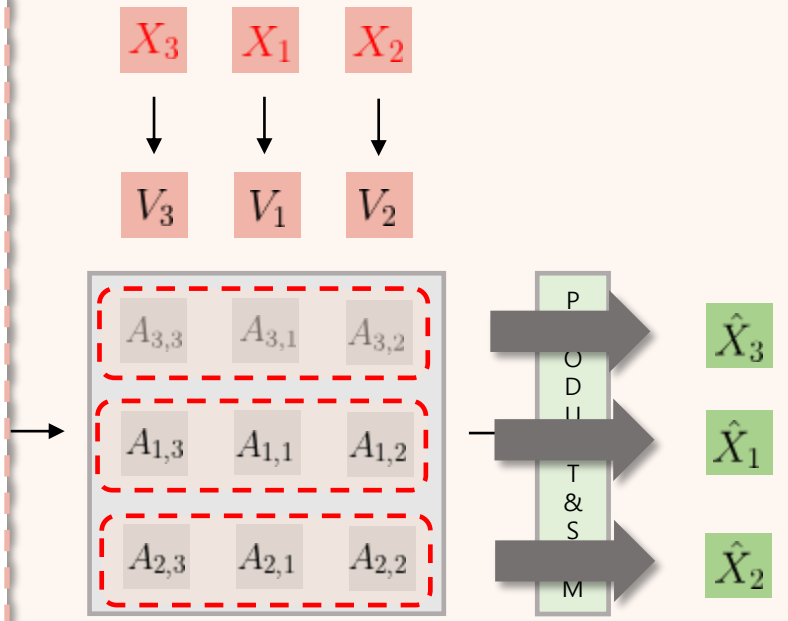
Permutation Equivariant

어떠한 인덱스에 대해 인풋의 순서를 바꾸게 되면
그 아웃풋 또한 순서만 바뀐 형태로 출력된다.

$$g_\phi(\mathcal{T}[x]) = \mathcal{T}(g_\phi(x))$$

Self-Attention 구조는 순서 정보를 반영하지 못한다

Transformer는 Positional Encoding으로 어느 정도 반영
하지만 Transformer의 핵심은 MHA이기 때문에 순서 정보에서 손실을 가져옴



Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Method

Are Transformers Effective for Time Series Forecasting?

❖ A Simple Baseline: LTSF-Linear

- 논문에서는 Transformer가 가지고 있는 문제를 해결할 수 있는 방법론으로 **하나의 Linear Layer를 제시**
- Linear Layer는 **Autoregressive하지 않고, Permutation Equivariant하지 않음**

Transformer

Error Accumulation

초기 시점에서 예측값을 잘못 예측했을 경우,
잘못된 값을 참조하기 때문에
뒤 예측값은 더 부정확해진다.

Permutation Equivariant

어떠한 인덱스에 대해 인풋의 순서를 바꾸게 되면
그 아웃풋 또한 순서만 바뀐 형태로 출력된다.

Simple Linear

Not Autoregressive

Matrix Multiplication 하나로
이전 timestep에서 예측하고자 하는 길이를 바로 산출할 수 있음

Permutation Inequivariant

Index가 바뀌면 그에 상응하는 Output 또한 바뀐다

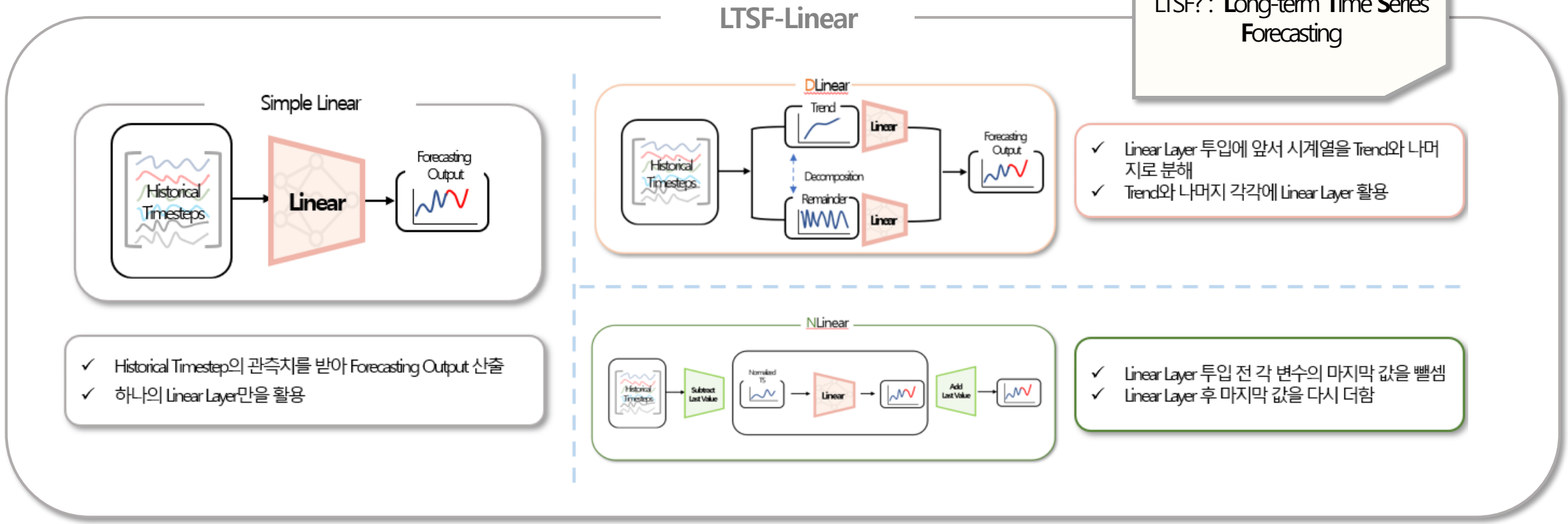
Method

Are Transformers Effective for Time Series Forecasting?

❖ A Simple Baseline: LTSF-Linear

- 본 논문에서 제시하고 있는 세 가지 방법론은 모두 Linear Layer를 토대로 함
  - ✓ 1) Simple Linear 2) D-Linear 3) N-Linear

LTSF? : **L**ong-term **T**ime **S**eries **F**orecasting



Method

Are Transformers Effective for Time Series Forecasting?

❖ Experiments

- 9개의 벤치마크 데이터셋 활용
- 입력: windowing process를 수행한 sub-series 형태 / 출력: 예측하고자 하는 길이 만큼의 다변량 forecasting output

Datasets	ETTh1&ETTh2	ETTm1 &ETTm2	Traffic	Electricity	Exchange-Rate	Weather	ILI
Variates	7	7	862	321	8	21	7
Timesteps	17,420	69,680	17,544	26,304	7,588	52,696	966
Granularity	1hour	5min	1hour	1hour	1day	10min	1week

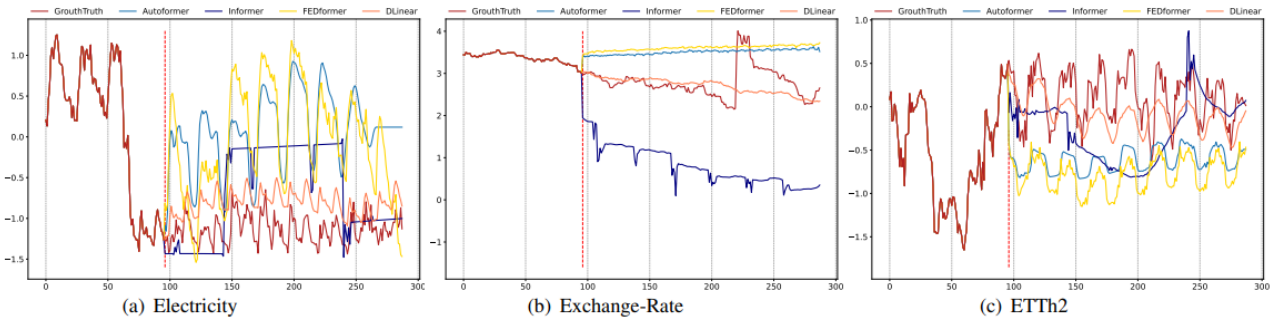
Method

Are Transformers Effective for Time Series Forecasting?

❖ Experiments

- LTSF-Linear가 Transformer 기반 모델들의 성능을 압도
- 9개의 벤치마크 데이터셋에서 LTSF Linear가 기존 대비 1~45%의 성능 개선을 보여줌

Methods		IMP.	Linear*		NLinear*		DLinear*		FEDformer		Autoformer		Informer	
Metric		MSE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	27.40%	**0.140**	**0.237**	0.141	**0.237**	**0.140**	**0.237**	0.193	0.308	0.201	0.317	0.274	0.368
	192	23.88%	**0.153**	0.250	0.154	**0.248**	**0.153**	0.249	0.201	0.315	0.222	0.334	0.296	0.386
	336	21.02%	**0.169**	0.268	0.171	**0.265**	**0.169**	0.267	0.214	0.329	0.231	0.338	0.300	0.394
	720	17.47%	**0.203**	0.301	0.210	**0.297**	**0.203**	0.301	0.246	0.355	0.254	0.361	0.373	0.439
Exchange	96	45.27%	0.082	0.207	0.089	0.208	**0.081**	0.203	0.148	0.278	0.197	0.323	0.847	0.752
	192	42.06%	0.167	0.304	0.180	0.300	**0.157**	0.293	0.271	0.380	0.300	0.369	1.204	0.895
	336	33.69%	0.328	0.432	0.331	0.415	**0.305**	0.414	0.460	0.500	0.509	0.524	1.672	1.036
	720	46.19%	0.964	0.750	1.033	0.780	**0.643**	**0.601**	1.195	0.841	1.447	0.941	2.478	1.310
Traffic	96	30.15%	**0.410**	0.282	**0.410**	**0.279**	**0.410**	0.282	0.587	0.366	0.613	0.388	0.719	0.391
	192	29.96%	**0.423**	0.287	**0.423**	**0.284**	**0.423**	0.287	0.604	0.373	0.616	0.382	0.696	0.379
	336	29.95%	0.436	0.295	**0.435**	**0.290**	0.436	0.296	0.621	0.383	0.622	0.337	0.777	0.420
	720	25.87%	0.466	0.315	**0.464**	**0.307**	0.466	0.315	0.626	0.382	0.660	0.408	0.864	0.472
Weather	96	18.89%	**0.176**	0.236	0.182	**0.232**	**0.176**	0.237	0.217	0.296	0.266	0.336	0.300	0.384
	192	21.01%	**0.218**	0.276	0.225	**0.269**	0.220	0.282	0.276	0.336	0.307	0.367	0.598	0.544
	336	22.71%	**0.262**	0.312	0.271	**0.301**	0.265	0.319	0.339	0.380	0.359	0.395	0.578	0.523
	720	19.85%	0.326	0.365	0.338	**0.348**	**0.323**	0.362	0.403	0.428	0.419	0.428	1.059	0.741
ILI	24	47.86%	1.947	0.985	**1.683**	**0.858**	2.215	1.081	3.228	1.260	3.483	1.287	5.764	1.677
	36	36.43%	2.182	1.036	**1.703**	**0.859**	1.963	0.963	2.679	1.080	3.103	1.148	4.755	1.467
	48	34.43%	2.256	1.060	**1.719**	**0.884**	2.130	1.024	2.622	1.078	2.669	1.085	4.763	1.469
	60	34.33%	2.390	1.104	**1.819**	**0.917**	2.368	1.096	2.857	1.157	2.770	1.125	5.264	1.564
ETTh1	96	0.80%	0.375	0.397	**0.374**	**0.394**	0.375	0.399	0.376	0.419	0.449	0.459	0.865	0.713
	192	3.57%	0.418	0.429	0.408	**0.415**	**0.405**	0.416	0.420	0.448	0.500	0.482	1.008	0.792
	336	6.54%	0.479	0.476	**0.429**	**0.427**	0.439	0.443	0.459	0.465	0.521	0.496	1.107	0.809
	720	13.04%	0.624	0.592	**0.440**	**0.453**	0.472	0.490	0.506	0.507	0.514	0.512	1.181	0.865
ETTh2	96	19.94%	0.288	0.352	**0.277**	**0.338**	0.289	0.353	0.346	0.388	0.358	0.397	3.755	1.525
	192	19.81%	0.377	0.413	**0.344**	**0.381**	0.383	0.418	0.429	0.439	0.456	0.452	5.602	1.931
	336	25.93%	0.452	0.461	**0.357**	**0.400**	0.448	0.465	0.496	0.487	0.482	0.486	4.721	1.835
	720	14.25%	0.698	0.595	**0.394**	**0.436**	0.605	0.551	0.463	0.474	0.515	0.511	3.647	1.625
ETTm1	96	21.10%	0.308	0.352	0.306	0.348	**0.299**	**0.343**	0.379	0.419	0.505	0.475	0.672	0.571
	192	21.36%	0.340	0.369	0.349	0.375	**0.335**	**0.365**	0.426	0.441	0.553	0.496	0.795	0.669
	336	17.07%	0.376	0.393	0.375	0.388	**0.369**	**0.386**	0.445	0.459	0.621	0.537	1.212	0.871
	720	21.73%	0.440	0.435	0.433	0.422	**0.425**	**0.421**	0.543	0.490	0.671	0.561	1.166	0.823
ETTm2	96	17.73%	0.168	0.262	**0.167**	**0.255**	**0.167**	0.260	0.203	0.287	0.255	0.339	0.365	0.453
	192	17.84%	0.232	0.308	**0.221**	**0.293**	0.224	0.303	0.269	0.328	0.281	0.340	0.533	0.563
	336	15.69%	0.320	0.373	**0.274**	**0.327**	0.281	0.342	0.325	0.366	0.339	0.372	1.363	0.887
	720	12.58%	0.413	0.435	**0.368**	**0.384**	0.397	0.421	0.421	0.415	0.433	0.432	3.379	1.338



