Paradigm Shifts in Time Series Forecasting

DMQA Open Seminar (2026. 05. 29)

Data Mining & Quality Analytics Lab.

박성수

발표자 소개

About Me

❖ 박성수 (Sungsu Park)

- 고려대학교 산업경영공학과 석박사통합과정 (2026.03 ~ Present)
- Data Mining & Quality Analytics Lab. (김성범 교수님)

❖ Research Interest

- Multivariate Time Series Forecasting

❖ Contact

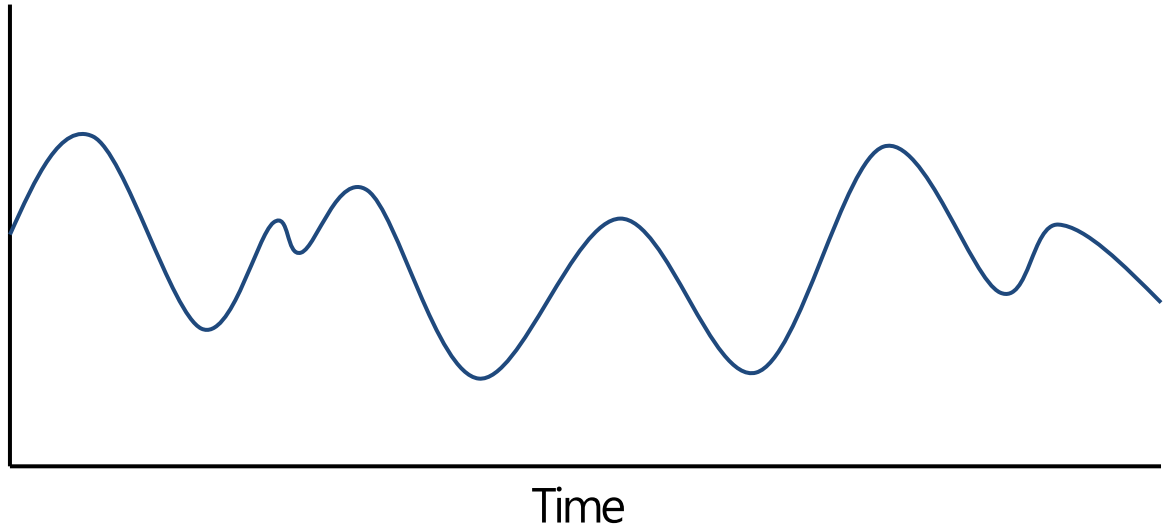
- sspark1@korea.ac.kr

Introduction

Time Series Forecasting (TSF)

❖ 시계열 데이터란?

- 일정한 시간 간격으로 수집된 순차적 데이터 포인트의 집합



❖ 시계열 데이터의 중요한 특성은 무엇일까?

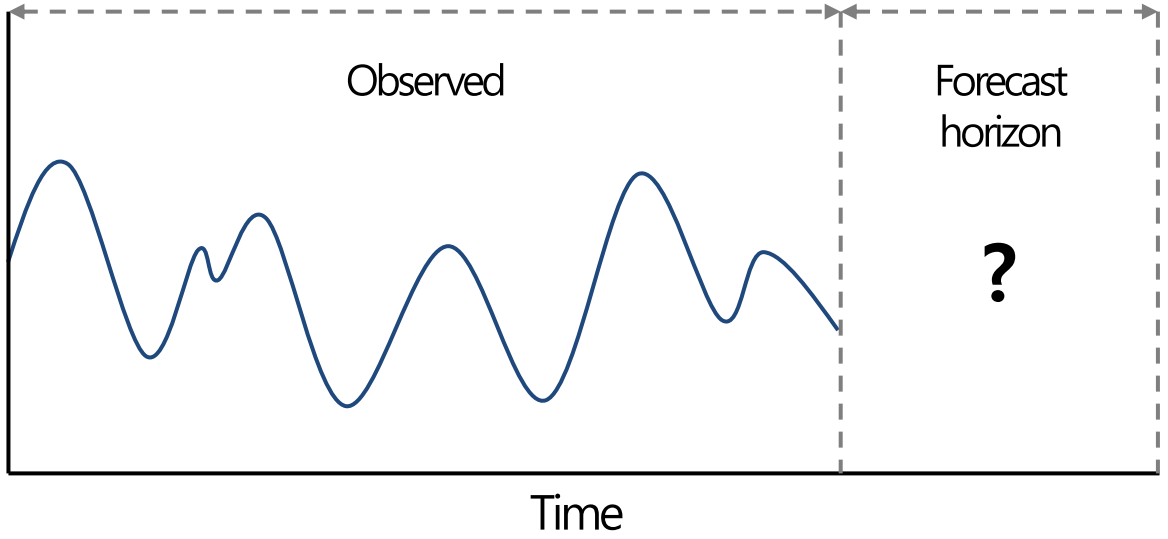
- **시간적 순서(Temporal order)** → 데이터 포인트 간의 선후 관계가 존재
- **자기상관성(Autocorrelation)** → 현재의 값이 과거의 패턴에 의존하는 특성
- **비정상성(Non-stationarity)** → 평균과 분산이 시간에 따라 변하는 통계적 불안정성

Introduction

Time Series Forecasting (TSF)

❖ 시계열 예측이란?

- 관측된 시계열 데이터를 분석하여 미래를 예측하는 문제

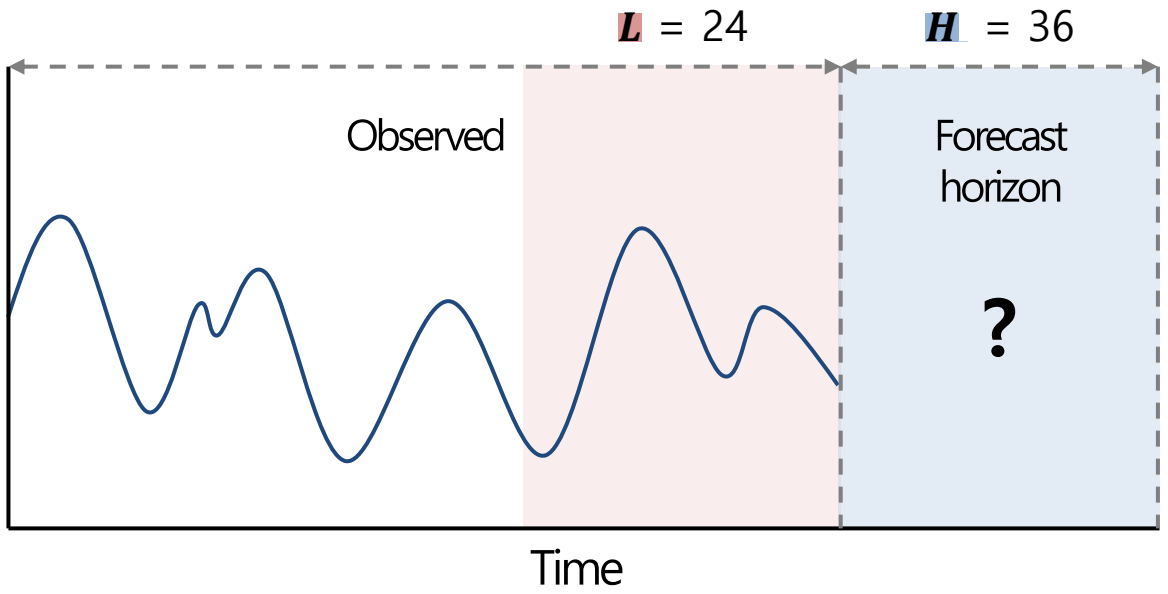


Introduction

Time Series Forecasting (TSF)

❖ 시계열 예측이란?

- 관측된 시계열 데이터를 분석하여 미래를 예측하는 문제



❖ $\hat{Y}_{t+1:t+H} = f_\theta(X_{t-L+1:t})$

- L = lookback window
- H = forecast horizon
- t = 현재 시점

Background

Evolution of Time Series Forecasting Models

❖ 시계열 예측 모델은 어떻게 발전해왔을까?

Conventional methods (statistical)

Background

Conventional Methods: Statistical

❖ Exponential Smoothing

- 최근 관측값에 더 큰 가중치를 두고, 과거로 갈수록 지수적으로 감소하는 가중치를 부여

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1}$$

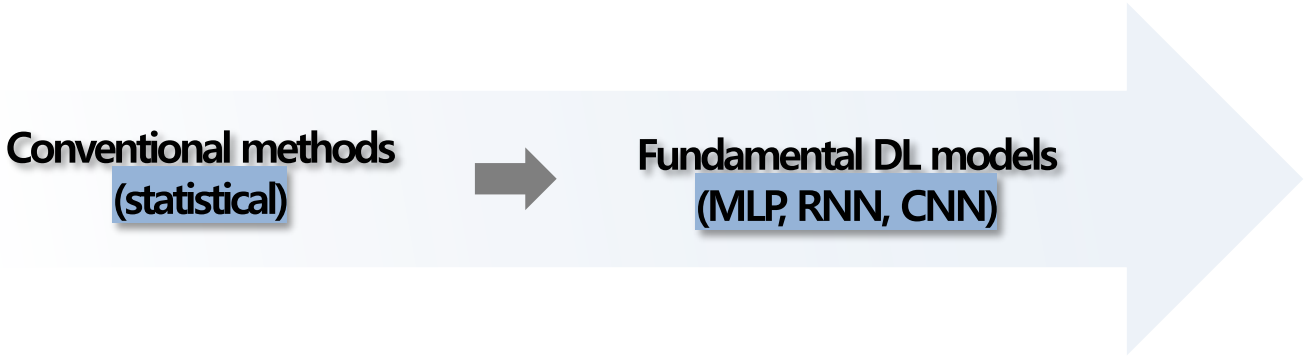
❖ ARIMA

- 자기상관성을 이용하면서 차분을 통해 비정상성을 완화한 뒤 예측
- AR(AutoRegressive): 과거 값의 선형 결합
- I (Integrated): 차분을 통해 비정상성 제거
- MA (Moving Average): 과거 오차 항의 선형 결합

Background

Evolution of Time Series Forecasting Models

❖ 시계열 예측 모델은 어떻게 발전해왔을까?