(a) Electricity (b) Exchange-Rate (c) ETTh2

Background
Multivariate Time Series Forecasting

LTSF-Linear

문제 제기

Transformer는
시계열 예측 문제에서 최적의
모델이 아니다.

PatchTST

반박

Channel Independent Strategy와
결합하면
시계열 예측에서 Transformer는
최적의 모델이다

The Capacity and Robustness Trade-off

원인

왜 Channel independent Strategy
는
시계열 예측에서 뛰어난
성능을 발휘하는가?

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ **A Time Series is Worth 64 Words: Long-term Forecasting with Transformers(2023, ICLR)**

- 2024년 10월 기준 734회 인용
- 기존 시계열 예측 Transformer model의 한계를 극복할 수 있는 **Channel independent Strategy(CI Strategy)**와 Patching Strategy를 도입

A TIME SERIES IS WORTH 64 WORDS:
LONG-TERM FORECASTING WITH TRANSFORMERS

Yuqi Nie[1]*, **Nam H. Nguyen**[2]*, **Phanwadee Sinthong**[2], **Jayant Kalagnanam**[2]
[1]Princeton University [2]IBM Research
ynie@princeton.edu, nnguyen@us.ibm.com, Gift.Sinthong@ibm.com,
jayant@us.ibm.com

ABSTRACT

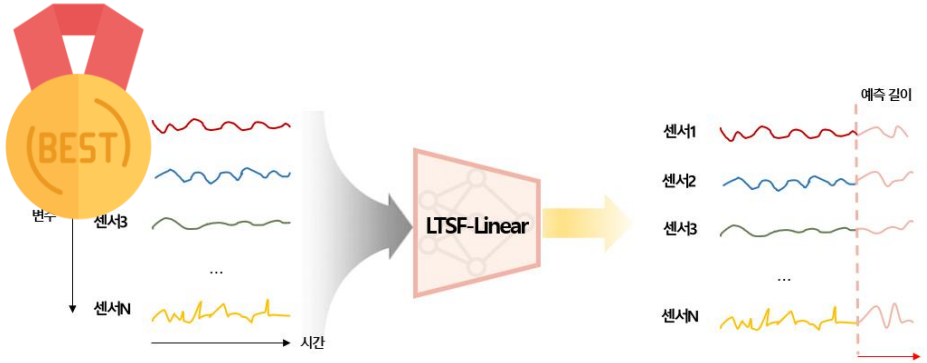
We propose an efficient design of Transformer-based models for multivariate time series forecasting and self-supervised representation learning. It is based on two key components: (i) segmentation of time series into subseries-level patches which are served as input tokens to Transformer; (ii) channel-independence where each channel contains a single univariate time series that shares the same embedding and Transformer weights across all the series. Patching design naturally has three-fold benefit: local semantic information is retained in the embedding; computation and memory usage of the attention maps are quadratically reduced given the same look-back window; and the model can attend longer history. Our channel-independent patch time series Transformer (PatchTST) can improve the long-term forecasting accuracy significantly when compared with that of SOTA Transformer-based models. We also apply our model to self-supervised pre-training tasks and attain excellent fine-tuning performance, which outperforms supervised training on large datasets. Transferring of masked pre-trained representation on one dataset to others also produces SOTA forecasting accuracy.

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Preview

- LTSF-Linear: Linear 기반 모델들이 Transformer 기반 모델 대비 우수한 성능을 가지고 있음을 실험적으로 증명
- Question: LTSF-Linear의 우수한 성능은 정말로 **Transformer의 구조적 한계 때문**인가?



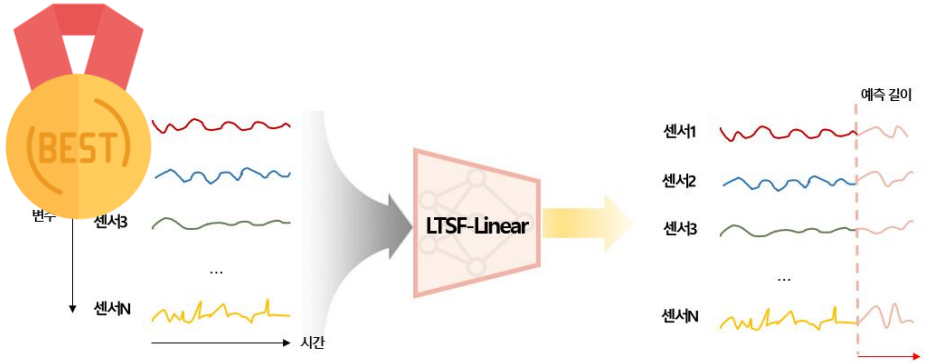
Time Series Forecasting에서 Linear가 Transformer보다 우수한 모델이다!

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Preview

- LTSF-Linear: Linear 기반 모델들이 Transformer 기반 모델 대비 우수한 성능을 가지고 있음을 실험적으로 증명
- Question: LTSF-Linear의 우수한 성능은 정말로 **Transformer의 구조적 한계 때문**인가?



Time Series Forecasting에서 Linear가 Transformer보다 우수한 모델이다!

LTSF-Linear의 **어떠한 요소가 성능을 높이는 주요한 원인**이다!

LTSF-Linear의 특정 Module + Transformer Backbone = **Optimal TS Forecasting Model?**

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ **Preview**

- LTSF-Linear: Linear 기반 모델들이 Transformer 기반 모델 대비 우수한 성능을 가지고 있음을 실험적으로 증명
- Question: LTSF-Linear의 우수한 성능은 정말로 **Transformer의 구조적 한계 때문**인가?

그렇다면 **LTSF-Linear의 어떤 요소**가 성능 향상을 불러오는가?

Time Series Forecasting에서 Linear가 Transformer보다 우수한 모델이다!

LTSF-Linear의 **어떠한 요소가 성능을 높이는 주요한 원인**이다!

LTSF-Linear의 특정 Module + Transformer Backbone = **Optimal TS Forecasting Model?**

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Channel independent Strategy

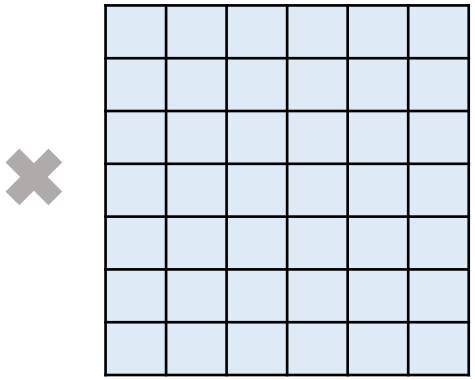
- LTSF-Linear는 **변수의 정보를 독립적으로(Channel independent)** 변수를 처리
- 기존 Transformer 모델은 Embedding space에서 **변수의 정보를 섞는 상태(Channel dependent)**로 임베딩을 생성

시점 L개를 받아 H개를 예측

Simple Linear

	시점 1	시점2	...	시점 L-1	시점 L
변수1	12	15	...	24	26
변수2	26	22	...	45	56
변수3	65	100	...	85	82
...	...	...	...	...	...
변수 K-1	36	45	...	48	55
변수 K	65	62	...	34	24

Input Time Series $\in \mathbb{R}^{K \times L}$



Weight Matrix $\in \mathbb{R}^{L \times H}$

	시점 L+1	...	시점 L+H-1	시점 L+H
변수1	24	...	27	32
변수2	55	...	45	53
변수3	79	...	80	85
...	...	...	...	...
변수 K-1	52	...	43	40
변수 K	18	...	57	60

Output Time Series $\in \mathbb{R}^{K \times H}$

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Channel independent Strategy

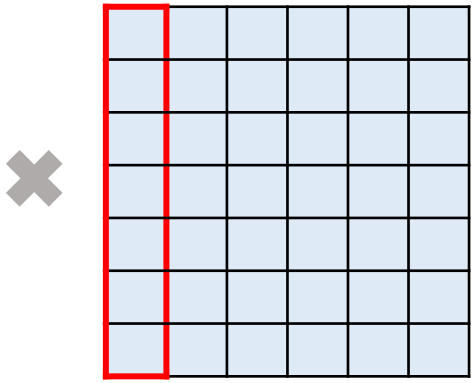
- LTSF-Linear는 **변수의 정보를 독립적으로(Channel independent)** 변수를 처리
- 기존 Transformer 모델은 Embedding space에서 **변수의 정보를 섞는 상태(Channel dependent)**로 임베딩을 생성

시점 L개를 받아 H개를 예측

Simple Linear

	시점 1	시점2	...	시점 L-1	시점 L
변수1	12	15	...	24	26
변수2	26	22	...	45	56
변수3	65	100	...	85	82
...	...	...	...	...	...
변수 K-1	36	45	...	48	55
변수 K	65	62	...	34	24

Input Time Series $\in \mathbb{R}^{K \times L}$



Weight Matrix $\in \mathbb{R}^{L \times H}$

	시점 L+1	...	시점 L+H-1	시점 L+H
변수1	24	...	27	32
변수2	55	...	45	53
변수3	79	...	80	85
...	...	...	...	...
변수 K-1	52	...	43	40
변수 K	18	...	57	60

Output Time Series $\in \mathbb{R}^{K \times H}$

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Channel independent Strategy