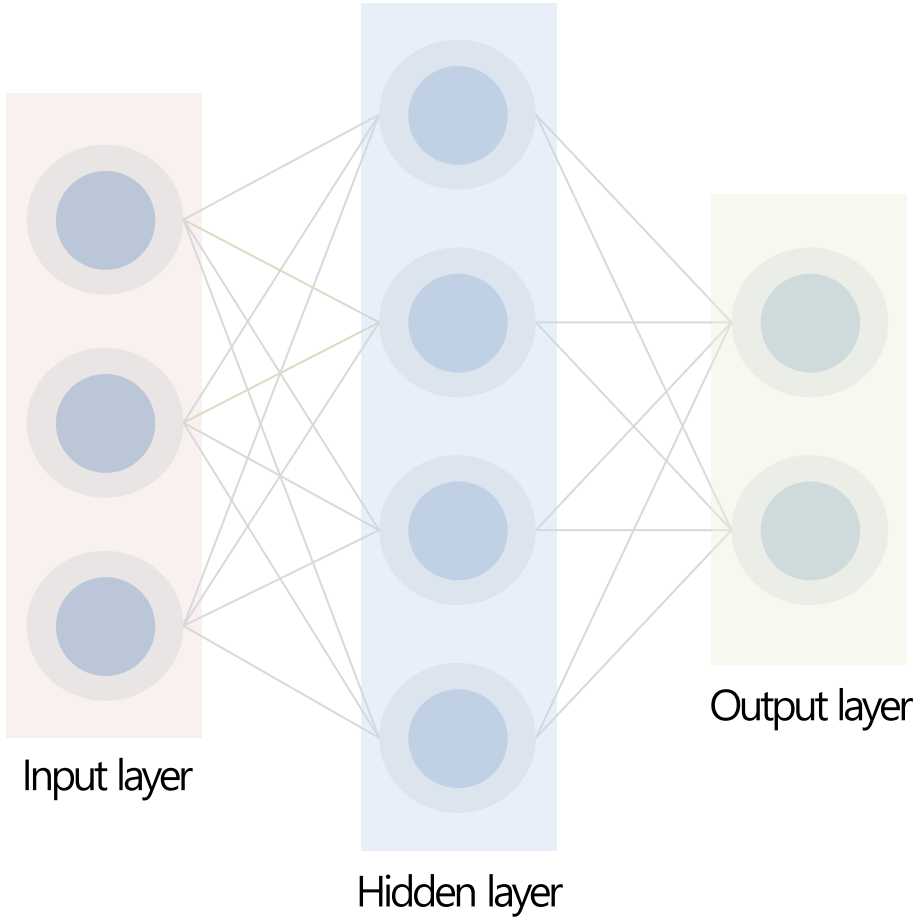


Method

Fundamental DL models: MLP

❖ Multi-layer perceptron

- 비선형 패턴 모델링에서 강력한 성능을 보여줌

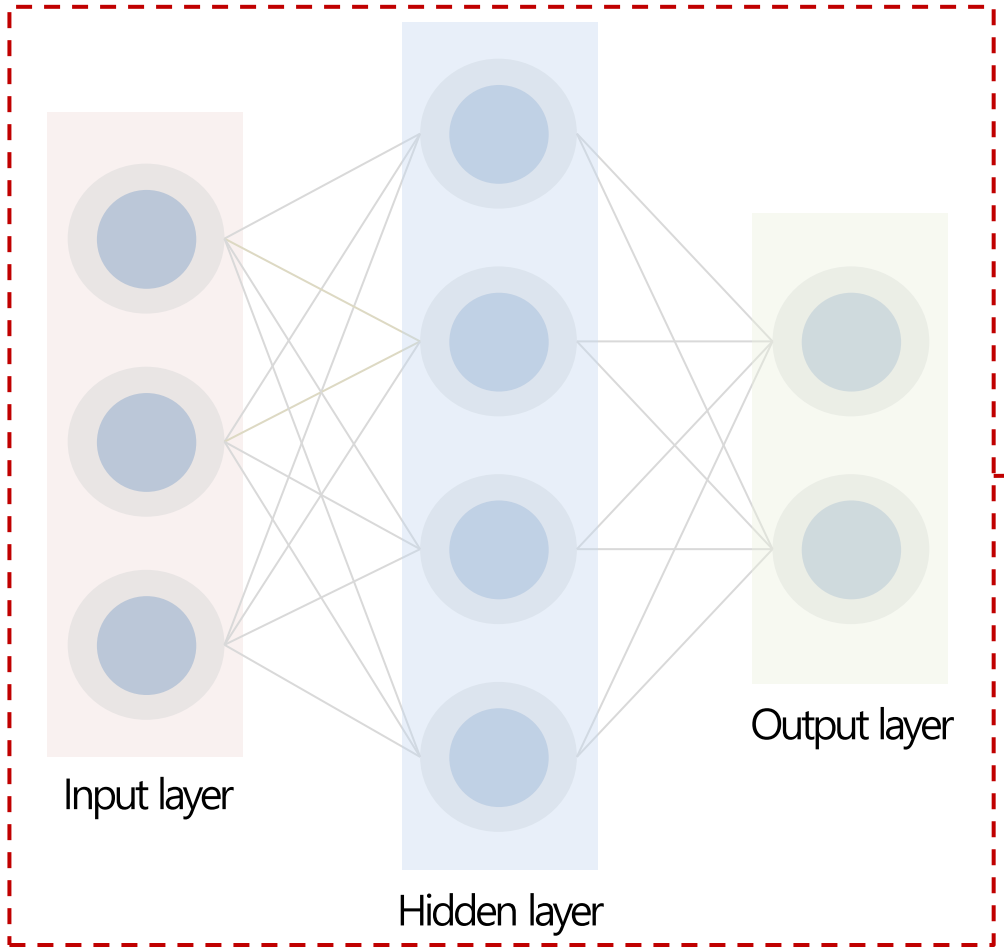


Method

Fundamental DL models: MLP

❖ Multi-layer perceptron

- 비선형 패턴 모델링에서 강력한 성능을 보여줌



Limitation ①: weak temporal inductive bias

- 시계열의 순서성과 시간적 의존성을 반영하기 어려움

Limitation ②: fixed lookback window

- 다양한 lookback window에 유연하게 대응하기 어려움

Limitation ③: parameter/data inefficiency

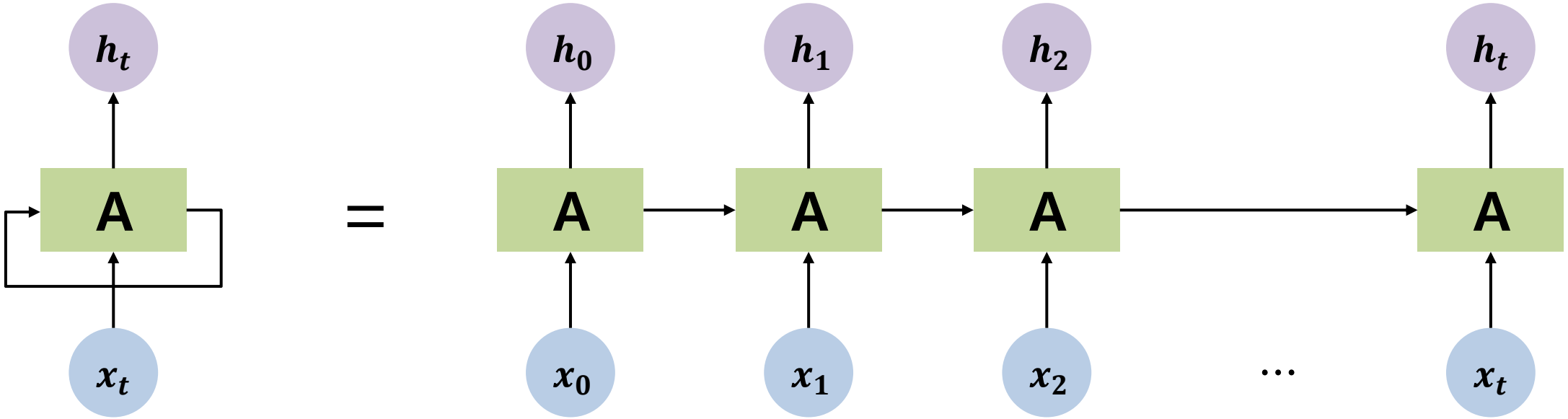
- 긴 lookback이나 많은 channel을 펼쳐 넣으면 parameter 수가 증가하고, 데이터 효율성이 낮아질 수 있음

Method

Fundamental DL models: RNN

❖ **Recurrent Neural Network (RNN)를 time series data에 어떻게 적용했을까?**

- 시계열 데이터와 같은 **sequence data를 처리하도록 설계된 모델**



Method

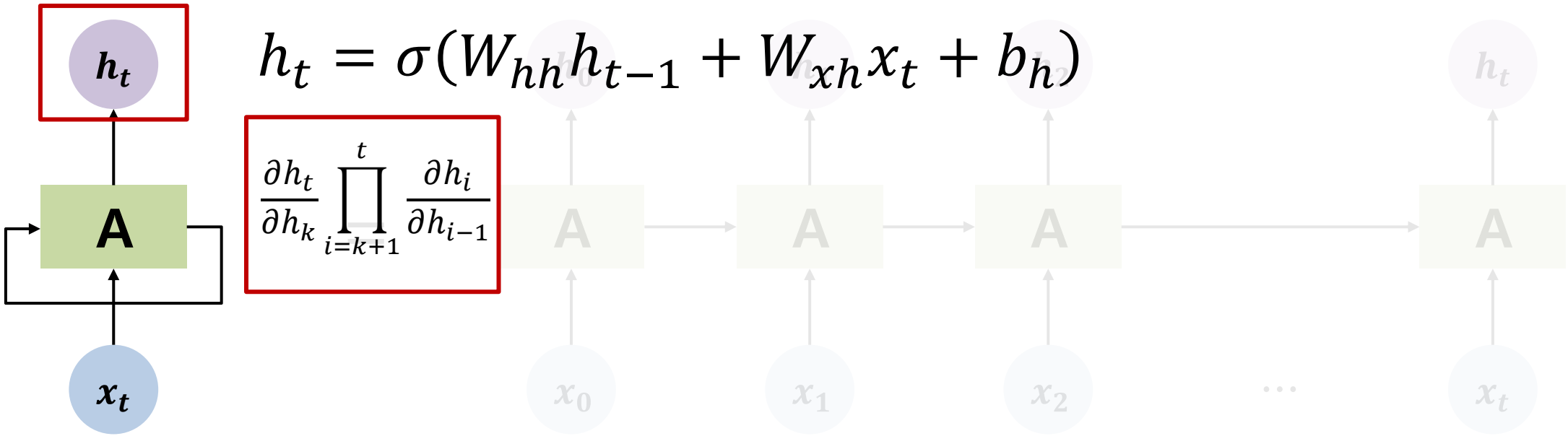
Fundamental DL models: RNN

❖ **Recurrent Neural Network (RNN)를 time series data에 어떻게 적용했을까?**

- 시계열 데이터와 같은 **sequence data를 처리하도록 설계된 모델**

기울기 소실/폭주 문제 발생!

시간축 병렬 처리 제한!

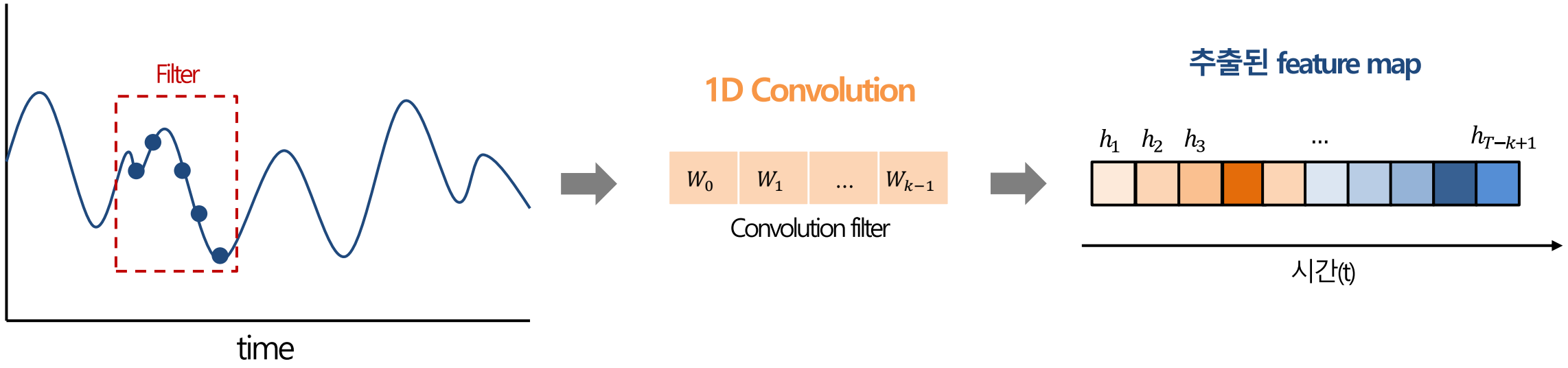


Method

Fundamental DL models: CNN

❖ **Convolutional Neural Network (CNN)을 시계열 예측에 어떻게 적용했을까?**

- 시계열 데이터를 시간축을 따라 흐르는 1차원 신호로 간주함
- 시계열의 국소적 영역에서 특징을 추출하여 **지역적 패턴(Local patterns)을 학습함**



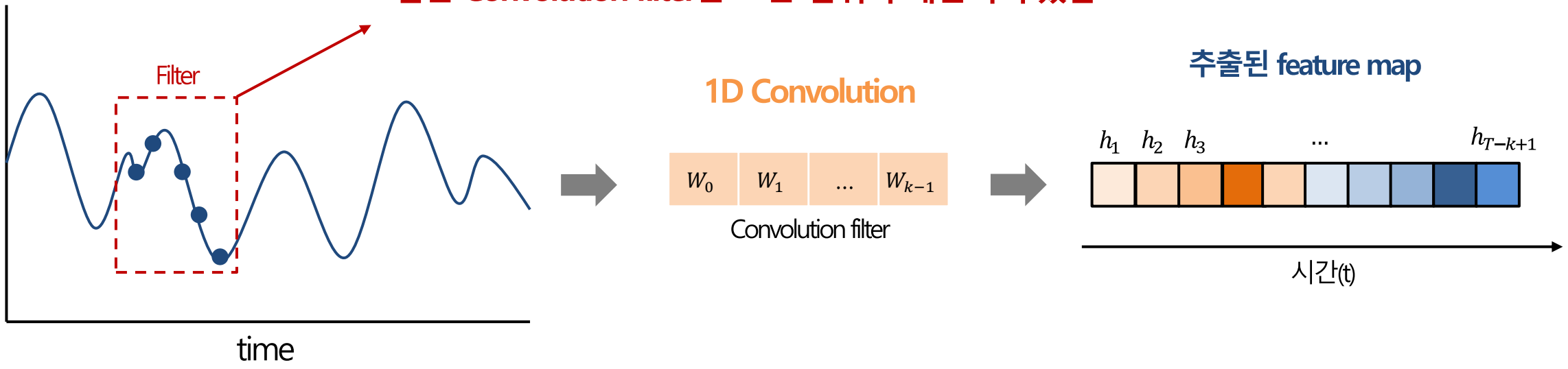
Method

Fundamental DL models: CNN

❖ **Convolutional Neural Network (CNN)을 시계열 예측에 어떻게 적용했을까?**

- 시계열 데이터를 시간축을 따라 흐르는 1차원 신호로 간주함
- 시계열의 국소적 영역에서 특징을 추출하여 **지역적 패턴(Local patterns)을 학습함**

얕은 Convolution filter는 보는 범위가 제한되어 있음

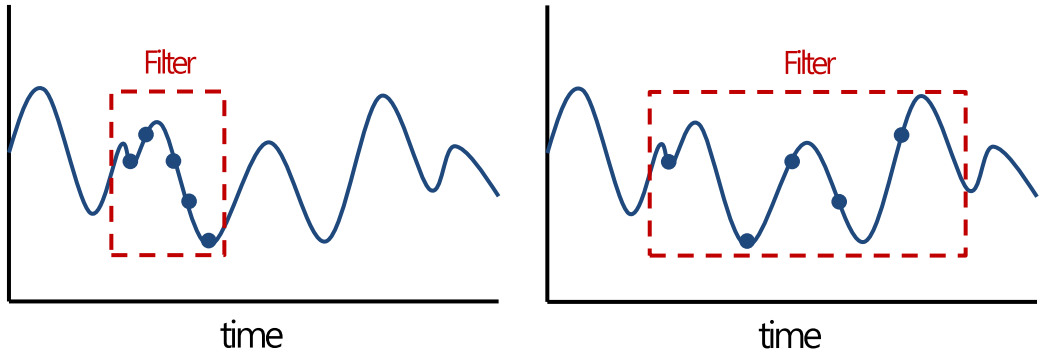


Method

Fundamental DL models: CNN

❖ **Dilated convolution은 왜 더 긴 과거를 볼 수 있을까?**

- 일반적인 CNN은 가까운 과거의 국소 정보만 보는 한계가 있음
- Dilation 계수 d를 키우면 파라미터 수 증가 없이 receptive field 확장 가능

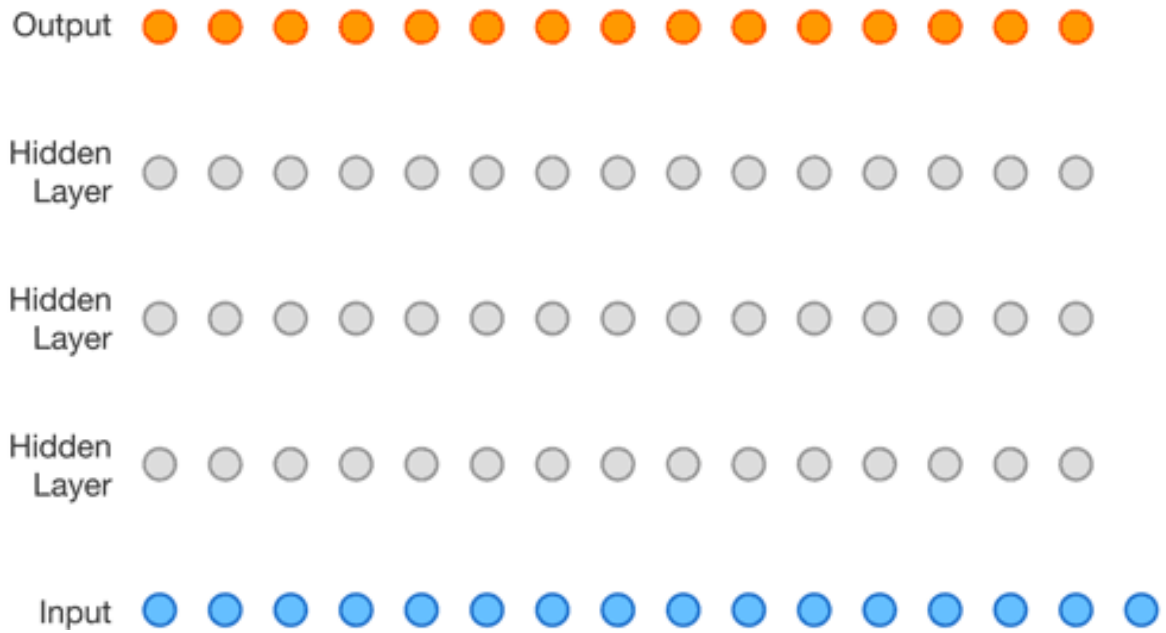


좁은 receptive field

더 넓은 receptive field

$$(x * f)(s) = \sum_{i} x(i) \cdot f(s-i)$$

$$(F *_d x)(s) = \sum_{i=0}^{k-1} f(i) \cdot x(s - d \cdot i)$$



Dilated convolution stacking

Background

Evolution of Time Series Forecasting Models

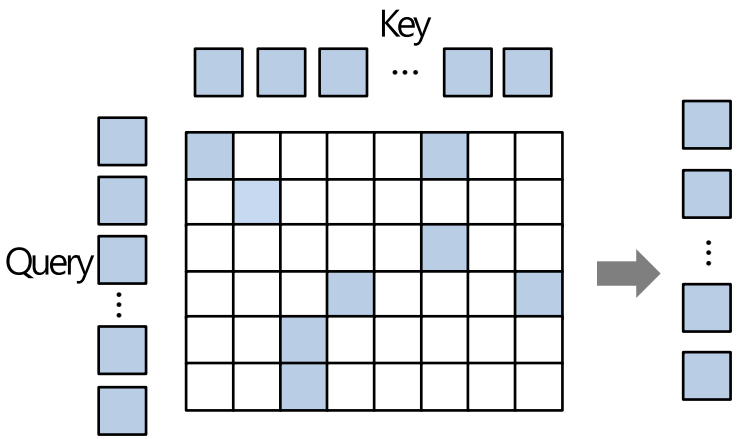
❖ 시계열 예측 모델은 어떻게 발전해왔을까?

Related Works

Transformer

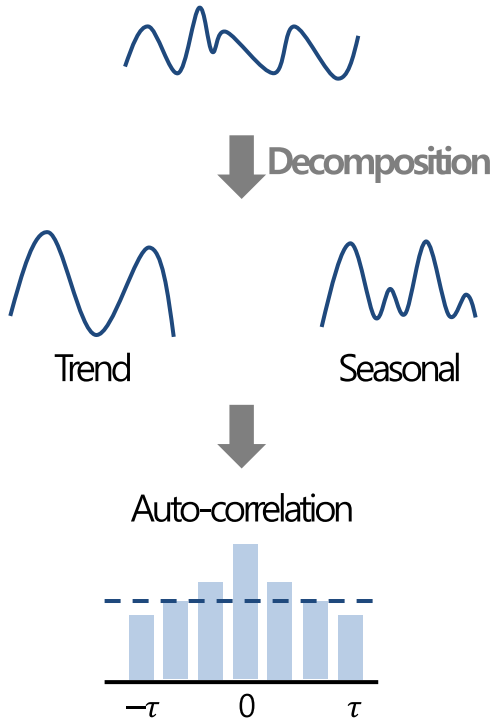
❖ 초기 transformer 기반 예측 모델은 무엇이 있을까?

Informer (AAAI, 2021)



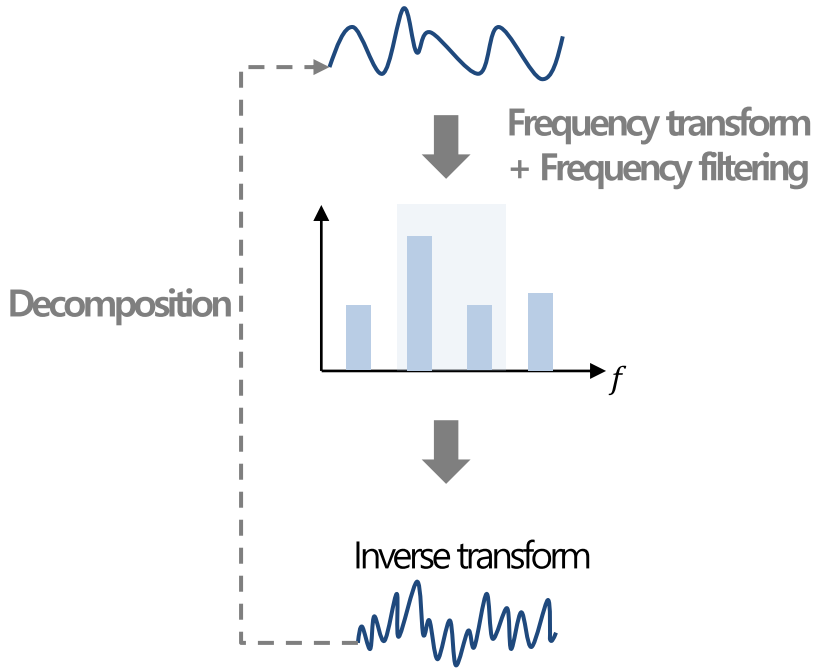
ProbSparse attention으로 긴 시계열의 계산량을 줄임

Autoformer (NeurIPS, 2021)



Decomposition과 auto-correlation으로 장기 패턴을 포착

FEDformer (ICML, 2022)



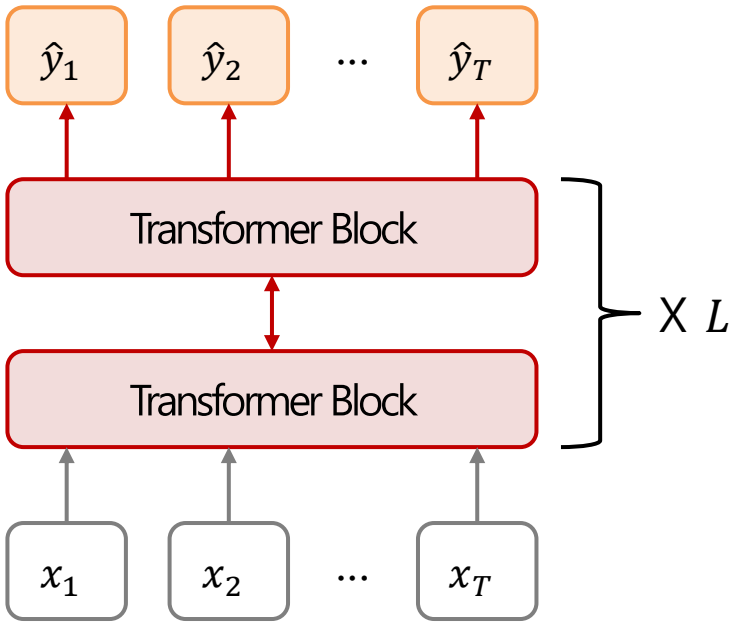
Frequency domain 정보를 활용해서 효율성과 성능 개선

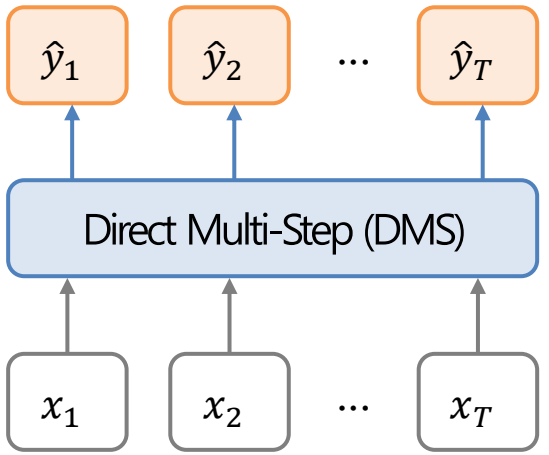
Background

Transformer

❖ **Transformer기반 모델의 한계는 무엇이었을까?**

- Self-attention은 pair-wise correlation에 강하지만, 시계열의 순서성과 연속성을 직접 보장하지는 않음
- 긴 lookback window를 늘려도 성능이 항상 좋아지지 않는 현상이 관찰됨
- 기존 비교 실험에서 단순하지만 강력한 DMS baseline이 충분히 고려되지 않았음





Related Works

Transformer

❖ **Are Transformer effective for Time Series Forecasting?** (Zeng et al., AAAI 2023)

- Transformer 기반 LTSF 모델의 성능 향상이 정말 self-attention 때문인지 재검토
- 단순한 DLinear baseline을 통해 "복잡한 Transformer가 항상 필요한가?"라는 질문 제기
- 이후 시계열 예측에서 모델 구조보다 시계열에 맞는 표현 방식이 중요하다는 논의로 확장

Are Transformers Effective for Time Series Forecasting?

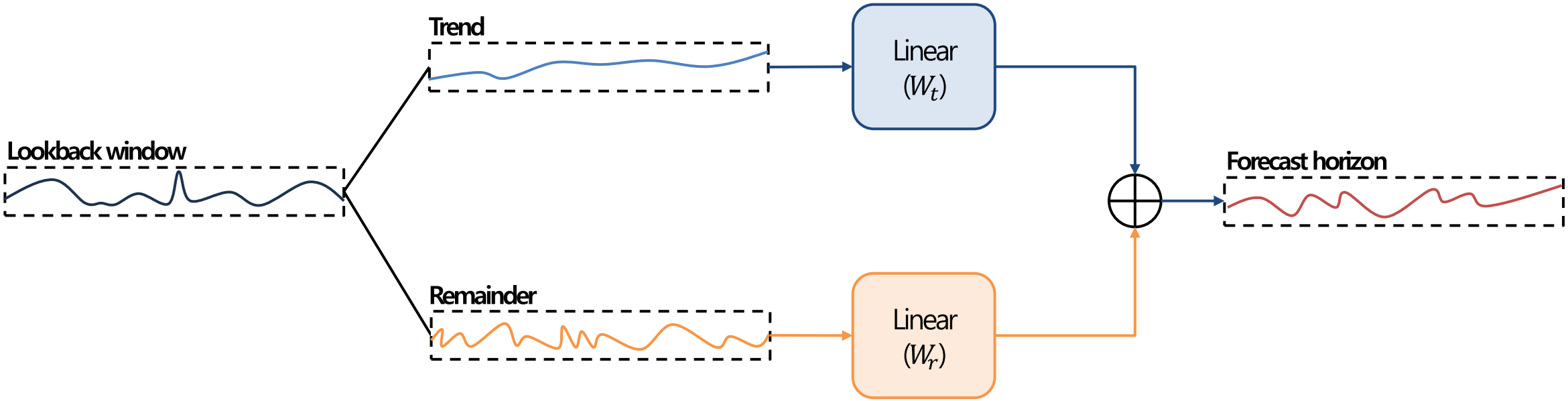
Ailing Zeng[1]*, Muxi Chen[1]*, Lei Zhang[2], Qiang Xu[1]
[1]The Chinese University of Hong Kong
[2]International Digital Economy Academy (IDEA)
{alzeng, mxchen21, qxu}@cse.cuhk.edu.hk
{leizhang}@idea.edu.cn

Method

Transformer

❖ 왜 DLinear가 중요한 전환점이 되었을까?

- 복잡한 attention 없이도 direct multi-step 예측 성능이 좋다는 것을 보여줌
- 입력 시계열을 trend와 remainder로 분해한 뒤, 각 성분에 linear layer를 적용

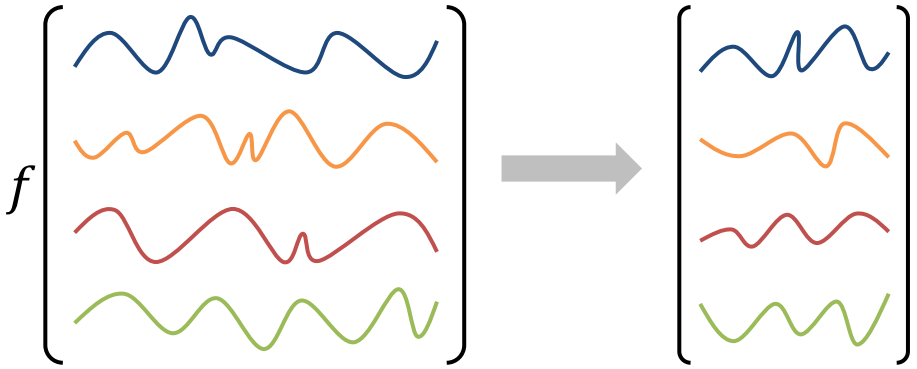


Method

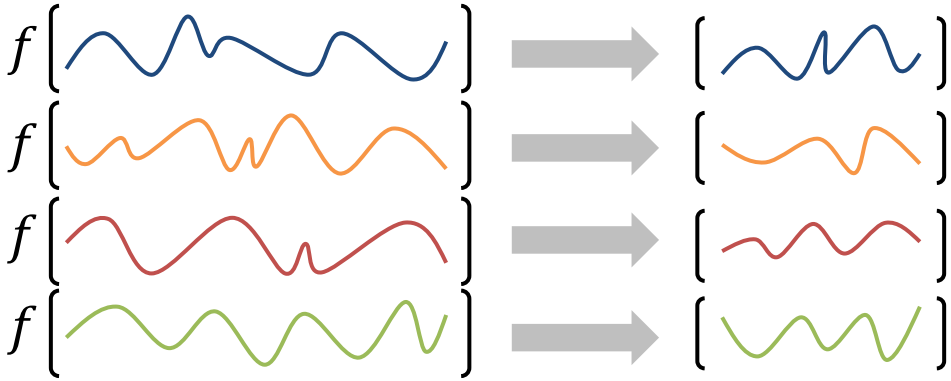
Transformer

❖ **이후 어떤 고민을 하게 되었을까?**

- 복잡한 attention을 쓰는 것보다, 시계열 구조에 맞는 단순하고 안정적인 표현이 중요해짐
- 특히 다변량 시계열에서는 **"채널을 독립적으로 볼 것인가, 함께 볼 것인가"**가 핵심 문제
- 이 논점이 **Channel Independence(CI)와 Channel Dependence(CD) 전략으로 이어짐**