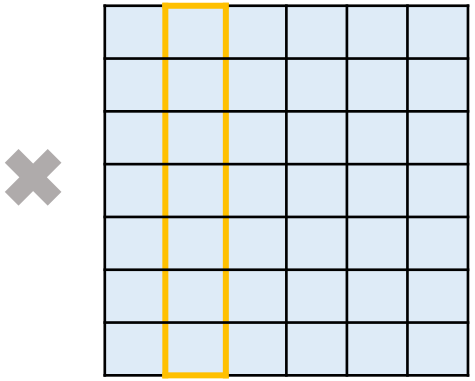
- LTSF-Linear는 **변수의 정보를 독립적으로(Channel independent)** 변수를 처리
- 기존 Transformer 모델은 Embedding space에서 **변수의 정보를 섞는 상태(Channel dependent)**로 임베딩을 생성

시점 L개를 받아 H개를 예측

Simple Linear

	시점 1	시점2	...	시점 L-1	시점 L
변수1	12	15	...	24	26
변수2	26	22	...	45	56
변수3	65	100	...	85	82
...	...	...	...	...	...
변수 K-1	36	45	...	48	55
변수 K	65	62	...	34	24

Input Time Series $\in \mathbb{R}^{K \times L}$



Weight Matrix $\in \mathbb{R}^{L \times H}$

	시점 L+1	...	시점 L+H-1	시점 L+H
변수1	24	...	27	32
변수2	55	...	45	53
변수3	79	...	80	85
...	...	...	...	...
변수 K-1	52	...	43	40
변수 K	18	...	57	60

Output Time Series $\in \mathbb{R}^{K \times H}$

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

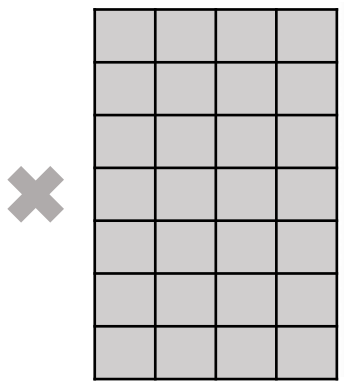
❖ Channel independent Strategy

- LTSF-Linear는 **변수의 정보를 독립적으로(Channel independent)** 변수를 처리
- 기존 Transformer 모델은 Embedding space에서 **변수의 정보를 섞는 상태(Channel dependent)**로 임베딩을 생성

Transformer

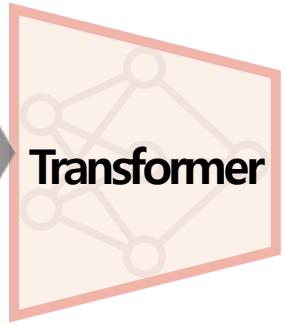
	변수 1	변수 2	...	변수 K-1	변수 K
시점 1	12	15	...	24	26
시점 2	26	22	...	45	56
...	...	...	...	...	...
시점 L-1	36	45	...	48	55
시점 L	65	62	...	34	24

Input Time Series $\in \mathbb{R}^{L \times K}$



Weight Matrix

$\in \mathbb{R}^{K \times d_{\text{model}}}$



	시점 L+1	...	시점 L+H-1	시점 L+H
변수1	24	...	27	32
변수2	55	...	45	53
변수3	79	...	80	85
...	...	...	...	...
변수 K-1	52	...	43	40
변수 K	18	...	57	60

Output Time Series $\in \mathbb{R}^{K \times H}$

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

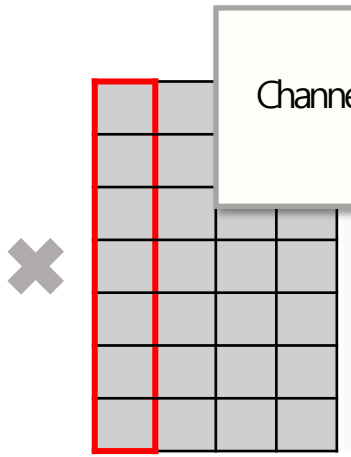
❖ Channel independent Strategy

- LTSF-Linear는 **변수의 정보를 독립적으로(Channel Independent)** 변수를 처리
- 기존 Transformer 모델은 Embedding space에서 **변수의 정보를 섞는 상태(Channel Dependent)**로 임베딩을 생성

Transformer

	변수 1	변수 2	…	변수 K-1	변수 K
시점 1	12	15	…	24	26
시점 2	26	22	…	45	56
…	…	…	…	…	…
시점 L-1	36	45	…	48	55
시점 L	65	62	…	34	24

Input Time Series $\in \mathbb{R}^{L \times K}$



Weight Matrix

$\in \mathbb{R}^{K \times d_{\text{model}}}$

Channel? = Multivariate Time Series의 **변수**

Embedding 생성 과정에서
변수들이 섞이는 형태로 임베딩 생성

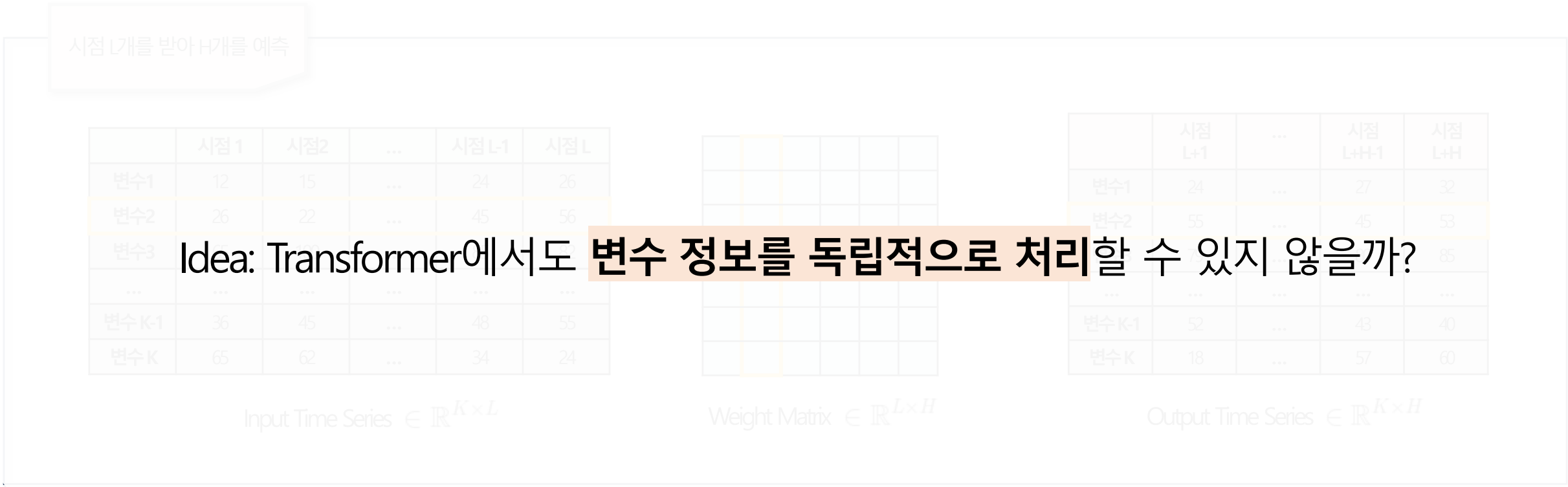
Channel **Dependent**!

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ **Channel Independent Strategy**

- LTSF-Linear는 **변수의 정보를 독립적으로(Channel Independent)** 변수를 처리
- 기존 Transformer 모델은 Embedding space에서 **변수의 정보를 섞는 상태(Channel dependent)**로 임베딩을 생성



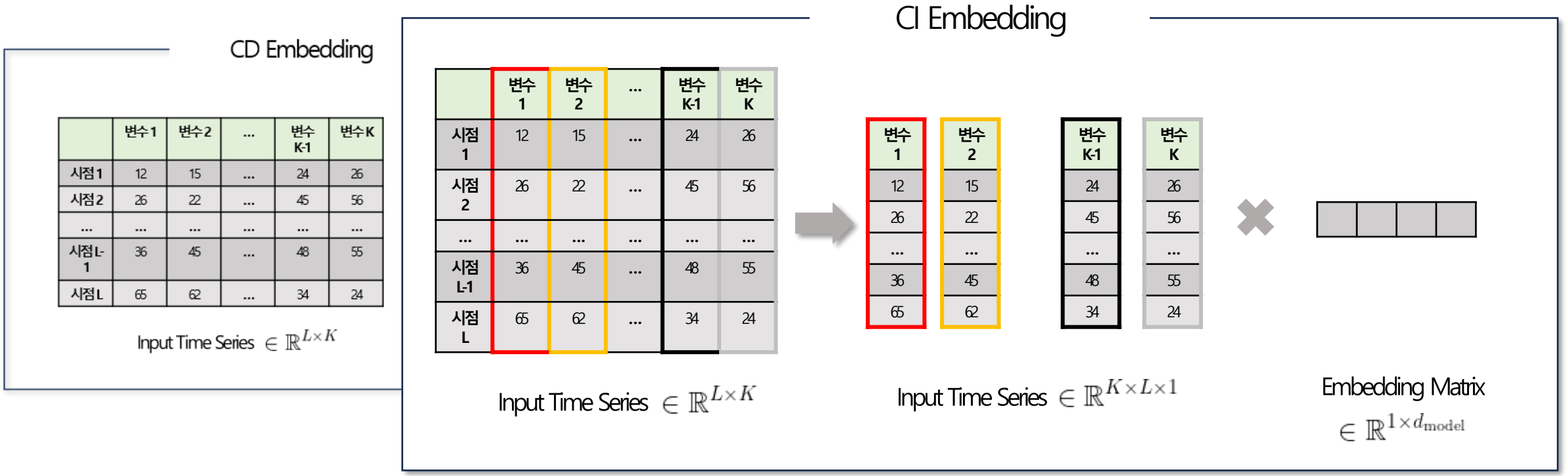
Idea: Transformer에서도 **변수 정보를 독립적으로 처리**할 수 있지 않을까?

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ **Channel Independent Strategy**

- Embedding 전 변수 별로 시퀀스 길이만큼 나눔
- Embedding Matrix는 L x 1의 sequence 길이를 임베딩 -> **Channel Independent!**



Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Patching Strategy

- PatchTST는 CI Strategy와 더불어 지역적인 정보를 추출하기 위해 **Patch 구조를 도입**
- Idea: 자연어와 달리 시계열의 하나의 timestep은 의미론적 정보를 가지지 않고, **주변 정보에 의존**한다.