Channel Dependent (CD)



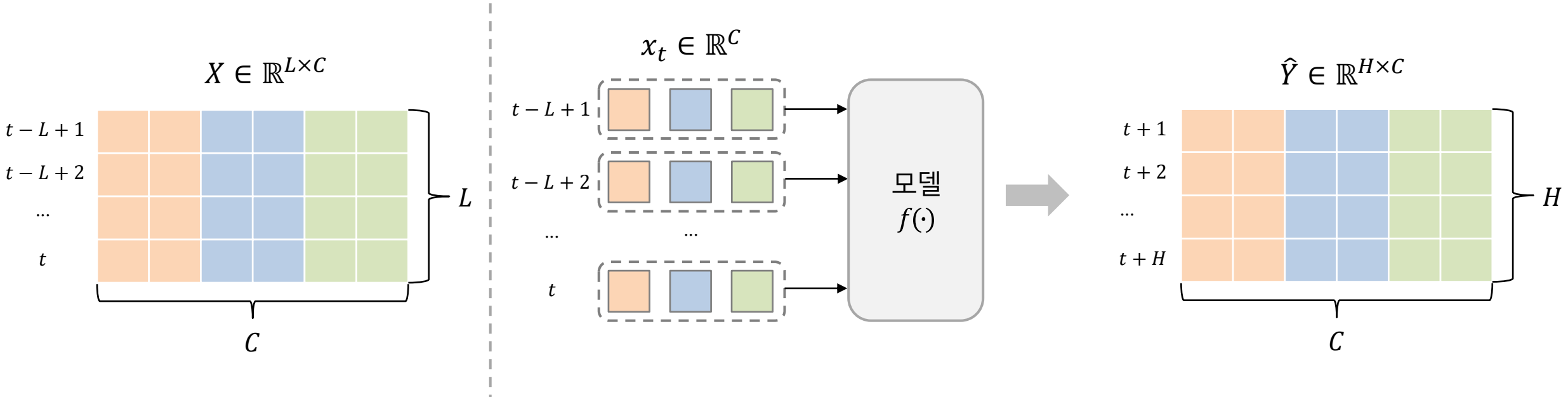
Channel Independent (CI)

Method

Transformer

❖ Channel-Dependent (CD) 전략이란 무엇일까?

- 시점 t에서 여러 변수를 하나의 벡터 $x_t = [x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(C)}]$로 묶어서 함께 처리함
- 예측 함수는 전체 행렬 $X \in \mathbb{R}^{L \times C}$를 입력으로 받아 **채널 간 상관관계를 학습**
- **과거 관계를 너무 강하게 학습했다면 오히려 미래 예측에서 불안정**

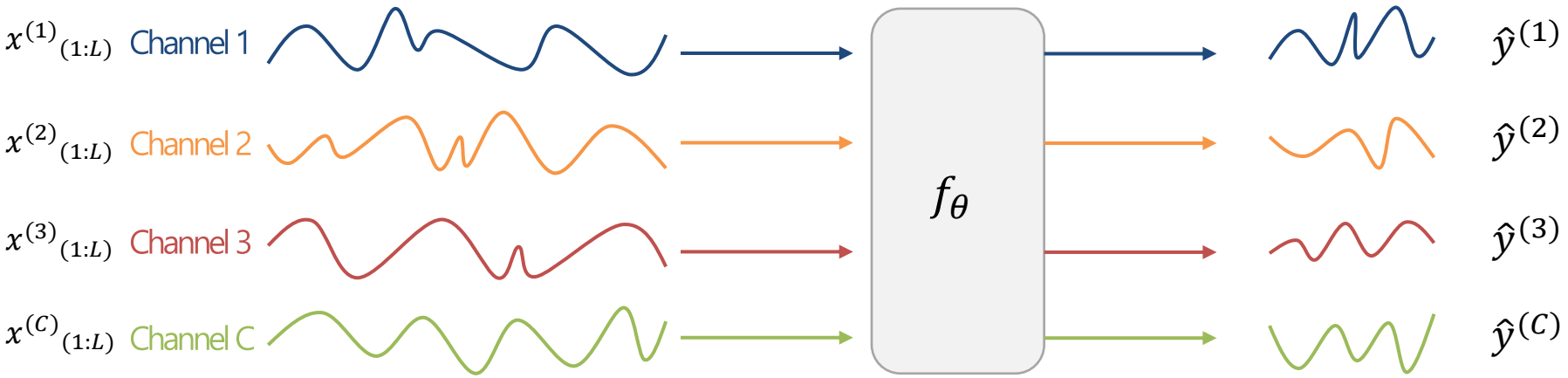


Method

Transformer

❖ Channel-Independent (CI) 전략이란 무엇일까?

- 다변량 시계열 X를 C개의 단변량 시계열 샘플로 처리
- 각 채널 c에 대해 과거 ${x^{(c)}}_{(1:L)}$만 사용하여 미래 $\hat{y}^{(c)}$를 예측
- **Distribution drift에 상대적으로 강하고, 채널 간 관계 변화에 덜 민감함**



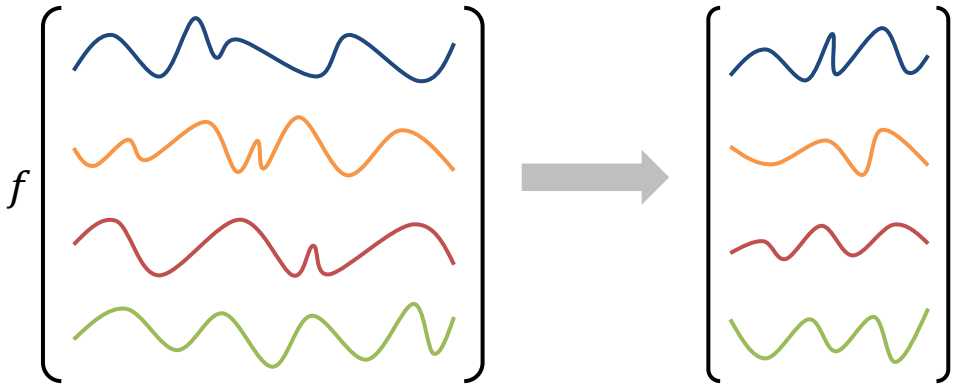
Method

Transformer

❖ **그렇다면 CI 전략만으로 충분할까?**

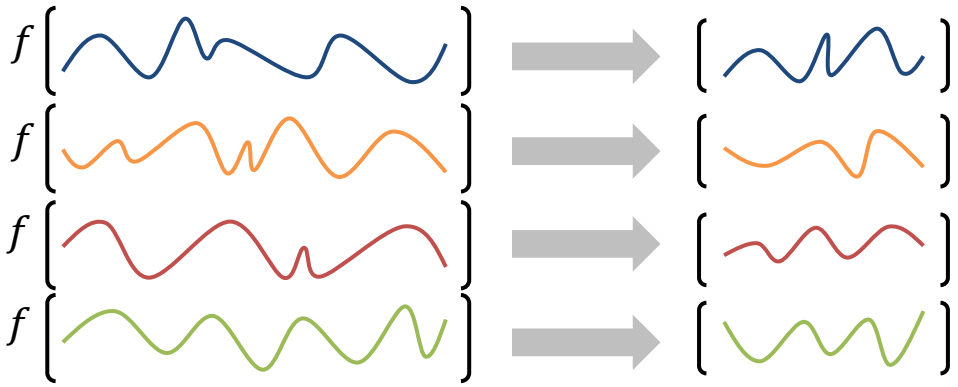
- CI는 각 채널을 독립적으로 다루기 때문에 drift에 비교적 안정적임
- 하지만 다른 변수의 정보를 직접 활용하지 못해 변수 간 상호작용을 놓칠 수 있음

Channel Independent



변수 간 상호작용을 예측 신호로 활용

Channel Dependent



다른 채널의 정보는 직접 사용하지 않음

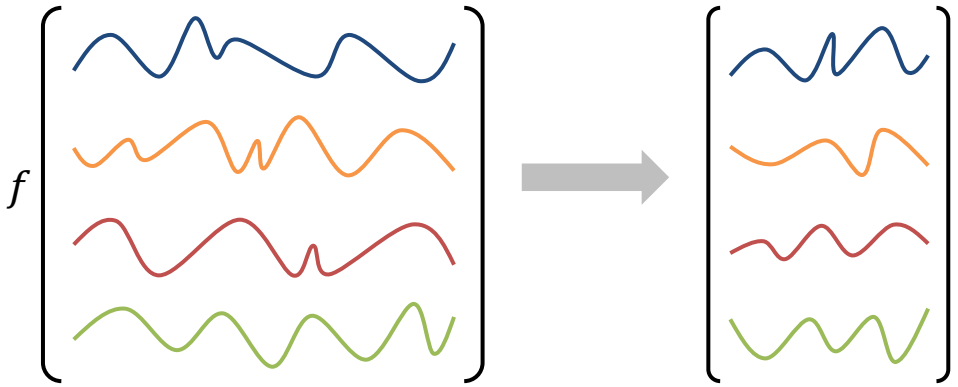
Method

Transformer

❖ **그렇다면 CI 전략만으로 충분할까?**

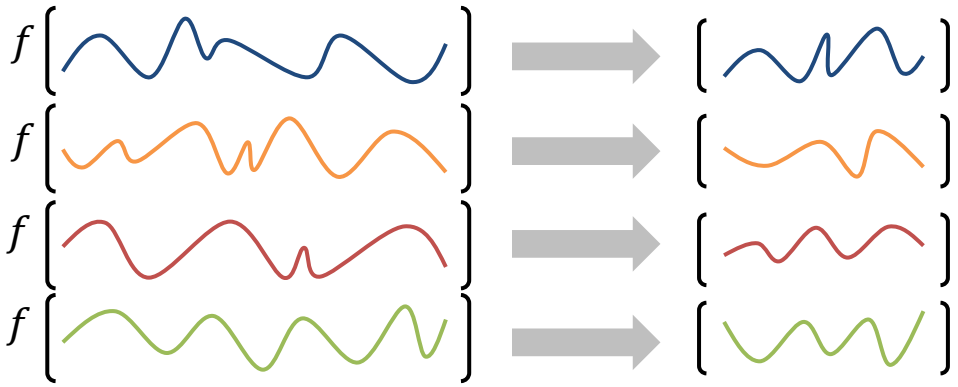
- CI는 각 채널을 독립적으로 다루기 때문에 drift에 비교적 안정적임
- 하지만 다른 변수의 정보를 직접 활용하지 못해 변수 간 상호작용을 놓칠 수 있음

Channel Independent



변수 간 상호작용을 예측 신호로 활용

Channel Dependent



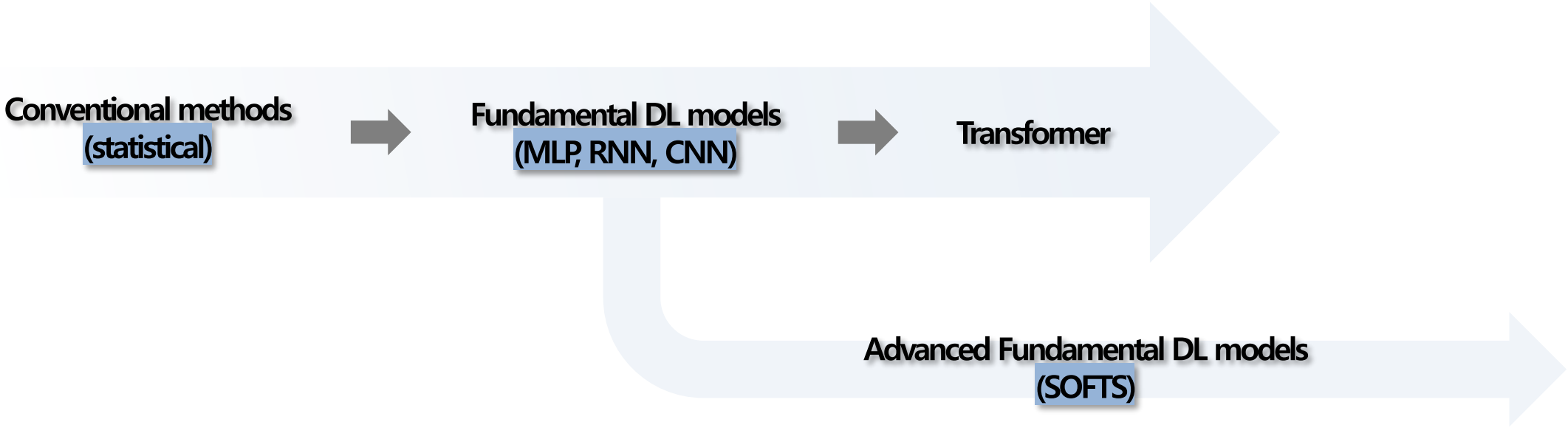
다른 채널의 정보는 직접 사용하지 않음

CD는 더 풍부한 정보를 활용할 수 있으므로, 두 전략을 함께 쓰는 방향이 중요!

Introduction

Evolution of Time Series Forecasting Models

❖ 시계열 예측 모델은 어떻게 발전해왔을까?



Related Works

Advanced Fundamental DL models: SOFTS

❖ **SOFTS: Efficient Multivariate Time Series Forecasting with Series-Core Fusion** (Han et al., NeurIPS 2024)

- **STAR 모듈**을 통해 여러 채널 정보를 core representation으로 집약하고, 이를 각 채널에 다시 전달하는 효율적인 구조 제안
- 여러 시계열 채널을 직접 서로 비교하지 않고, 하나의 **core representation**을 거쳐 간접적으로 정보를 주고 받음

SOFTS: Efficient Multivariate Time Series Forecasting with Series-Core Fusion

Lu Han, Xu-Yang Chen, Han-Jia Ye, De-Chuan Zhan
School of Artificial Intelligence, Nanjing University, China
National Key Laboratory for Novel Software Technology, Nanjing University, China
{hanlu, chenxy, yehj, zhandc}@lamda.nju.edu.cn

Abstract

Multivariate time series forecasting plays a crucial role in various fields such as finance, traffic management, energy, and healthcare. Recent studies have highlighted the advantages of channel independence to resist distribution drift but neglect channel correlations, limiting further enhancements. Several methods utilize mechanisms like attention or mixer to address this by capturing channel correlations, but they either introduce excessive complexity or rely too heavily on the correlation to achieve satisfactory results under distribution drifts, particularly with a large number of channels. Addressing this gap, this paper presents an efficient MLP-based model, the Series-cOre Fused Time Series forecaster (SOFTS), which incorporates a novel STar Aggregate-Redistribute (STAR) module. Unlike traditional approaches that manage channel interactions through distributed structures, *e.g.*, attention, STAR employs a centralized strategy to improve efficiency and reduce reliance on the quality of each channel. It aggregates all series to form a global core representation, which is then dispatched and fused with individual series representations to facilitate channel interactions effectively. SOFTS achieves superior performance over existing state-of-the-art methods with only linear complexity. The broad applicability of the STAR module across different forecasting models is also demonstrated empirically. We have made our code publicly available at https://github.com/Secilia-Cxy/SOFTS.

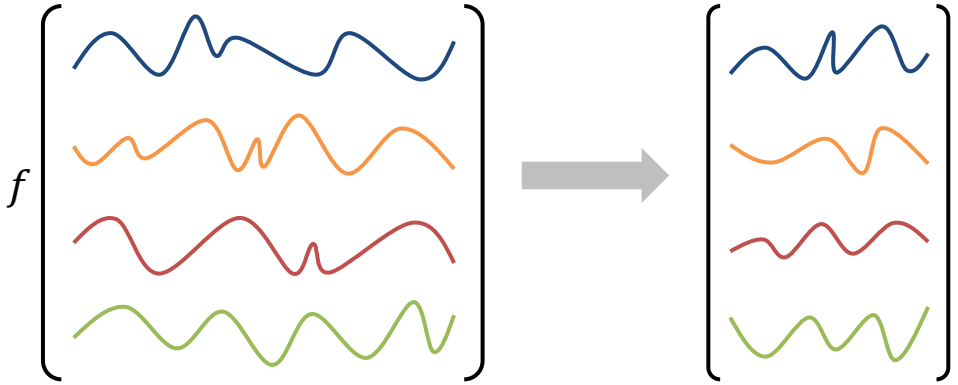
Motivation

Advanced Fundamental DL models: SOFTS

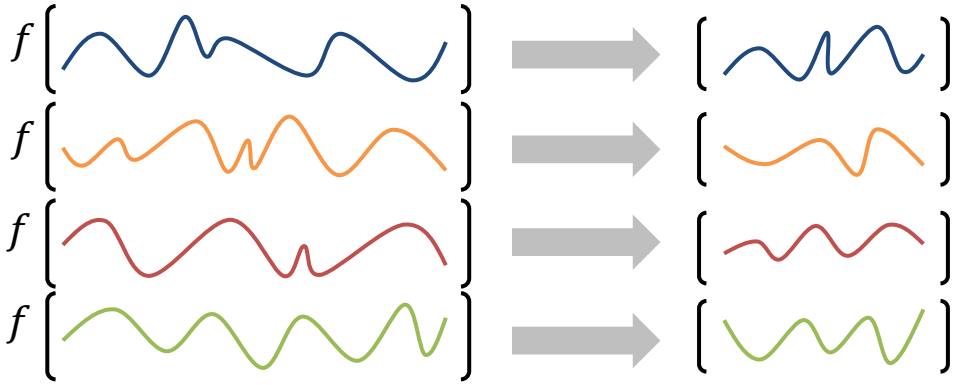
❖ **기존 시계열 예측 모델들은 어떤 딜레마를 가지고 있을까?**

- CI 전략은 분포 변화와 노이즈에 강하지만, **채널 간 상관관계를 활용하기 어려움**
- CD는 채널 간 상호작용을 반영하지만, **복잡도가 커지고 비정상 채널에 민감함**

Channel Independent



Channel Dependent

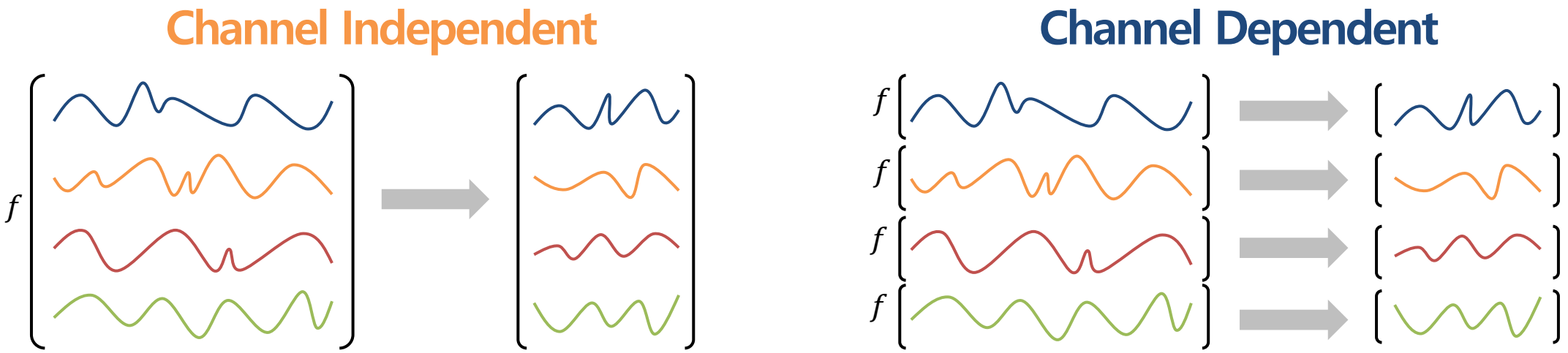


Motivation

Advanced Fundamental DL models: SOFTS

❖ **기존 시계열 예측 모델들은 어떤 딜레마를 가지고 있을까?**

- CI 전략은 분포 변화와 노이즈에 강하지만, **채널 간 상관관계를 활용하기 어려움**
- CD는 채널 간 상호작용을 반영하지만, **복잡도가 커지고 비정상 채널에 민감함**



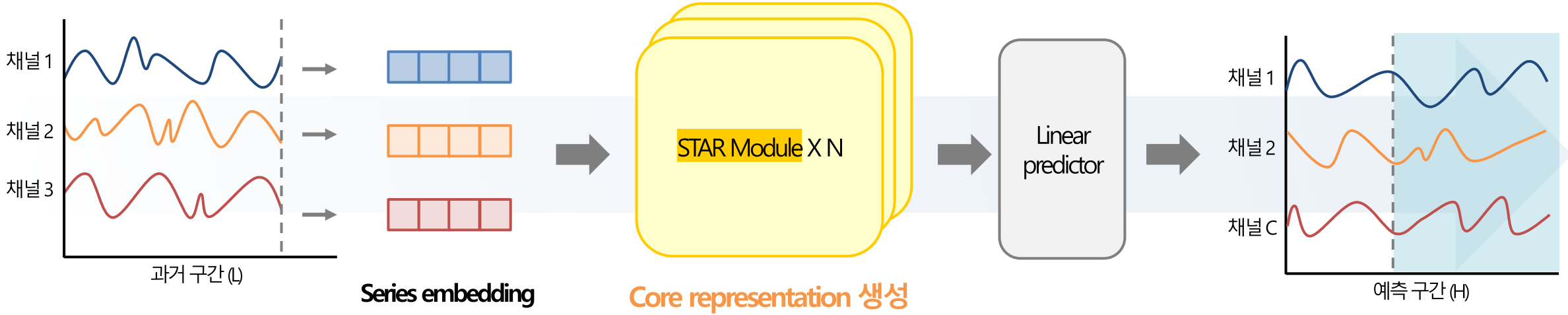
🧐 CI의 안정성과 CD의 채널 상호작용의 장점을 동시에 가져갈 수 없을까?

Method

Advanced Fundamental DL models: SOFTS

❖ SOFTS 아키텍처는 어디에서 CI와 CD를 결합할까?

- 각 채널별 시계열 데이터를 독립적으로 임베딩한 후, STAR 모듈 내 중앙 서버 역할을 하는 core representation 생성
- core representation을 개별 채널 표현과 결합 및 융합(Fusion)함으로써 채널 간의 간접적인 정보 교환 및 상호작용 수행

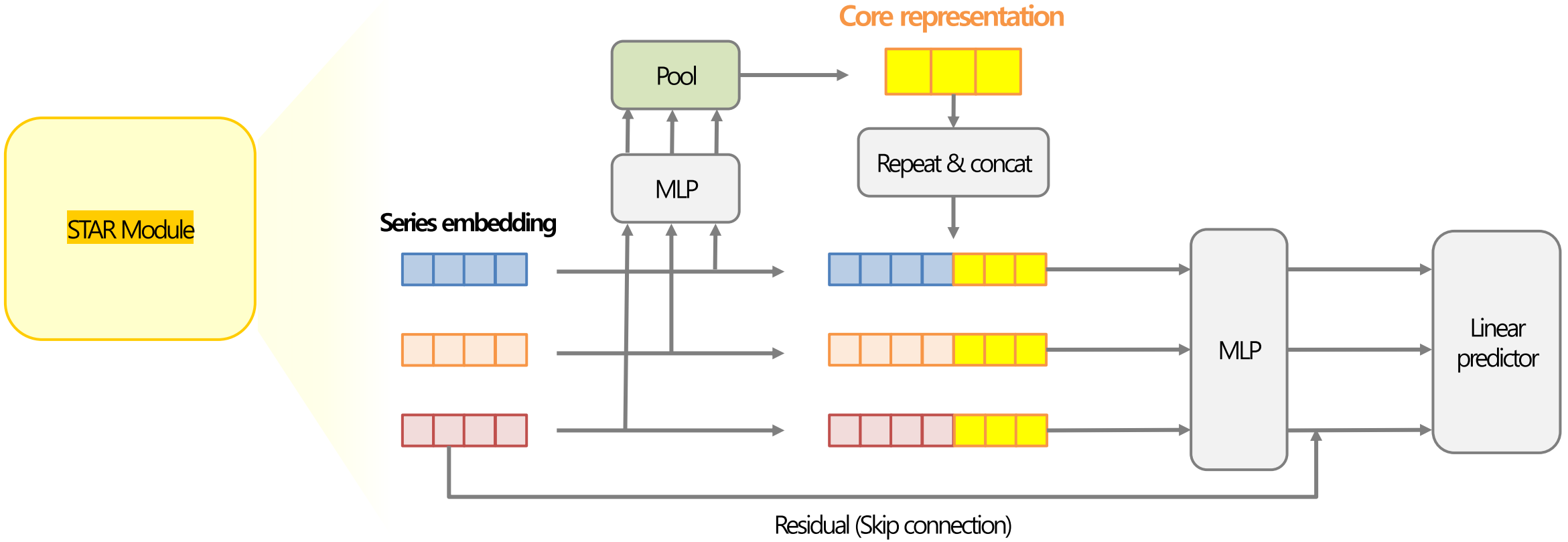


Method

Advanced Fundamental DL models: SOFTS

❖ SOFTS 아키텍처는 어디에서 CI와 CD를 결합할까?

- 각 채널별 시계열 데이터를 독립적으로 임베딩한 후, STAR 모듈 내 중앙 서버 역할을 하는 core representation 생성
- core representation을 개별 채널 표현과 결합 및 융합(Fusion)함으로써 채널 간의 간접적인 정보 교환 및 상호작용 수행



Experiment

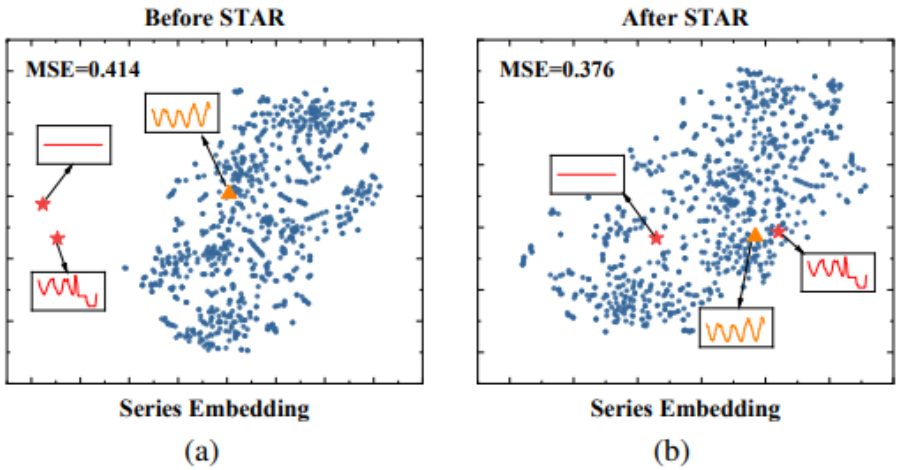
Advanced Fundamental DL models: SOFTS

❖ **SOFTS는 기존 방법론 대비 좋은 성능을 보여주었을까?**

- 12개의 벤치마크 데이터셋에서 기존 CI/CD 계열 모델 대비 좋은 성능
- STAR 적용 이후 abnormal channel이 정상 cluster 쪽으로 보정되며 예측 오차 감소

Table 2: Multivariate forecasting results with horizon $H \in \{12, 24, 48, 96\}$ for PEMS and $H \in \{96, 192, 336, 720\}$ for others and fixed lookback window length $L = 96$. Results are averaged from all prediction horizons. Full results are listed in Table 6.