언어

From the moment we are born, we are slowly heading toward the death.

born과 death는 단어 그 자체로 의미를 가지고 있기 때문에 인접하지 않아도 **유사성 추출 가능**

시계열

	시점 1	시점2	시점3	시점 4	시점 5	…	시점 20	시점 21	시점 22	시점 23	시점 24
변수	12	15	22	24	26		15	14	13	19	22

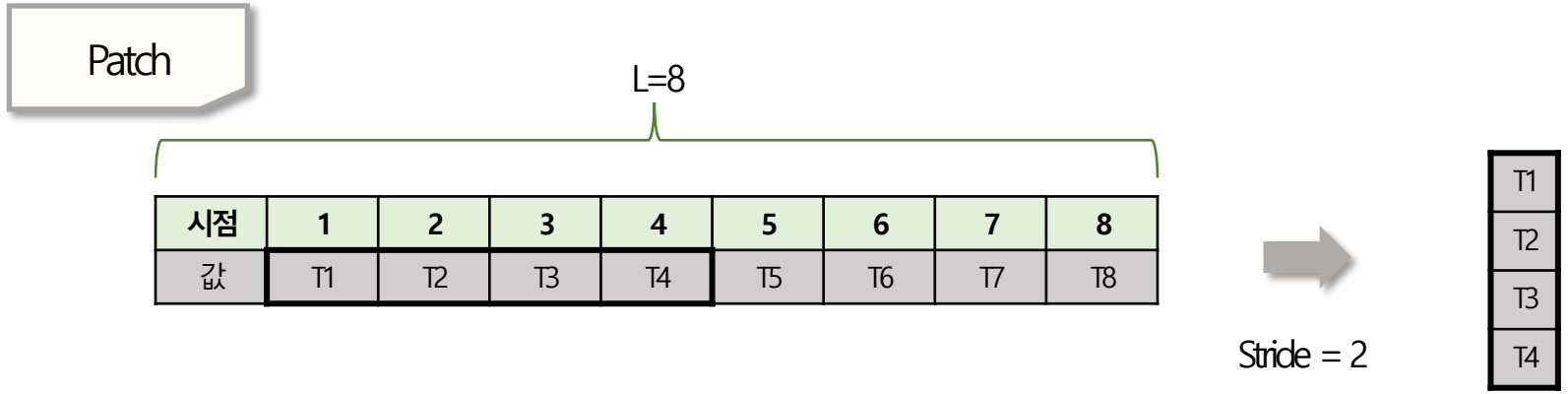
시계열의 각 시점의 정보는 **값 이외의 의미론적 정보**가 없기 때문에 주변 영역에 의존

Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Patching Strategy

- **How?:** 스트라이드를 이동하면서 window에 있는 정보를 그대로 가져옴
- 길이가 충분하지 않을 경우 마지막 값을 뒤에 복사해서 붙여놓음으로써 패치를 완성

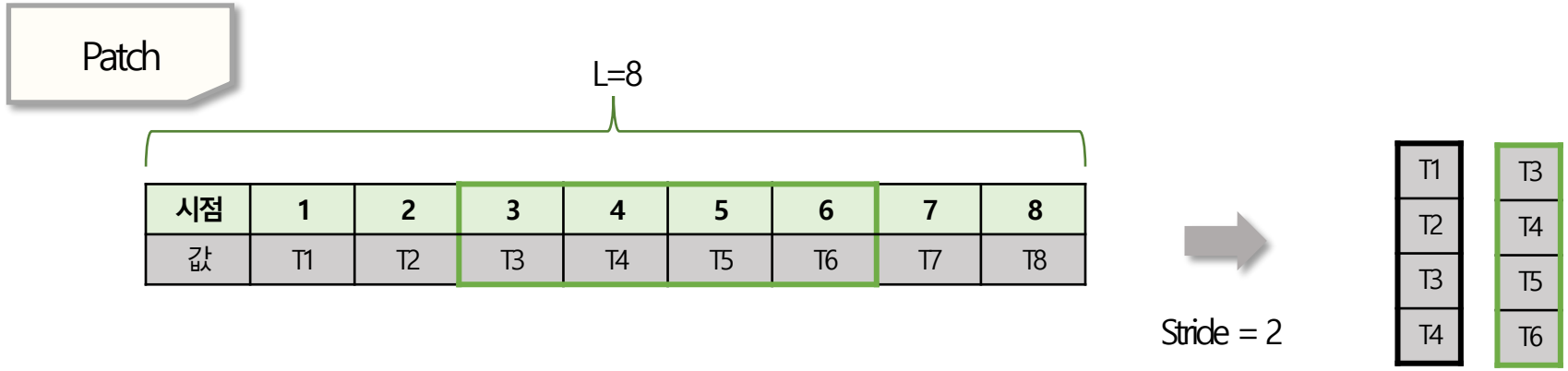


Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Patching Strategy

- **How?:** 스트라이드를 이동하면서 window에 있는 정보를 그대로 가져옴
- 길이가 충분하지 않을 경우 마지막 값을 뒤에 복사해서 붙여놓음으로써 패치를 완성

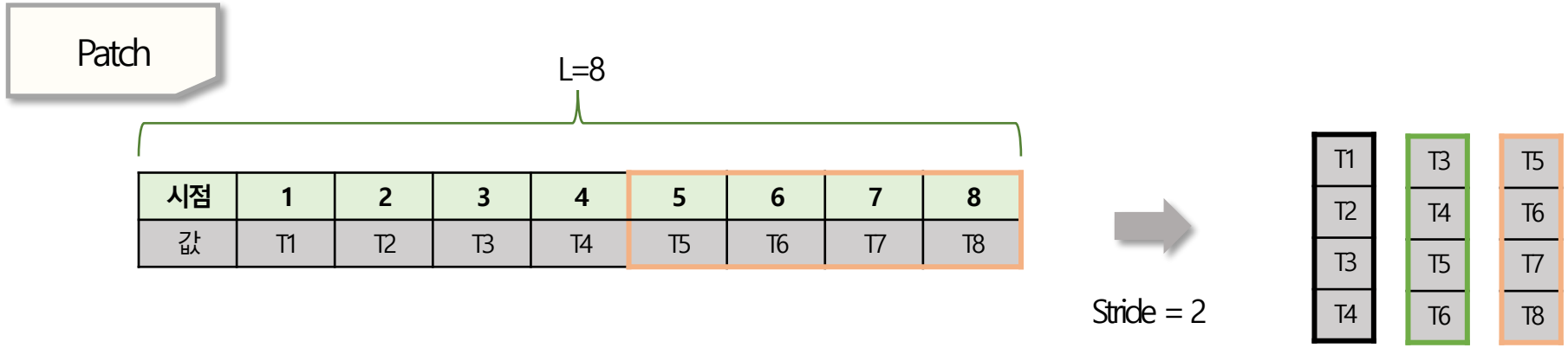


Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Patching Strategy

- **How?:** 스트라이드를 이동하면서 window에 있는 정보를 그대로 가져옴
- 길이가 충분하지 않을 경우 마지막 값을 뒤에 복사해서 붙여놓음으로써 패치를 완성

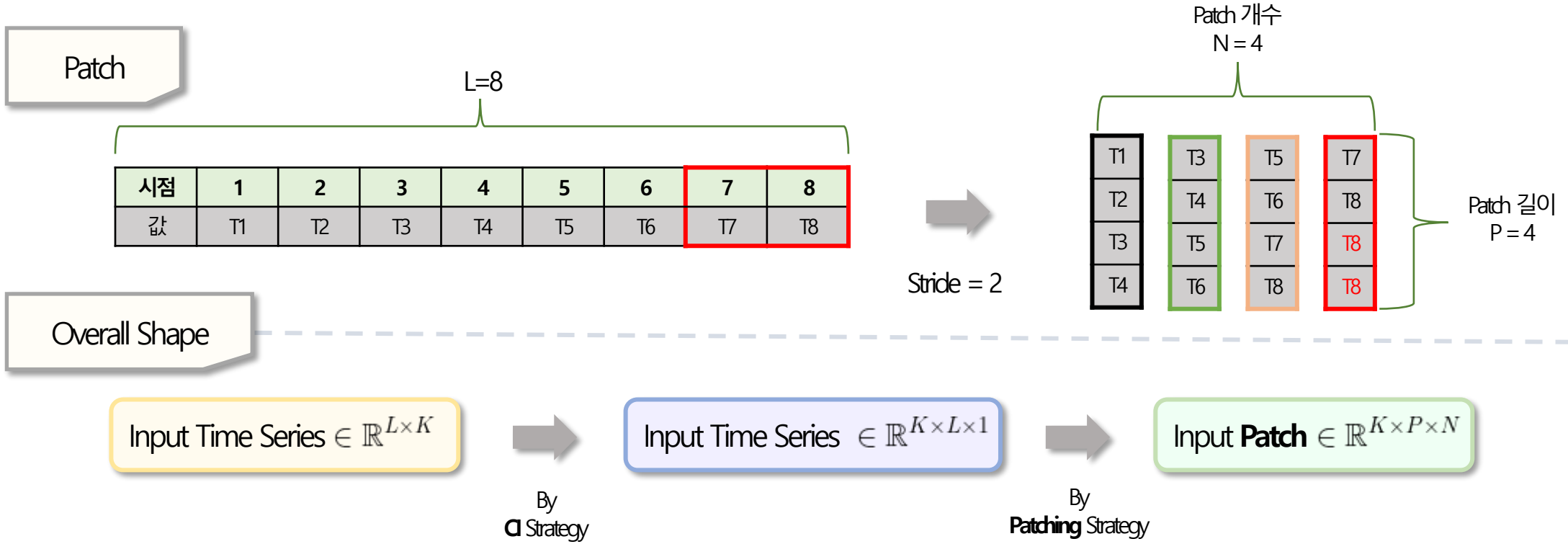


Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ Patching Strategy

- **How?:** 스트라이드를 이동하면서 window에 있는 정보를 그대로 가져옴
- 길이가 충분하지 않을 경우 마지막 값을 뒤에 복사해서 붙여놓음으로써 패치를 완성

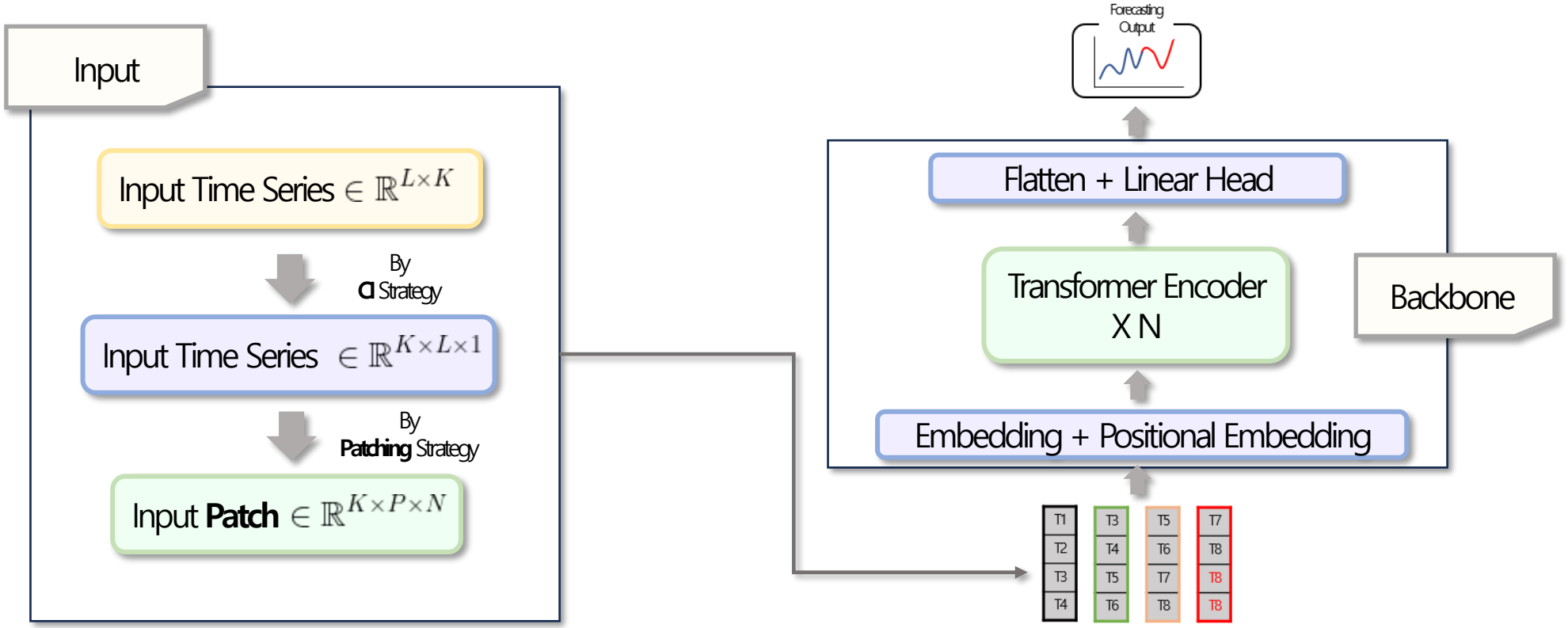


Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ PatchTST Architecture

- Backbone: Transformer **Encoder**
- Channel Independent와 Patch를 적용한 입력을 Backbone에 넣어서 변수 별 표현을 얻는 구조