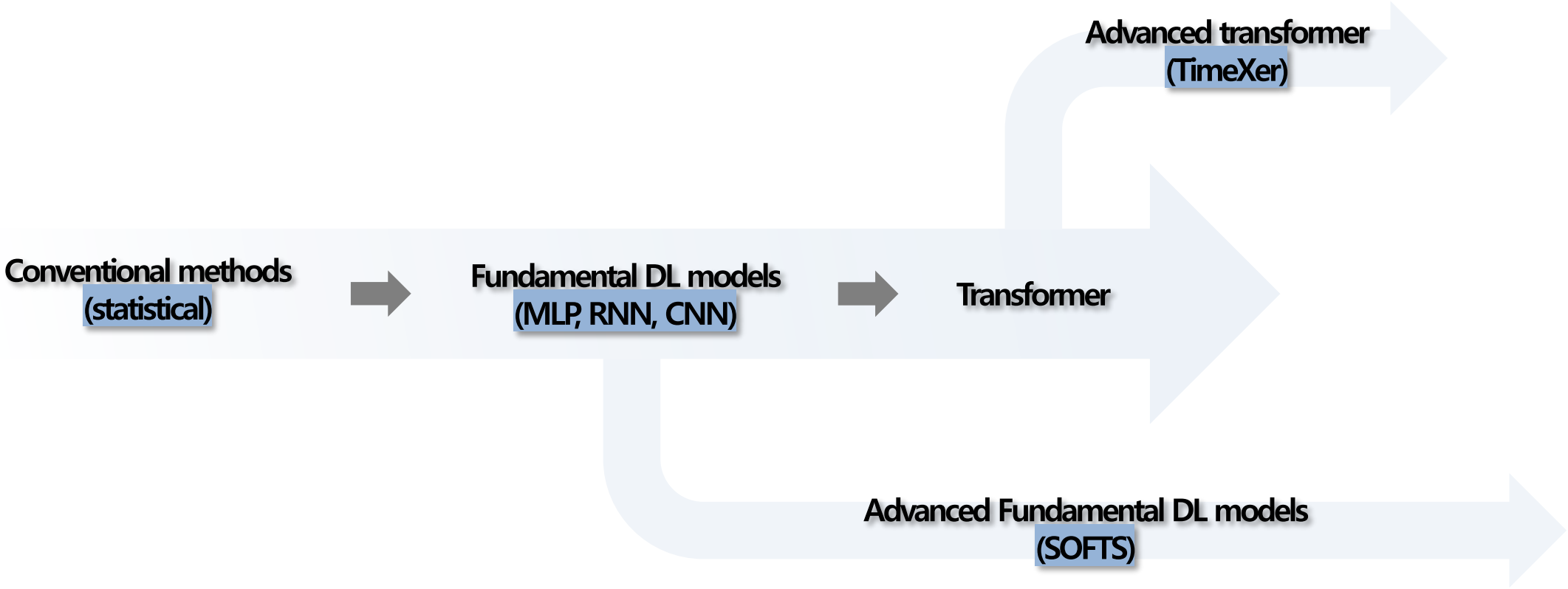
Models	SOFTS (ours)		iTransformer		PatchTST		TSMixer		Crossformer		TiDE		TimesNet		DLinear		SCINet		FEDformer		Stationary	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	**0.174**	**0.264**	0.178	0.270	0.189	0.276	0.186	0.287	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300	0.268	0.365	0.214	0.327	0.193	0.296
Traffic	**0.409**	**0.267**	0.428	0.282	0.454	0.286	0.522	0.357	0.550	0.304	0.760	0.473	0.620	0.336	0.625	0.383	0.804	0.509	0.610	0.376	0.624	0.340
Weather	**0.255**	**0.278**	0.258	**0.278**	0.256	0.279	0.256	0.279	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317	0.292	0.363	0.309	0.360	0.288	0.314
Solar-Energy	**0.229**	**0.256**	0.233	0.262	0.236	0.266	0.260	0.297	0.641	0.639	0.347	0.417	0.301	0.319	0.330	0.401	0.282	0.375	0.291	0.381	0.261	0.381
ETTm1	**0.393**	**0.403**	0.407	0.410	0.396	0.406	0.398	0.407	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407	0.485	0.481	0.448	0.452	0.481	0.456
ETTm2	**0.287**	**0.330**	0.288	0.332	**0.287**	**0.330**	0.289	0.333	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401	0.571	0.537	0.305	0.349	0.306	0.347
ETTh1	0.449	**0.442**	0.454	0.447	0.453	0.446	0.463	0.452	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452	0.747	0.647	**0.440**	0.460	0.570	0.537
ETTh2	**0.373**	**0.400**	0.383	0.407	0.385	0.410	0.401	0.417	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515	0.954	0.723	0.437	0.449	0.526	0.516
PEMS03	**0.104**	**0.210**	0.113	0.221	0.137	0.240	0.119	0.233	0.169	0.281	0.326	0.419	0.147	0.248	0.278	0.375	0.114	0.224	0.213	0.327	0.147	0.249
PEMS04	0.102	0.208	0.111	0.221	0.145	0.249	0.103	0.215	0.209	0.314	0.353	0.437	0.129	0.241	0.295	0.388	**0.092**	**0.202**	0.231	0.337	0.127	0.240
PEMS07	**0.087**	**0.184**	0.101	0.204	0.144	0.233	0.112	0.217	0.235	0.315	0.380	0.440	0.124	0.225	0.329	0.395	0.119	0.234	0.165	0.283	0.127	0.230
PEMS08	**0.138**	**0.219**	0.150	0.226	0.200	0.275	0.165	0.261	0.268	0.307	0.441	0.464	0.193	0.271	0.379	0.416	0.158	0.244	0.286	0.358	0.201	0.276



Introduction

Evolution of Time Series Forecasting Models

❖ 시계열 예측 모델은 어떻게 발전해왔을까?

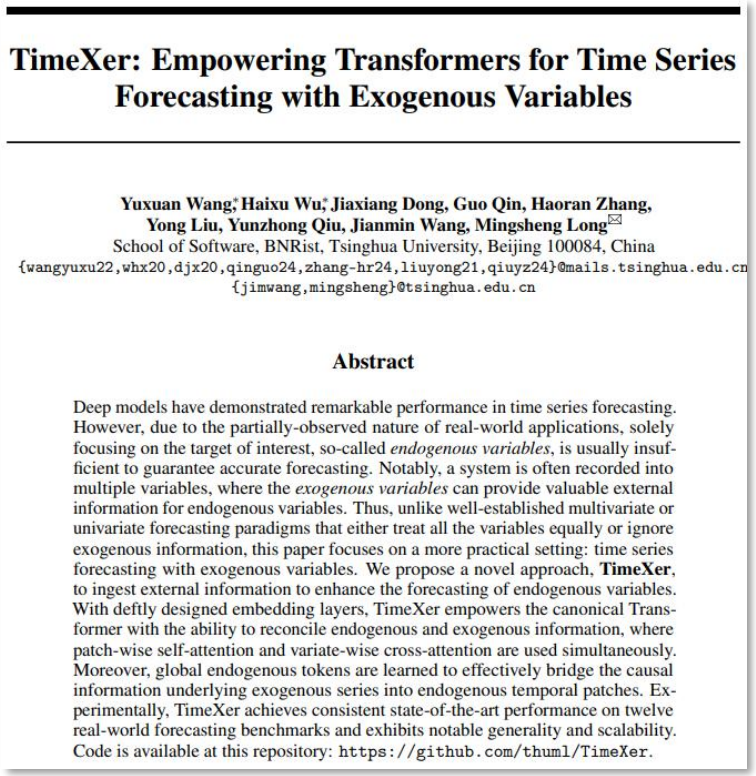


Related Works

Advanced transformer: TimeXer

❖ **TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables** (Wang et al., NeurIPS 2024)

- 예측 대상 변수는 endogenous variable로 두고, 자기 자신의 과거 흐름은 patch-level self-attention으로 학습
- 나머지 변수 또는 외부 정보는 exogenous variable로 보고, variate-level token으로 표현
- 예측 대상 변수를 중심으로 독립적으로 모델링하면서, cross-attention을 통해 외생 변수 정보를 선택적으로 활용



TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables

Yuxuan Wang,* Haixu Wu,* Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, Mingsheng Long✉
School of Software, BNRist, Tsinghua University, Beijing 100084, China
{wangyuxu22,whx20,djx20,qinguo24,zhang-hr24,liuyong21,qiuyz24}@mails.tsinghua.edu.cn
{jimwang,mingsheng}@tsinghua.edu.cn

Abstract

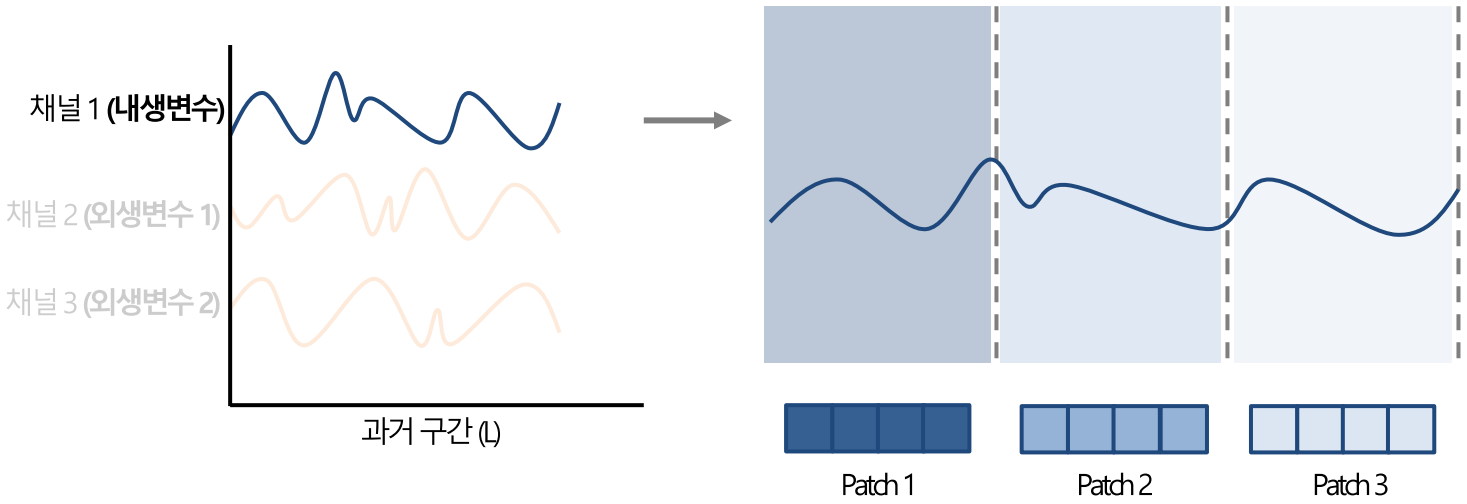
Deep models have demonstrated remarkable performance in time series forecasting. However, due to the partially-observed nature of real-world applications, solely focusing on the target of interest, so-called *endogenous variables*, is usually insufficient to guarantee accurate forecasting. Notably, a system is often recorded into multiple variables, where the *exogenous variables* can provide valuable external information for endogenous variables. Thus, unlike well-established multivariate or univariate forecasting paradigms that either treat all the variables equally or ignore exogenous information, this paper focuses on a more practical setting: time series forecasting with exogenous variables. We propose a novel approach, **TimeXer**, to ingest external information to enhance the forecasting of endogenous variables. With deftly designed embedding layers, TimeXer empowers the canonical Transformer with the ability to reconcile endogenous and exogenous information, where patch-wise self-attention and variate-wise cross-attention are used simultaneously. Moreover, global endogenous tokens are learned to effectively bridge the causal information underlying exogenous series into endogenous temporal patches. Experimentally, TimeXer achieves consistent state-of-the-art performance on twelve real-world forecasting benchmarks and exhibits notable generality and scalability. Code is available at this repository: `https://github.com/thuml/TimeXer`.

Method

Advanced transformer: TimeXer

❖ **TimeXer의 전체적인 아키텍처는 어떻게 구성되어 있을까?**

- 내생 변수(예측 대상 변수)의 시간 변화와 국소 패턴을 보존하기 위해 patch-level token 사용
- 각 patch의 시간적 순서를 유지하기 위해 positional encoding을 더해서 position-aware token 생성

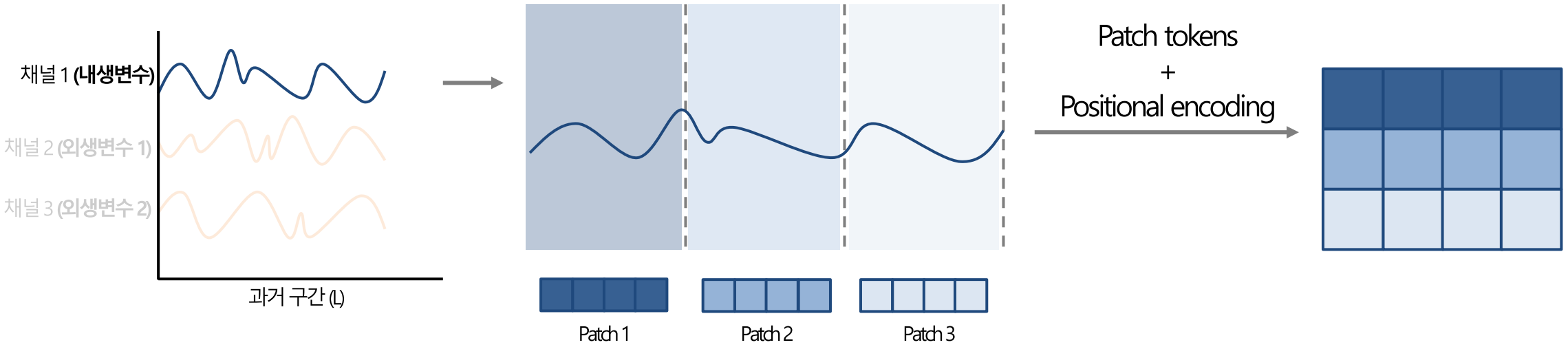


Method

Advanced transformer: TimeXer

❖ **TimeXer의 전체적인 아키텍처는 어떻게 구성되어 있을까?**

- 내생 변수(예측 대상 변수)의 시간 변화와 국소 패턴을 보존하기 위해 patch-level token 사용
- 각 patch의 시간적 순서를 유지하기 위해 positional encoding을 더해서 position-aware token 생성

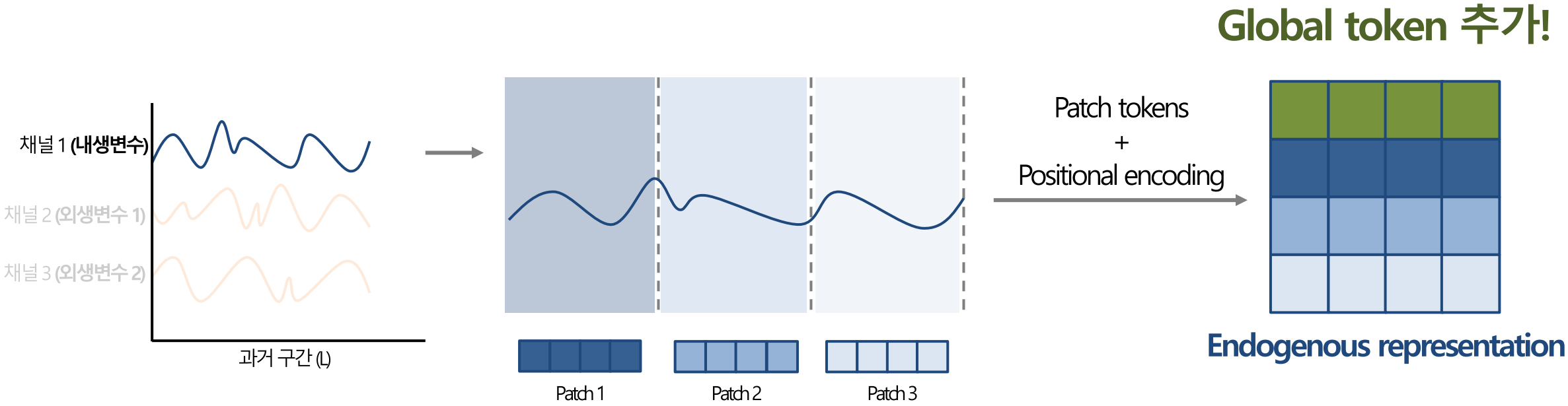


Method

Advanced transformer: TimeXer

❖ **TimeXer의 전체적인 아키텍처는 어떻게 구성되어 있을까?**

- **Global token**을 추가해 내생변수 시계열 전체를 대표하는 전역 표현 생성
- 이후 외생 변수의 정보를 받아오는 **bridge 역할을 수행**

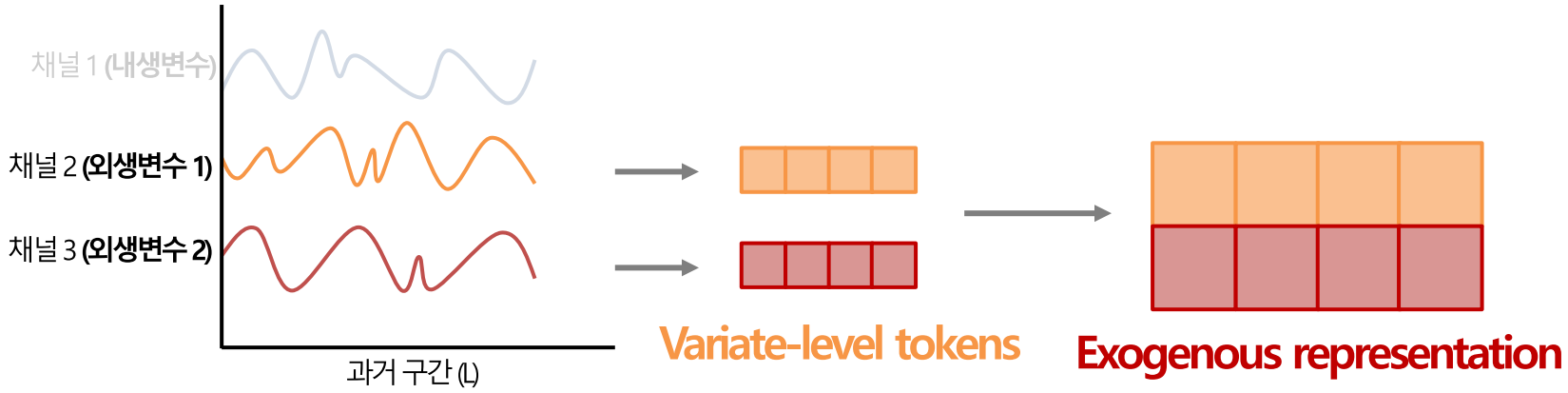


Method

Advanced transformer: TimeXer

❖ 외생변수는 어떻게 표현할까?

- 외생변수는 예측 대상이 아니라 target 예측을 돕는 정보를 제공하는 역할
- 각 변수가 target에 주는 영향력을 요약하는 것이 중요



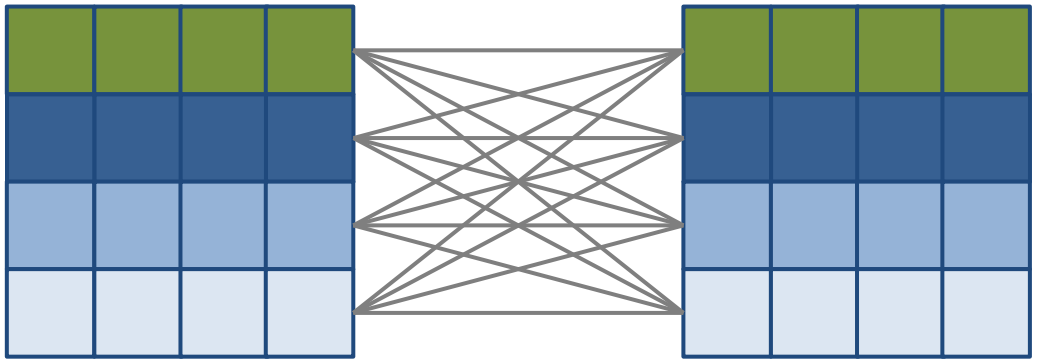
Method

Advanced transformer: TimeXer

❖ **내생변수와 외생변수의 정보는 어떻게 결합될까?**

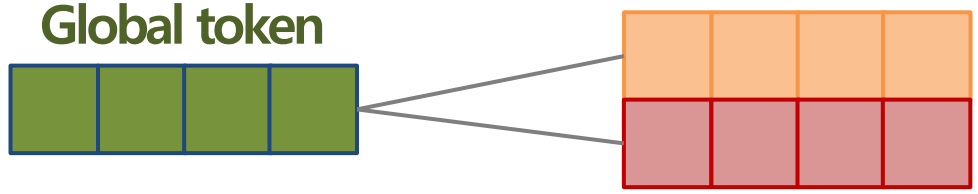
- Self-attention: patch token과 global token이 상호작용하며 target 내부의 시간 의존성 학습
- Cross-attention: global token이 외생변수 token에서 필요한 외부 정보를 선택적으로 수집

Self-attention



Endogenous representation Endogenous representation

Cross-attention



Experiment

Advanced transformer: TimeXer

❖ **TimeXer은 exogenous 정보를 실제로 잘 활용했을까?**

- 7개의 벤치마크 데이터셋에서 기존 CI/CD 계열 모델 대비 좋은 성능
- 외생변수를 활용해 기존 모델 대비 경쟁력 있는 성능을 보여줌

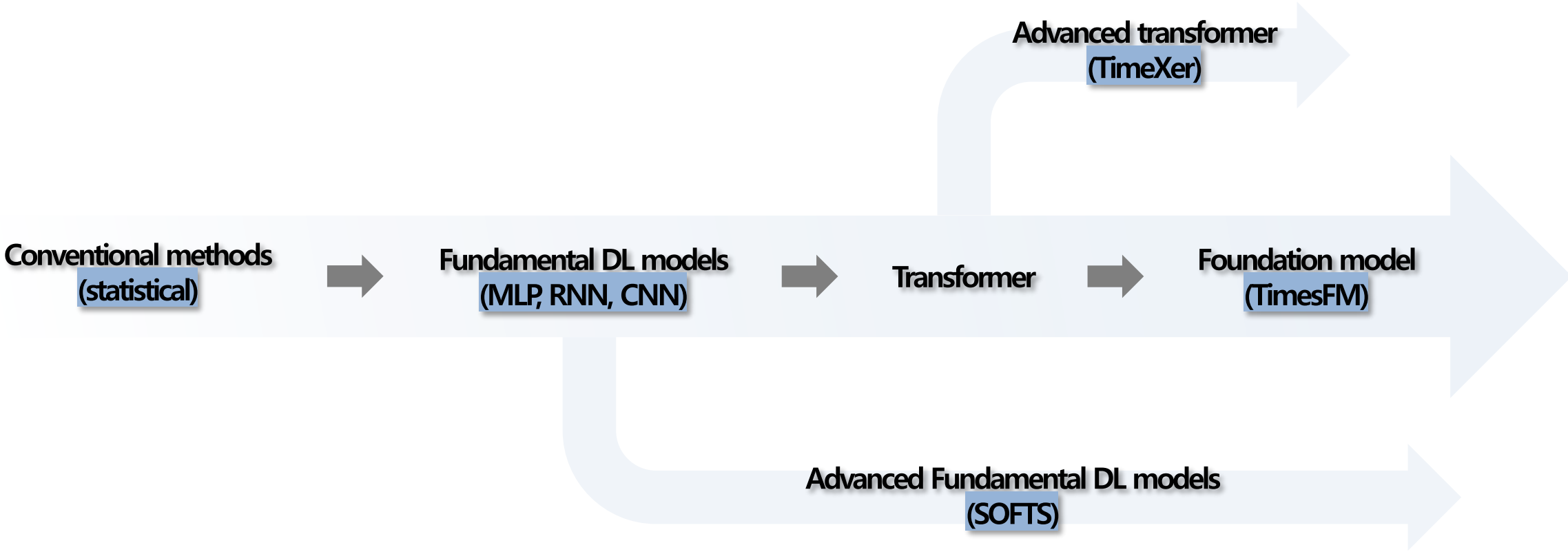
Table 3: Multivariate forecasting results. We compare extensive competitive models under different prediction lengths following the setting of iTransformer [23]. The look-back length L is set to 96 for all baselines. Results are averaged from all prediction lengths $S = \{96, 192, 336, 720\}$.

Model	**TimeXer**		iTransformer		RLinear		PatchTST		Crossformer		TiDE		TimesNet		DLinear		SCINet		Autoformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	**0.171**	**0.270**	0.178	0.270	0.219	0.298	0.205	0.290	0.244	0.334	0.251	0.244	0.192	0.295	0.212	0.300	0.268	0.365	0.227	0.338
Weather	**0.241**	**0.271**	0.258	0.278	0.272	0.291	0.259	0.281	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317	0.292	0.363	0.338	0.382
ETTh1	**0.437**	**0.437**	0.454	0.447	0.446	0.434	0.469	0.454	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452	0.747	0.647	0.496	0.487
ETTh2	**0.367**	**0.396**	0.383	0.407	0.374	0.398	0.387	0.407	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515	0.954	0.723	0.450	0.459
ETTm1	**0.382**	**0.397**	0.407	0.410	0.414	0.407	0.387	0.400	0.512	0.496	0.419	0.419	0.400	0.406	0.403	0.407	0.485	0.481	0.588	0.517
ETTm2	**0.274**	**0.322**	0.288	0.332	0.286	0.327	0.281	0.326	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401	0.571	0.537	0.327	0.371
Traffic	0.466	0.287	**0.428**	**0.282**	0.626	0.378	0.481	0.304	0.550	0.304	0.760	0.473	0.620	0.336	0.625	0.383	0.804	0.509	0.628	0.379

Introduction

Evolution of Time Series Forecasting Models

❖ 시계열 예측 모델은 어떻게 발전해왔을까?

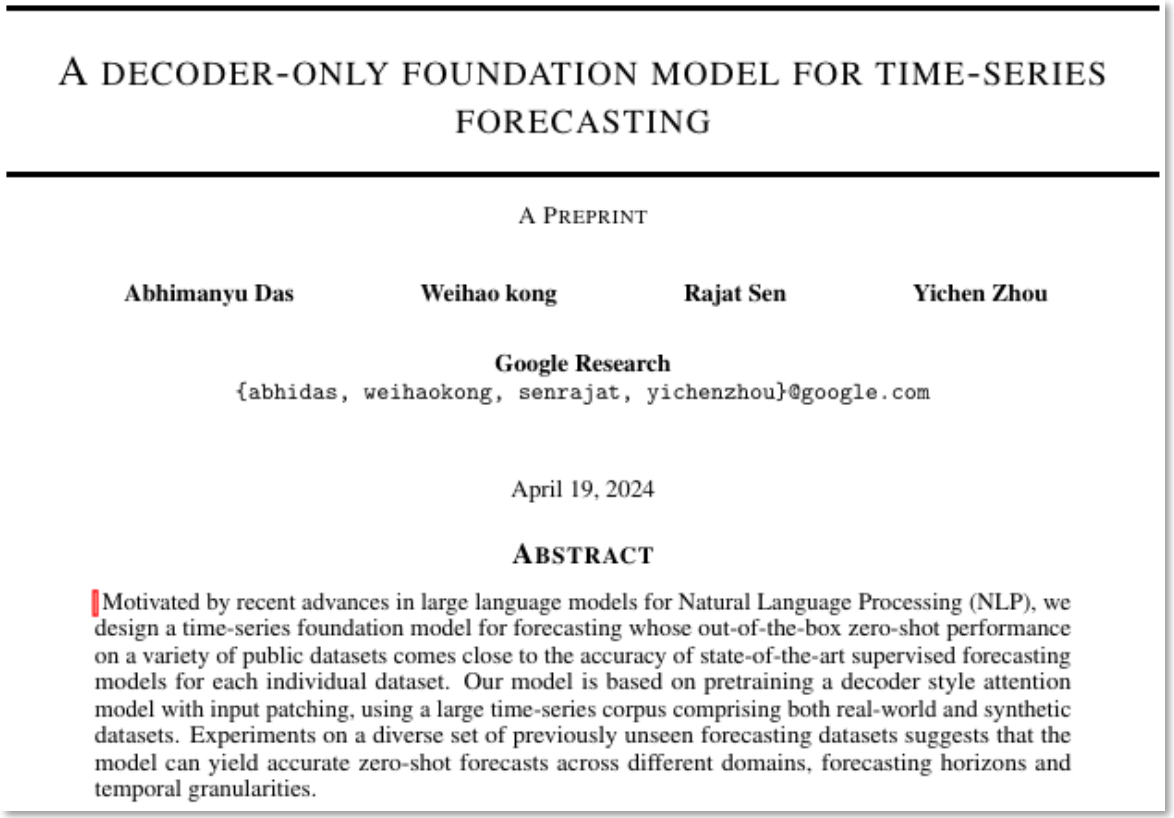


Related Works

Foundation model

❖ **A Decoder-only Foundation Model for Time-Series Forecasting** (Das et al., ICML 2024)

- 대규모 데이터를 얻기 위해 실제 데이터(예: 구글 트렌드, 위키미디어 페이지뷰)와 합성 데이터를 활용하여 다양한 학습 데이터 확보
- 다양한 컨텍스트 길이를 입력받고 forecast horizon을 처리할 수 있도록 patching을 이용한 디코더 어텐션 아키텍쳐를 설계



A DECODER-ONLY FOUNDATION MODEL FOR TIME-SERIES FORECASTING

A PREPRINT

Abhimanyu Das **Weihao kong** **Rajat Sen** **Yichen Zhou**

Google Research
{abhidas, weihaokong, senrajat, yichenzhou}@google.com

April 19, 2024

ABSTRACT

Motivated by recent advances in large language models for Natural Language Processing (NLP), we design a time-series foundation model for forecasting whose out-of-the-box zero-shot performance on a variety of public datasets comes close to the accuracy of state-of-the-art supervised forecasting models for each individual dataset. Our model is based on pretraining a decoder style attention model with input patching, using a large time-series corpus comprising both real-world and synthetic datasets. Experiments on a diverse set of previously unseen forecasting datasets suggests that the model can yield accurate zero-shot forecasts across different domains, forecasting horizons and temporal granularities.