Method

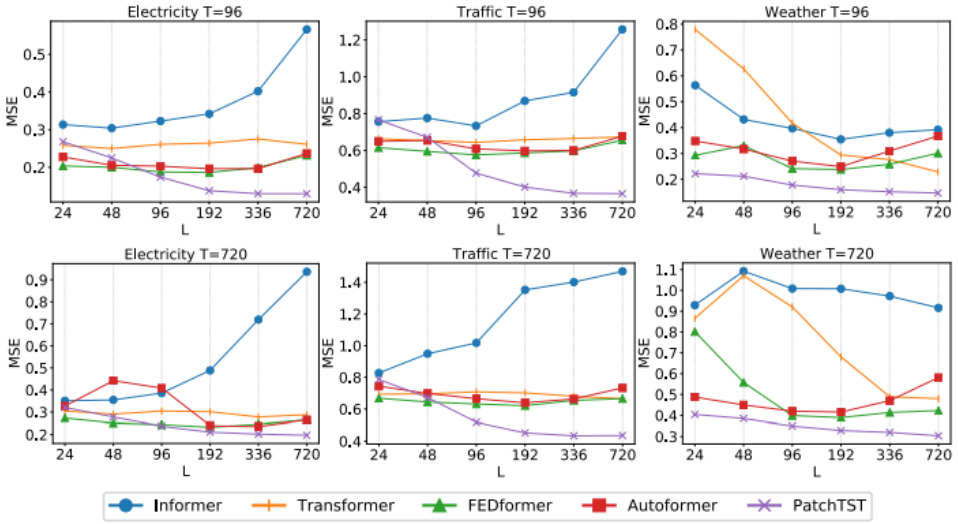
A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ **Experiments: 8개의 벤치마크 데이터셋 활용**

- Multivariate Time Series Forecasting에서 LTSF-Linear에 비해 전반적으로 좋은 성능 도출

Models		PatchTST/64		PatchTST/42		DLinear		FEDformer		Autoformer		Informer		Pyraformer		LogTrans	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.149**	**0.198**	0.152	0.199	0.176	0.237	0.238	0.314	0.249	0.329	0.354	0.405	0.896	0.556	0.458	0.490
	192	**0.194**	**0.241**	0.197	0.243	0.220	0.282	0.275	0.329	0.325	0.370	0.419	0.434	0.622	0.624	0.658	0.589
	336	**0.245**	**0.282**	0.249	0.283	0.265	0.319	0.339	0.377	0.351	0.391	0.583	0.543	0.739	0.753	0.797	0.652
	720	**0.314**	**0.334**	0.320	0.335	0.323	0.362	0.389	0.409	0.415	0.426	0.916	0.705	1.004	0.934	0.869	0.675
Traffic	96	**0.360**	**0.249**	0.367	0.251	0.410	0.282	0.576	0.359	0.597	0.371	0.733	0.410	2.085	0.468	0.684	0.384
	192	**0.379**	**0.256**	0.385	0.259	0.423	0.287	0.610	0.380	0.607	0.382	0.777	0.435	0.867	0.467	0.685	0.390
	336	**0.392**	**0.264**	0.398	0.265	0.436	0.296	0.608	0.375	0.623	0.387	0.776	0.434	0.869	0.469	0.734	0.408
	720	**0.432**	**0.286**	0.434	0.287	0.466	0.315	0.621	0.375	0.639	0.395	0.827	0.466	0.881	0.473	0.717	0.396
Electricity	96	**0.129**	**0.222**	0.130	**0.222**	0.140	0.237	0.186	0.302	0.196	0.313	0.304	0.393	0.386	0.449	0.258	0.357
	192	**0.147**	**0.240**	0.148	**0.240**	0.153	0.249	0.197	0.311	0.211	0.324	0.327	0.417	0.386	0.443	0.266	0.368
	336	**0.163**	**0.259**	0.167	0.261	0.169	0.267	0.213	0.328	0.214	0.327	0.333	0.422	0.378	0.443	0.280	0.380
	720	**0.197**	**0.290**	0.202	0.291	0.203	0.301	0.233	0.344	0.236	0.342	0.351	0.427	0.376	0.445	0.283	0.376
ILI	24	**1.319**	**0.754**	1.522	0.814	2.215	1.081	2.624	1.095	2.906	1.182	4.657	1.449	1.420	2.012	4.480	1.444
	36	1.579	0.870	**1.430**	**0.834**	1.963	0.963	2.516	1.021	2.585	1.038	4.650	1.463	7.394	2.031	4.799	1.467
	48	**1.553**	**0.815**	1.673	0.854	2.130	1.024	2.505	1.041	3.024	1.145	5.004	1.542	7.551	2.057	4.800	1.468
	60	**1.470**	**0.788**	1.529	0.862	2.368	1.096	2.742	1.122	2.761	1.114	5.071	1.543	7.662	2.100	5.278	1.560
ETTh1	96	**0.370**	0.400	0.375	**0.399**	0.375	**0.399**	0.376	0.415	0.435	0.446	0.941	0.769	0.664	0.612	0.878	0.740
	192	0.413	0.429	0.414	0.421	**0.405**	**0.416**	0.423	0.446	0.456	0.457	1.007	0.786	0.790	0.681	1.037	0.824
	336	**0.422**	0.440	0.431	**0.436**	0.439	0.443	0.444	0.462	0.486	0.487	1.038	0.784	0.891	0.738	1.238	0.932
	720	**0.447**	0.468	0.449	**0.466**	0.472	0.490	0.469	0.492	0.515	0.517	1.144	0.857	0.963	0.782	1.135	0.852
ETTh2	96	**0.274**	0.337	**0.274**	**0.336**	0.289	0.353	0.332	0.374	0.332	0.368	1.549	0.952	0.645	0.597	2.116	1.197
	192	0.341	0.382	**0.339**	**0.379**	0.383	0.418	0.407	0.446	0.426	0.434	3.792	1.542	0.788	0.683	4.315	1.635
	336	**0.329**	0.384	0.331	**0.380**	0.448	0.465	0.400	0.447	0.477	0.479	4.215	1.642	0.907	0.747	1.124	1.604
	720	**0.379**	**0.422**	**0.379**	**0.422**	0.605	0.551	0.412	0.469	0.453	0.490	3.656	1.619	0.963	0.783	3.188	1.540
ETTm1	96	0.293	0.346	**0.290**	**0.342**	0.299	0.343	0.326	0.390	0.510	0.492	0.626	0.560	0.543	0.510	0.600	0.546
	192	0.333	0.370	**0.332**	0.369	0.335	**0.365**	0.365	0.415	0.514	0.495	0.725	0.619	0.557	0.537	0.837	0.700
	336	0.369	0.392	**0.366**	0.392	0.369	**0.386**	0.392	0.425	0.510	0.492	1.005	0.741	0.754	0.655	1.124	0.832
	720	**0.416**	**0.420**	0.420	0.424	0.425	0.421	0.446	0.458	0.527	0.493	1.133	0.845	0.908	0.724	1.153	0.820
ETTm2	96	0.166	0.256	**0.165**	**0.255**	0.167	0.260	0.180	0.271	0.205	0.293	0.355	0.462	0.435	0.507	0.768	0.642
	192	0.223	0.296	**0.220**	**0.292**	0.224	0.303	0.252	0.318	0.278	0.336	0.595	0.586	0.730	0.673	0.989	0.757
	336	**0.274**	**0.329**	0.278	**0.329**	0.281	0.342	0.324	0.364	0.343	0.379	1.270	0.871	1.201	0.845	1.334	0.872
	720	**0.362**	**0.385**	0.367	**0.385**	0.397	0.421	0.410	0.420	0.414	0.419	3.001	1.267	3.625	1.451	3.048	1.328

Table 3: Multivariate long-term forecasting results with supervised PatchTST. We use prediction lengths $T \in \{24, 36, 48, 60\}$ for ILI dataset and $T \in \{96, 192, 336, 720\}$ for the others. The best results are in **bold** and the second best are underlined.



Method

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

❖ **Experiments**

- CI와 Patch 각각에 대해 ablation study 시행
- **CI Strategy를 적용했을 때 성능 향상이 두드러지는 모습**

Models		PatchTST								FEDformer	
		P+CI		CI		P		Original			
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.152**	**0.199**	0.164	0.213	0.168	0.223	0.177	0.236	0.238	0.314
	192	**0.197**	**0.243**	0.205	0.250	0.213	0.262	0.221	0.270	0.275	0.329
	336	**0.249**	**0.283**	0.255	0.289	0.266	0.300	0.271	0.306	0.339	0.377
	720	**0.320**	**0.335**	0.327	0.343	0.351	0.359	0.340	0.353	0.389	0.409
Traffic	96	**0.367**	**0.251**	0.397	0.271	0.595	0.376	-	-	0.576	0.359
	192	**0.385**	**0.259**	0.411	0.276	0.612	0.387	-	-	0.610	0.380
	336	**0.398**	**0.265**	0.423	0.282	0.651	0.391	-	-	0.608	0.375
	720	**0.434**	**0.287**	0.457	0.309	-	-	-	-	0.621	0.375
Electricity	96	**0.130**	**0.222**	0.136	0.231	0.196	0.307	0.205	0.318	0.186	0.302
	192	**0.148**	**0.240**	0.164	0.263	0.215	0.323	-	-	0.197	0.311
	336	**0.167**	**0.261**	0.168	0.262	0.228	0.338	-	-	0.213	0.328
	720	**0.202**	**0.291**	0.219	0.312	0.244	0.345	-	-	0.233	0.344

Background
Multivariate Time Series Forecasting

LTSF-Linear

문제 제기

Transformer는
시계열 예측 문제에서 최적의
모델이 아니다.

PatchTST

반박

Channel Indepenence Strategy와
결합하면
시계열 예측에서 Transformer는
최적의 모델이다

The Capacity and Robustness Trade-off

원인

왜 Channel Independent Strategy는
시계열 예측에서 뛰어난
성능을 발휘하는가?

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ **The Capacity and Robustness Trade-off: Revisiting the Channel Indepnent Strategy for Multivariate Time Series Forecasting(TKDE, 2024)**

- 2024년 10월 기준 59회 인용
- 다변량 시계열 예측에서 **왜 Channel Indepent Strategy가 뛰어난 성능**을 발휘하는지 이론적 근거를 제시

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

Lu Han, Han-Jia Ye, De-Chuan Zhan
State Key Laboratory for Novel Software Technology, Nanjing University
{hanlu,yehj}@lamda.nju.edu.cn,zhandc@nju.edu.cn

ABSTRACT

Multivariate time series data comprises various channels of variables. The multivariate forecasting models need to capture the relationship between the channels to accurately predict future values. However, recently, there has been an emergence of methods that employ the Channel Independent (CI) strategy. These methods view multivariate time series data as separate univariate time series and disregard the correlation between channels. Surprisingly, our empirical results have shown that models trained with the CI strategy outperform those trained with the Channel Dependent (CD) strategy, usually by a significant margin. Nevertheless, the reasons behind this phenomenon have not yet been thoroughly explored in the literature. This paper provides comprehensive empirical and theoretical analyses of the characteristics of multivariate time series datasets and the CI/CD strategy. Our results conclude that the CD approach has higher capacity but often lacks robustness to accurately predict distributionally drifted time series. In contrast, the CI approach trades capacity for robust prediction. Practical measures inspired by these analyses are proposed to address the capacity and robustness dilemma, including a modified CD method called Predict Residuals with Regularization (PRReg) that can surpass the CI strategy. We hope our findings can raise awareness among researchers about the characteristics of multivariate time series and inspire the construction of better forecasting models.

explicitly modeled by performing forecasting tasks [6, 7, 22, 39, 40]. Two widely used methods for time series forecasting are recurrent neural networks (RNNs) and convolutional neural networks (CNNs). RNNs model successive time points based on the Markov assumption [5, 16, 32], while CNNs extract variation information along the temporal dimension using techniques such as temporal convolutional networks (TCNs) [2, 12]. However, due to the Markov assumption in RNN and the local reception property in TCN, both of the two models are unable to capture the long-term dependencies in sequential data. Recently, Transformers with attention mechanisms have gained increasing popularity in other fields like natural language processing [8], speech recognition [9], and even computer vision [10]. Researchers have also explored the potential of Transformer models in long-term multivariate time series forecasting (MTSF) tasks [24, 38, 43, 44].

Despite the significant progress made by Transformer-based methods in forecasting long-term future values, a recent paper questions the effectiveness of Transformer [42]. The authors have demonstrated that a simple linear model can outperform all state-of-the-art Transformer-based methods. However, it is important to note that the linear model used by the authors employs a channel-independent training strategy that is different from previous works. Instead of considering all the channels as a whole, the authors train a univariate forecast model that is shared across all the channels. This training strategy is closely related to the global [33] or cross-

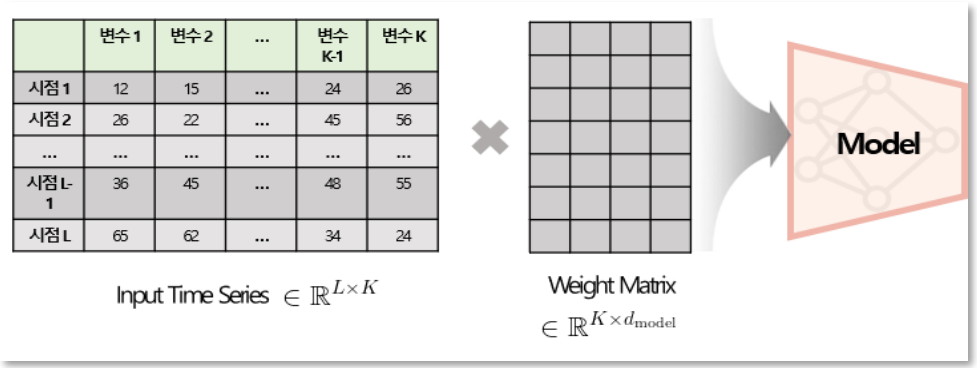
Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ Background

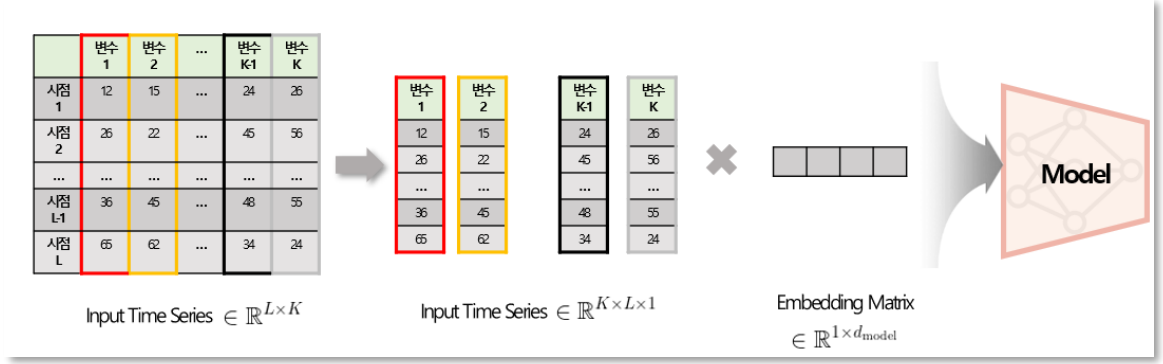
- CI Strategy의 효용성에 대해서는 PatchTST 등 **실험적인 근거 존재**
- 하지만 CI Strategy의 이론적인 background의 부재

CD Strategy



변수 간 **Correlation 고려 가능**

CI Strategy



변수 간 **Correlation 고려 불가**

Q. 그런데 왜 CI가 더 잘 될까?

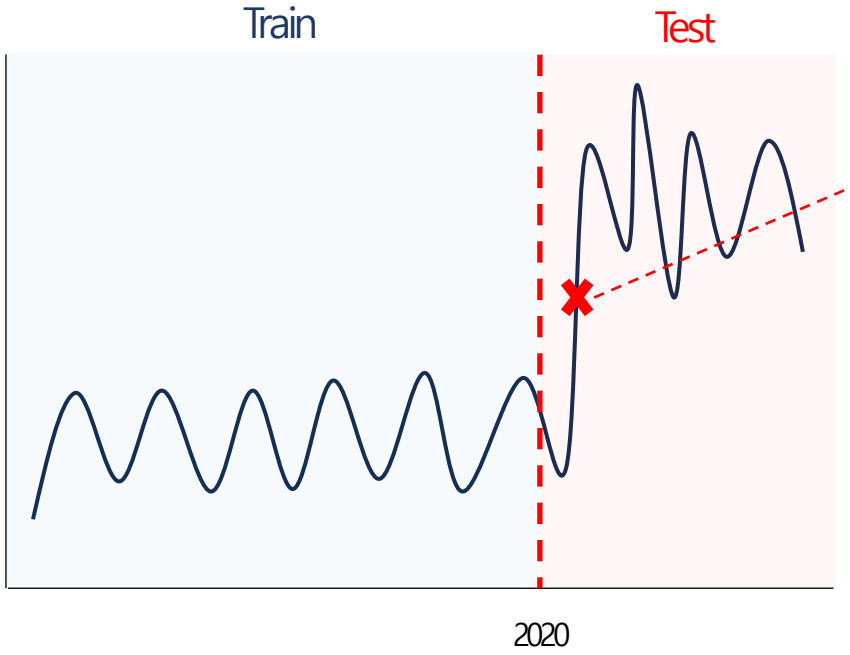
Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ **Why does CI overwhelms CDs? – Distribution Drift**

- 저자들은 실제 데이터셋에서 일어나는 분포 이동(**Distribution Drift**)가 **성능에 영향**을 미친다고 주장
- 시계열 데이터에서 일어날 수 있는 Distribution Drift는 **훈련 데이터와 테스트 데이터** 간의 분포 차이

<Ex. 천연가스 가격>



러시아-우크라이나 전쟁 발발

➡ 천연가스 주 수출국 러시아에 대한 무역 제재

➡ Test로 사용하는 2020년 이후의 데이터의 **분포 이동**

훈련 데이터와 테스트 데이터 간의 분포 차이는 존재한다!

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

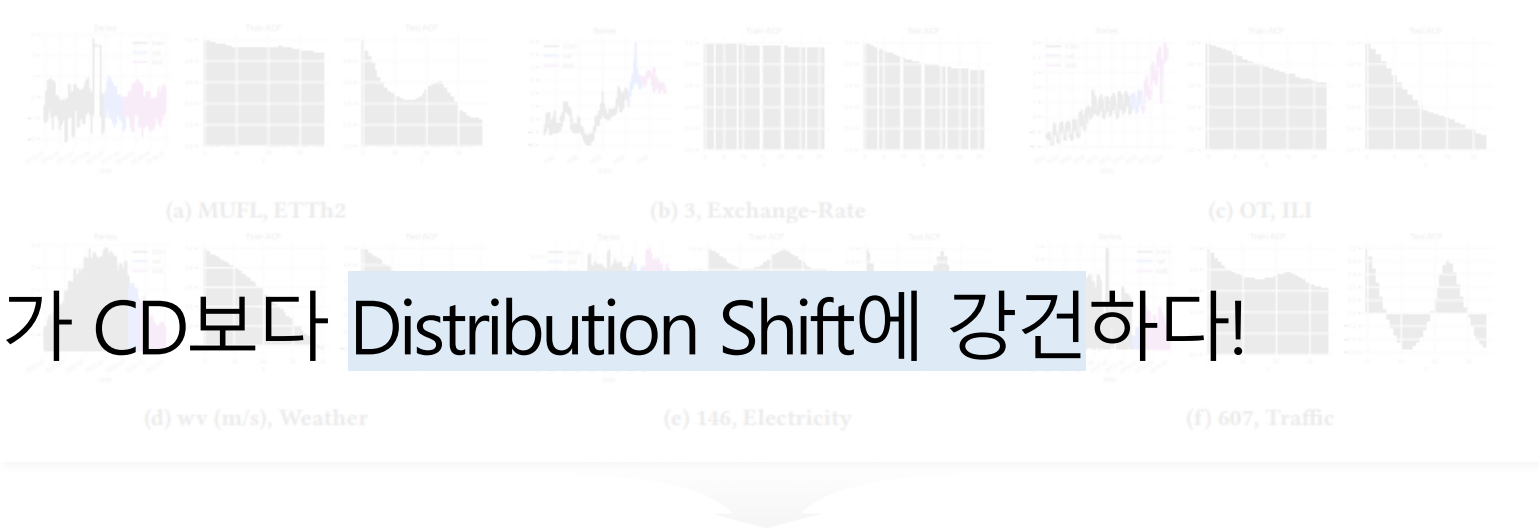
❖ **Why does CI overwhelms CDs? – Distribution Drift**

- 벤치마크 데이터셋에서 Distribution Drift가 일어나는지 관찰
- 이를 위해 Train set과 Test set의 ACF 패턴 차이가 있는지 조사

ACF(AutoCorrelation Function)

시간 차이에 대한 두 변수들의 상관계수

$$\rho(t_1, t_2) = \frac{\gamma(t_1, t_2)}{\sqrt{\sigma^2(t_1)\sigma^2(t_2)}}$$



Main Idea: CI가 CD보다 Distribution Shift에 강건하다!

ACF 패턴의 차이?: **같은 시간 차이에 대해서 다르게 영향을 받는다**
Train set과 Test set 사이의 **Distribution Shift**가 존재한다.

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

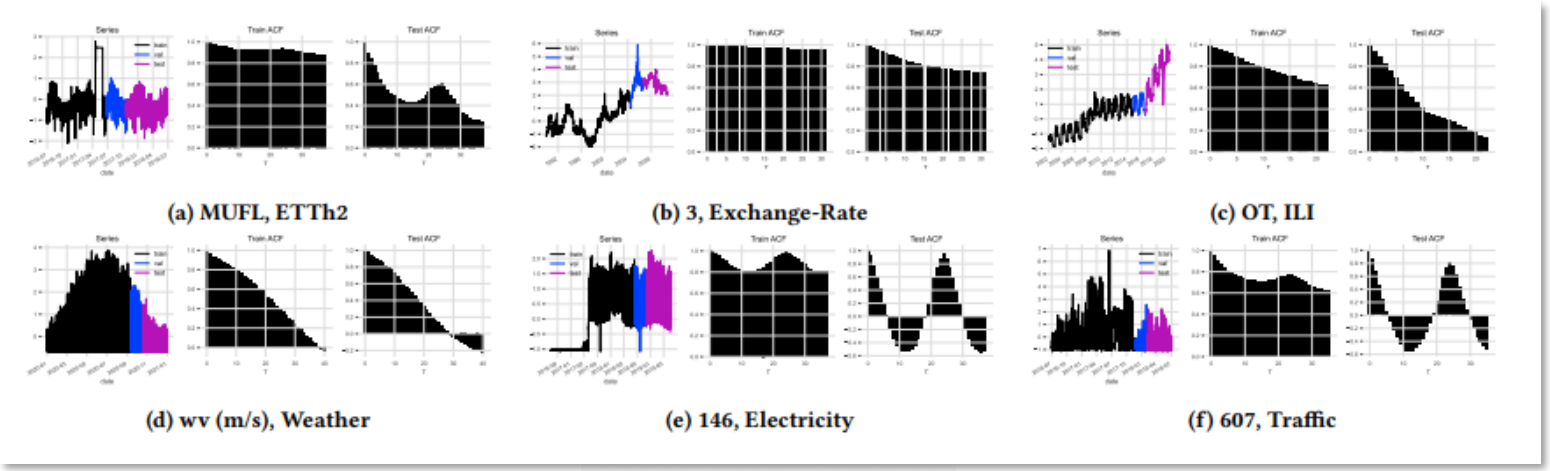
❖ **Why does CI overwhelms CDs? – Distribution Drift**

- 벤치마크 데이터셋에서 Distribution Drift가 일어나는지 관찰
- 이를 위해 Train set과 Test set의 ACF 패턴 차이가 있는지 조사

ACF(AutoCorrelation Function)

시간 차이에 대한 관측치들의 상관계수

$$\rho(t_1, t_2) = \frac{\gamma(t_1, t_2)}{\sqrt{\sigma^2(t_1)\sigma^2(t_2)}}$$



ACF 패턴의 차이?: **같은 시간 차이에 대해서 다르게 영향을 받는다**
Train set과 Test set 사이의 **Distribution Shift가 존재**한다.

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ CI Alleviates Distribution Drift

- 저자들은 CI Strategy가 시계열에서 일어나는 Distribution Drift를 **완화**시킬 수 있다고 주장

Notation

- ✓ Input time series $\mathbf{X} \in \mathbb{R}^{N \times C \times L}$
- ✓ Output time series $\mathbf{Y} \in \mathbb{R}^{N \times C \times H}$
  - N은 시계열의 개수, C는 변수 개수, L은 시퀀스 길이
  - L개의 sequence를 받아 H개의 sequence를 예측

Step 1 각 변수의 univariate time series를 가정

$\mathbf{A}^{(c)} \in \mathbb{R}^{N \times L}$ ✓ 채널 c에서의 Input time series

$\mathbf{B}^{(c)} \in \mathbb{R}^{N \times H}$ ✓ 채널 c에서의 Output time series

Step 2 Loss function

$$\mathcal{L} = \|\mathbf{AW} - \mathbf{B}\|_F^2$$

- ✓ $\mathbf{A}$는 $\mathbf{A}^{(c)}$을 Concat한 것
- ✓ $\mathbf{B}$는 $\mathbf{B}^{(c)}$을 Concat한 것

Step 3 OLS Estimator

$$\mathbf{A}^{\mathrm{T}}\mathbf{AW} = \mathbf{A}^{\mathrm{T}}\mathbf{B}$$

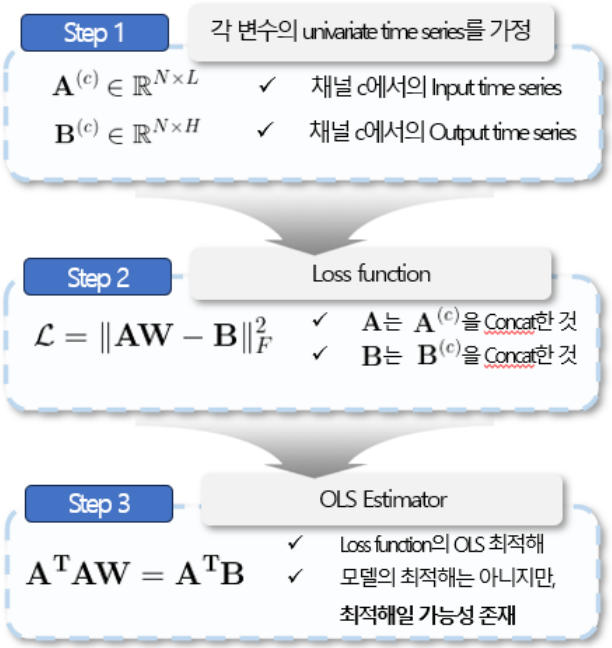
- ✓ Loss function의 OLS 최적해
- ✓ 모델의 최적해는 아니지만, **최적해일 가능성 존재**

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ CI Alleviates Distribution Drift

- 저자들은 CI Strategy가 시계열에서 일어나는 Distribution Drift를 **완화**시킬 수 있다고 주장



Channel dependent

어떻게 Concat?

$$\mathbf{A}_{cd} = \left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(C)}\right],\ \mathbf{A}_{cd} \in \mathbb{R}^{N\times(C*L)}$$
$$\mathbf{B}_{cd} = \left[\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \cdots, \mathbf{B}^{(C)}\right],\ \mathbf{B}_{cd} \in \mathbb{R}^{N\times(C*H)}$$

각 univariate time series를 **vertical concat**

Objective Function

$$\mathcal{L}_{cd} = \|\mathbf{A}_{cd}\mathbf{W} - \mathbf{B}_{cd}\|_F^2$$

OLS estimates

$$(\mathbf{A}_{cd}^T\mathbf{A}_{cd})\mathbf{W}_{cd} = \mathbf{A}_{cd}^T\mathbf{B}_{cd}$$

앞에서 정의된 Loss function와 OLS estimates

OLS coefficients

$$\begin{bmatrix} R'_{1,1} & R'_{1,2} & \cdots & R'_{1,C} \\ R'_{2,1} & R'_{2,2} & \cdots & R'_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ R'_{C,1} & R'_{C,2} & \cdots & R'_{C,C} \end{bmatrix} = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,C} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ R_{C,1} & R_{C,2} & \cdots & R_{C,C} \end{bmatrix} \boldsymbol{W}^*_{cd}$$

OLS 식을 풀면 **모든 변수 각각의 ACF에 대해** Optimal coefficient가 정해진다

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ CI Alleviates Distribution Drift

- 저자들은 CI Strategy가 시계열에서 일어나는 Distribution Drift를 **완화**시킬 수 있다고 주장

$$R'_{c_1,c_2} = \begin{bmatrix} \rho(1) & \rho(2) & \cdots & \rho(H) \\ \rho(2) & \rho(3) & \cdots & \rho(H+1) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(L) & \rho(L+1) & \cdots & \rho(H+L-1) \end{bmatrix} \in \mathbb{R}^{L \times H}$$

$$R_{c_1,c_2} = \begin{bmatrix} \rho_{c_1,c_2}(0) & \rho_{c_1,c_2}(-1) & \cdots & \rho_{c_1,c_2}(-L+1) \\ \rho_{c_1,c_2}(1) & \rho_{c_1,c_2}(0) & \cdots & \rho_{c_1,c_2}(-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{c_1,c_2}(L-1) & \rho_{c_1,c_2}(L-2) & \cdots & \rho_{c_1,c_2}(0) \end{bmatrix} \in \mathbb{R}^{L \times L}$$

Step 1

$A^{(c)} \in \mathbb{R}^{N \times L}$

$B^{(c)} \in \mathbb{R}^{N \times H}$

각 univariate time series를 **vertical concat**

$B_{cd} = [B^{(1)}, B^{(2)}, \cdots, B^{(C)}]$, $B_{cd} \in \mathbb{R}^{N \times (C*H)}$

$A^{(c_1)T} B^{(c_2)}$ 는 Covariance Matrix의 estimates

$A^{(c_1)T} A^{(c_2)}$ 는 Covariance Matrix의 estimates

Step 2

$\mathcal{L} = \|AW - B\|_F^2$

✓ A는 $A^{(c)}$을 Concat한 것

✓ B는 B 것

$\mathcal{L}_{cd} = \|A$

앞에서 정의된 Loss function와 OLS estimates

$(A_{cd}^T A_{cd}) W_{cd} = A_{cd}^T B_{cd}$에서 각 변수들의 분산이 같다 가정하면 Correlation Matrix

Step 3

OLS Estimator

$A^T AW = A^T B$

✓ Loss function의 OLS 최적해

✓ 모델의 최적해는 아니지만, 최적해일 가능성 존재

OLS coefficients

$$\begin{bmatrix} R'_{1,1} & R'_{1,2} & \cdots & R'_{1,C} \\ R'_{2,1} & R'_{2,2} & \cdots & R'_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ R'_{C,1} & R'_{C,2} & \cdots & R'_{C,C} \end{bmatrix} = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,C} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ R_{C,1} & R_{C,2} & \cdots & R_{C,C} \end{bmatrix} W^*_{cd}$$

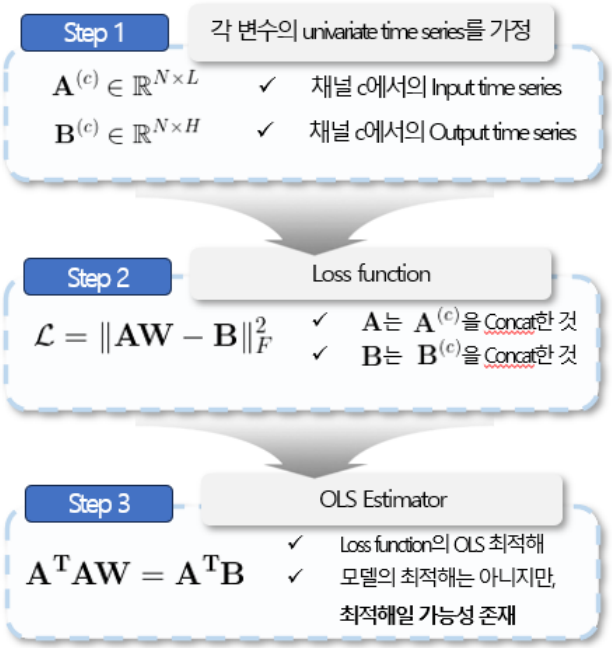
OLS 식을 풀면 **모든 채널 각각의 ACF에 대해** Optimal coefficient가 정해진다

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ CI Alleviates Distribution Drift

- 저자들은 CI Strategy가 시계열에서 일어나는 Distribution Drift를 **완화**시킬 수 있다고 주장



Channel independent

어떻게 Concat?

$$\mathbf{A}_{ci} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \cdots \\ \mathbf{A}^{(\mathbf{C})} \end{bmatrix},\ \mathbf{A}_{ci} \in \mathbb{R}^{(N*C)\times L} \quad \mathbf{B}_{ci} = \begin{bmatrix} \mathbf{B}^{(1)} \\ \mathbf{B}^{(2)} \\ \cdots \\ \mathbf{B}^{(\mathbf{C})} \end{bmatrix},\ \mathbf{B}_{ci} \in \mathbb{R}^{(N*C)\times H}$$

각 univariate time series를 **horizon concat**

Objective Function

$$\mathcal{L}_{ci} = \|\mathbf{A}_{ci}\mathbf{W} - \mathbf{B}_{ci}\|_F^2$$

OLS estimates

$$(\mathbf{A}_{ci}^T\mathbf{A}_{ci})\mathbf{W}_{ci} = \mathbf{A}_{ci}^T\mathbf{B}_{ci}$$

앞에서 정의된 Loss function와 OLS estimates

OLS coefficients

$$\sum_{c=1}^{C} R'_{c,c} = (\sum_{c=1}^{C} R_{c,c})\mathbf{W}^*_{ci}$$

OLS 식을 풀면 **모든 변수 ACF 합에 의해 결정**된다

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ CI Alleviates Distribution Drift

- 저자들은 CI Strategy가 시계열에서 일어나는 Distribution Drift를 **완화**시킬 수 있다고 주장

$$\mathbf{A}_{ci}^T\mathbf{B}_{ci} = \begin{bmatrix}\mathbf{A}^{(1)}\\ \cdots \\ \mathbf{A}^{(C)}\end{bmatrix}^T \begin{bmatrix}\mathbf{B}^{(1)}\\ \cdots \\ \mathbf{B}^{(C)}\end{bmatrix} = \sum_{c=1}^{C}\mathbf{A}_{(c)}^T\mathbf{B}_{(c)}$$

$$\mathbf{A}_{ci}^T\mathbf{A}_{ci} = \begin{bmatrix}\mathbf{A}^{(1)}\\ \mathbf{A}^{(2)}\\ \cdots \\ \mathbf{A}^{(C)}\end{bmatrix}^T \begin{bmatrix}\mathbf{A}^{(1)}\\ \mathbf{A}^{(2)}\\ \cdots \\ \mathbf{A}^{(C)}\end{bmatrix} = \sum_{c=1}^{C}\mathbf{A}_{(c)}^T\mathbf{A}_{(c)}$$

horizon concat

$\mathbf{A}^{(c_1)^T}\mathbf{B}^{(c_2)}$ 는 Covariance Matrix의 estimates

$\mathbf{A}^{(c_1)^T}\mathbf{A}^{(c_2)}$ 는 Covariance Matrix의 estimates

앞에서 정의된 Loss function와 OLS estimates

$(\mathbf{A}_{ci}^T\mathbf{A}_{ci})\mathbf{W}_{ci} = \mathbf{A}_{ci}^T\mathbf{B}_{ci}$ 에서 각 변수들의 분산이 같다 가정하면 Correlation Matrix

OLS coefficients

$$\sum_{c=1}^{C} R'_{c,c} = (\sum_{c=1}^{C} R_{c,c})\mathbf{W}^*_{ci}$$

OLS 식을 풀면 **모든 채널 ACF 합에 의해 결정**된다

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ CI Alleviates Distribution Drift

- 저자들은 CI Strategy가 시계열에서 일어나는 Distribution Drift를 **완화**시킬 수 있다고 주장

Channel dependent

OLS 식을 풀면 **모든 변수 각각의 ACF에 대해** Optimal coefficient가 정해진다

Channel Independent

OLS 식을 풀면 **모든 변수 ACF 합에 의해 결정**된다

Conclusion

Train과 Test 간
변수 내 ACF 차이가 커지면 **CD가 불리**
ACF 합의 차이가 커지면 **CI**가 불리

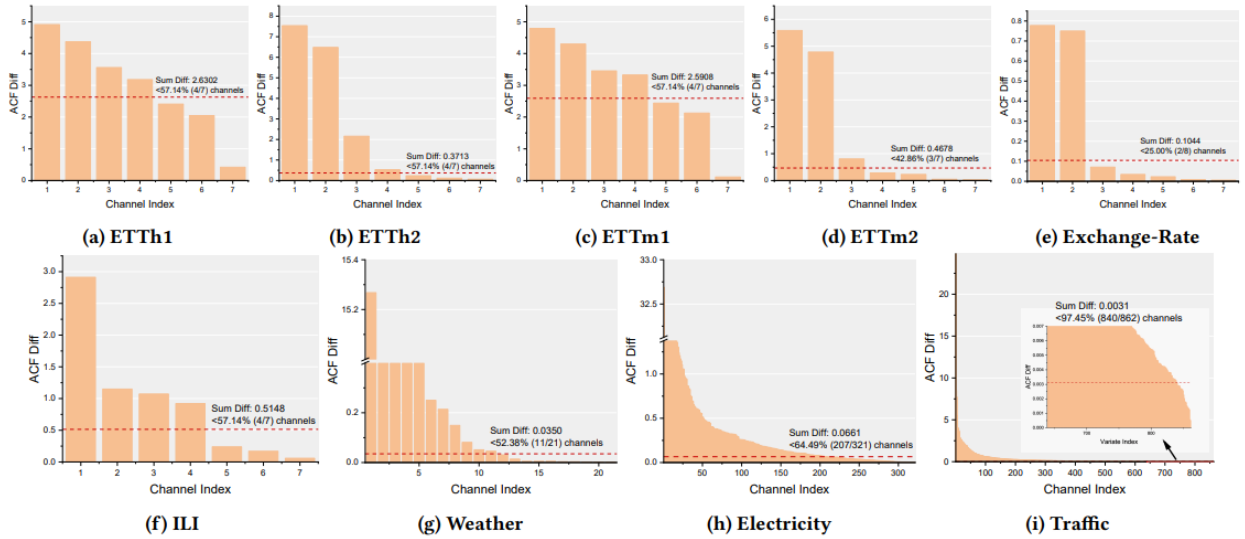
Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ CI Alleviates Distribution Drift

- 저자들은 CI Strategy가 시계열에서 일어나는 Distribution Drift를 **완화**시킬 수 있다고 주장

How about benchmark dataset?



대부분 데이터셋의 50% 이상의 변수에서 **ACF의 평균 차이는 변수 내 ACF보다 작음**
-> CI가 Distribution Drift 상황에서 더 유리하다!

Conclusion

Train과 Test 간
변수 내 ACF 차이가 커지면 **CD가 불리**
ACF 합의 차이가 커지면 **CI**가 불리

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ **Experiments**

- 8개의 다변량 시계열 데이터셋에 대해 CD Strategy와 CI Strategy 비교
- 머신러닝 기반, Linear 기반, CNN 기반, RNN 기반, Transformer 기반 등 다양한 모델에 대해 CI Strategy를 적용 후 성능 비교

Table 1: Statistics of nine multivariate time series datasets.

Dataset(s)	channels	Timesteps	Granularity
ETTh1&ETTh2	7	17,420	1hour
ETTm1&ETTm2	7	69,680	5min
Traffic	862	17,544	1hour
Electricity	321	26,304	1hour
Exchange-Rate	8	7,588	1day
Weather	21	52,696	10min
ILI	7	966	1week

Method

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

❖ Experiments

- Dataset과 Method에 관계 없이 CI가 평균적으로 19%~38% 성능 우세
- **실험적으로 CI는 CD를 압도**한다!

Table 3: MSE on nine multivariate time series datasets across various forecasting models.

Dataset	Electricity		ETTh1		ETTh2		ETTm1		ETTm2		Exchange_Rate		Traffic		Weather		ILI		Mean
Horizon	48	96	48	96	48	96	48	96	48	96	48	96	48	96	48	96	24	36	
Linear (CD)	0.442	0.444	0.402	0.514	0.711	1.520	0.404	0.433	0.161	0.269	0.119	0.274	-	-	0.142	0.165	**2.343**	**2.436**	**11/16**
Linear (CI)	0.195	0.196	**0.345**	**0.386**	**0.226**	**0.319**	0.354	**0.351**	**0.147**	**0.189**	**0.051**	**0.088**	0.703	0.651	0.169	0.202	2.847	2.857	4/16
Improve (%)	+55.88	+55.91	+14.17	+24.94	+68.16	+79.04	+12.20	+18.85	+8.45	+29.73	+56.95	+67.84	-	-	-19.15	-22.16	-21.52	-17.29	+25.75
GBRT (CD)	-	-	0.497	0.592	1.039	1.633	0.428	0.500	0.370	0.606	0.919	1.387	-	-	0.539	0.475	5.128	4.845	12/14
GBRT (CI)	**0.165**	0.171	0.365	0.414	0.636	1.167	**0.341**	0.367	0.236	0.318	0.270	0.335	**0.532**	0.550	**0.146**	**0.185**	5.186	4.983	0/14
Improve (%)	-	-	+26.63	+29.99	+38.77	+28.57	+20.43	+26.63	+36.13	+47.50	+70.58	+75.87	-	-	+72.96	+61.01	-1.13	-2.85	+37.93
MLP (CD)	0.293	0.305	0.517	0.695	1.664	3.651	0.453	0.507	0.323	0.303	0.590	0.802	1257.104	1118.137	0.140	0.167	2.959	3.494	12/18
MLP (CI)	0.199	0.199	0.360	0.408	0.254	0.321	0.457	0.513	0.157	0.197	0.172	0.118	0.666	0.639	0.169	0.202	3.618	3.840	3/18
Improve (%)	+32.31	+34.57	+30.40	+41.36	+84.74	+91.22	-0.73	-1.14	+51.33	+34.85	+70.87	+85.28	+99.95	+99.94	-21.35	-21.37	-22.28	-9.88	+37.78
DeepAR (CD)	0.316	0.293	0.755	0.918	1.326	1.609	0.736	0.735	0.444	0.747	0.912	1.093	0.644	0.691	0.380	0.473	5.593	5.418	14/18
DeepAR (CI)	0.231	0.247	0.723	0.724	0.601	0.714	0.616	0.566	0.200	0.268	0.824	0.878	0.641	0.708	0.173	0.221	4.590	4.501	0/18
Improve (%)	+26.73	+15.65	+4.30	+21.08	+54.71	+55.63	+16.26	+22.92	+54.91	+64.16	+9.67	+19.65	+0.55	-2.48	+54.38	+53.22	+17.94	+16.92	+28.12
TCN (CD)	0.359	0.383	0.735	0.890	1.453	1.539	1.095	0.834	0.858	1.114	1.453	1.334	1.088	1.095	0.377	0.348	5.224	4.775	13/18
TCN (CI)	0.258	0.290	0.401	0.507	0.404	0.663	0.614	0.534	0.251	0.313	1.488	1.562	0.784	0.835	0.290	0.339	6.671	5.142	2/18
Improve (%)	+28.28	+24.47	+45.40	+42.99	+72.17	+56.94	+43.93	+35.90	+70.77	+71.87	-2.41	-17.11	+27.98	+23.73	+23.05	+2.56	-27.70	-7.68	+28.62
Informer (CD)	0.326	0.349	0.689	0.959	1.270	3.137	0.517	0.632	0.310	0.370	0.790	0.894	0.715	0.736	0.322	0.301	5.377	5.288	16/18
Informer (CI)	0.208	0.183	0.560	0.532	0.311	0.382	0.366	0.426	0.156	0.262	0.169	0.190	0.601	0.549	0.162	0.260	4.980	5.254	0/18
Improve (%)	+36.07	+47.47	+18.67	+44.58	+75.49	+87.83	+29.29	+32.61	+49.70	+29.10	+78.64	+78.69	+15.95	+25.43	+49.74	+13.41	+7.38	+0.65	+40.04
Transformer (CD)	0.250	0.257	0.861	0.966	1.031	1.868	0.458	0.554	0.281	0.520	0.511	0.659	0.645	0.650	0.251	0.423	5.309	5.406	17/18
Transformer (CI)	0.185	**0.163**	0.655	0.533	0.274	0.466	0.379	0.496	0.148	0.237	0.101	0.137	0.558	**0.526**	0.168	0.225	4.307	5.033	0/18
Improve (%)	+26.10	+36.59	+23.85	+44.84	+73.43	+75.07	+17.32	+10.43	+47.27	+54.40	+80.27	+79.30	+13.43	+19.13	+33.14	+46.87	+18.88	+6.89	+39.29

Conclusion

The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting

- **Transformer 기반 시계열 예측**
  - RNN은 장기 예측에서의 한계 존재, 이를 Transformer로 해결하려는 시도 등장
- **Are Transformers Effective for Time Series Forecasting?**
  - Transformer 모델이 시계열 예측에 효과적인가에 대한 의문
  - 간단한 Linear 모델로 Transformer based 모델의 성능을 증가
- **A Time Series is Worth 64 Words: Long-Term Forecasting with Transformers**
  - Channel Independent Strategy를 도입하여 Transformer 모델의 예측 성능 향상
- **The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting**
  - Channel Independent Strategy의 예측 성능이 좋은 이유를 이론적, 실험적으로 설명

고맙습니다