Background

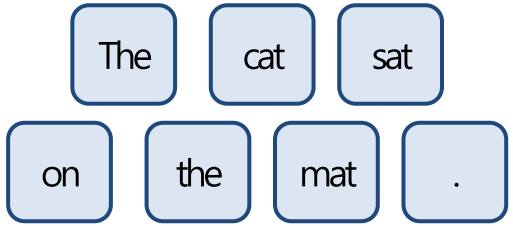
Foundation model

❖ **왜 시계열 분야에서는 범용 모델이 어려웠을까?**

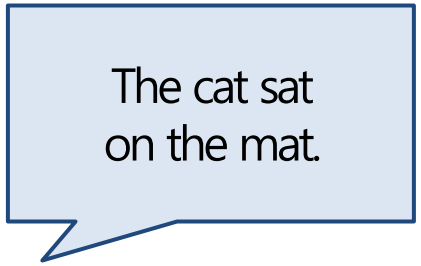
- 시계열에는 자연어처럼 명확한 vocabulary나 grammar가 없음
- context length, forecast horizon, temporal granularity가 데이터마다 다름
- LLM 수준의 대규모 공개 시계열 corpus 확보가 어려움

NLP

단어/토큰



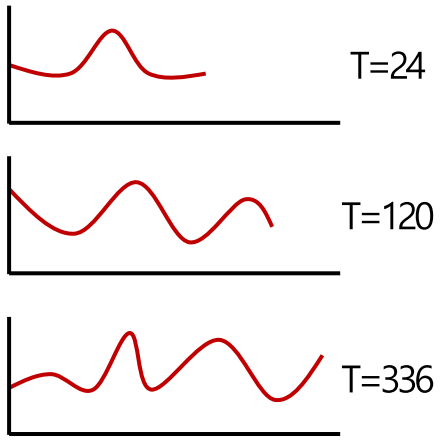
문장



대규모 텍스트 코퍼스

Time Series

가변 길이



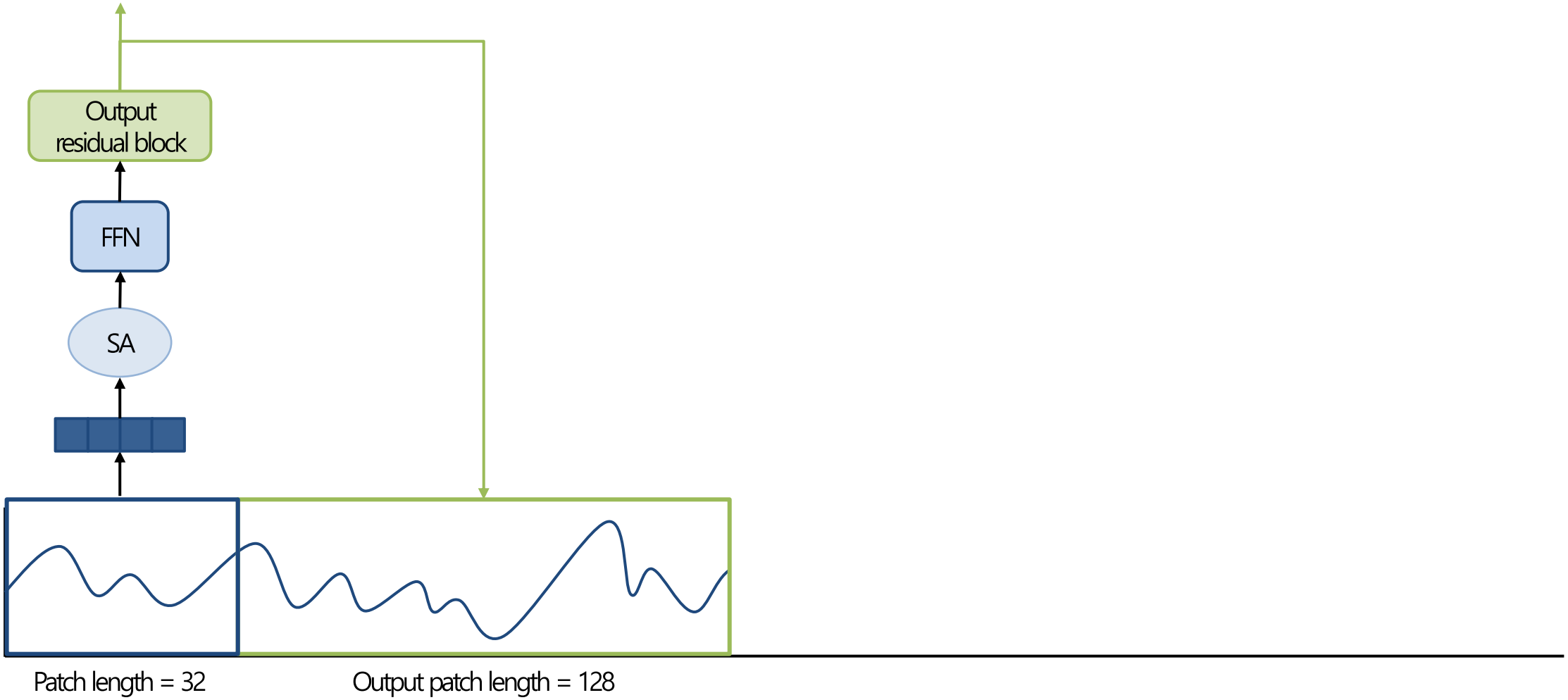
다양한 주기/빈도

제한된 대규모 데이터

Method

Foundation model

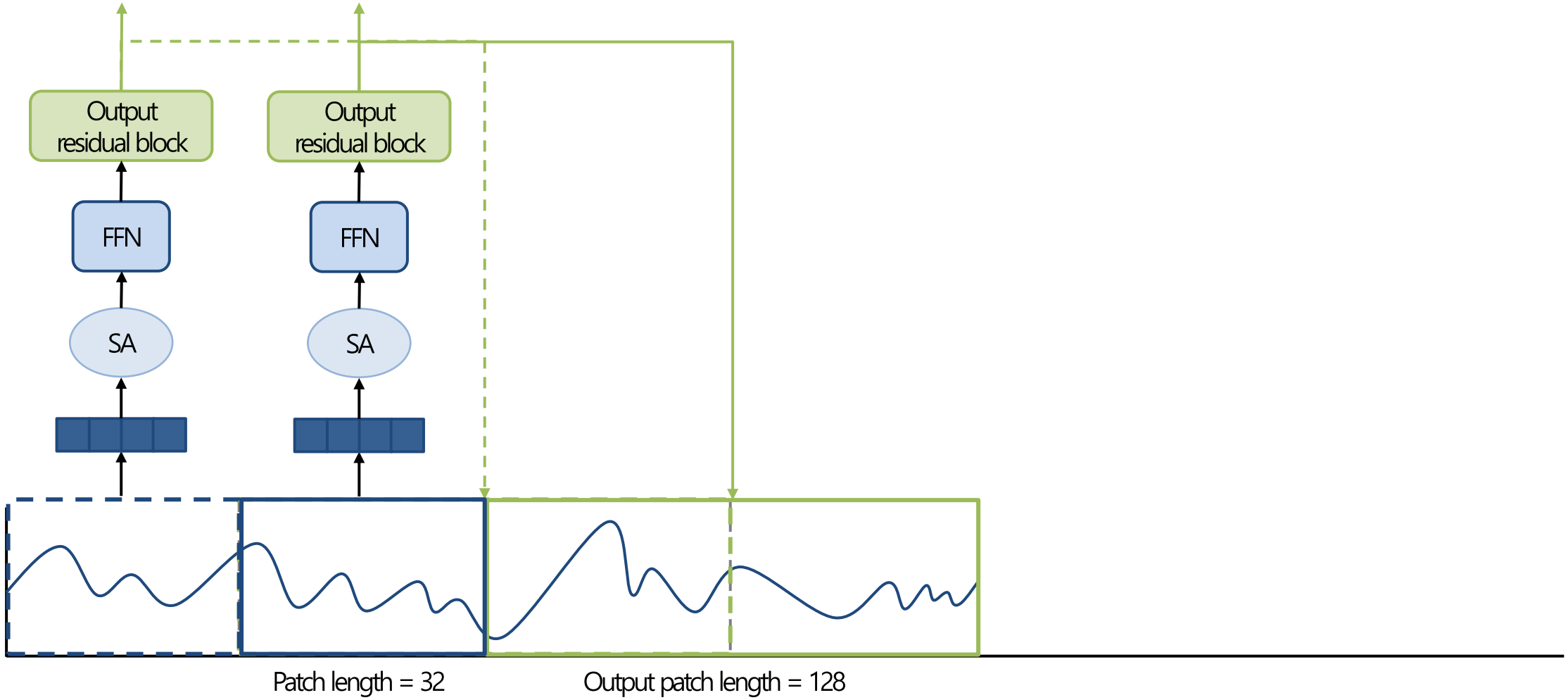
❖ **TimesFM은 어떤 구조를 가지고 있을까?**



Method

Foundation model

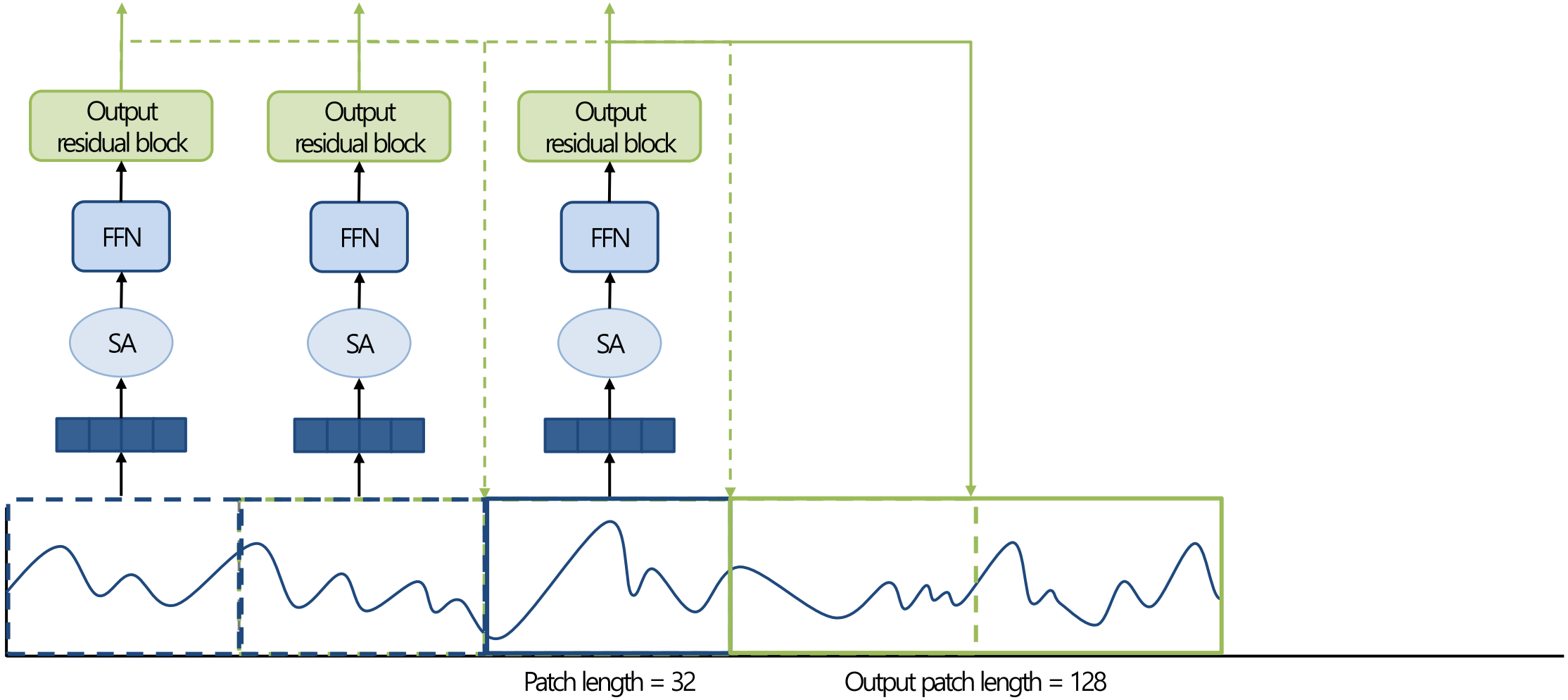
❖ **TimesFM은 어떤 구조를 가지고 있을까?**



Method

Foundation model

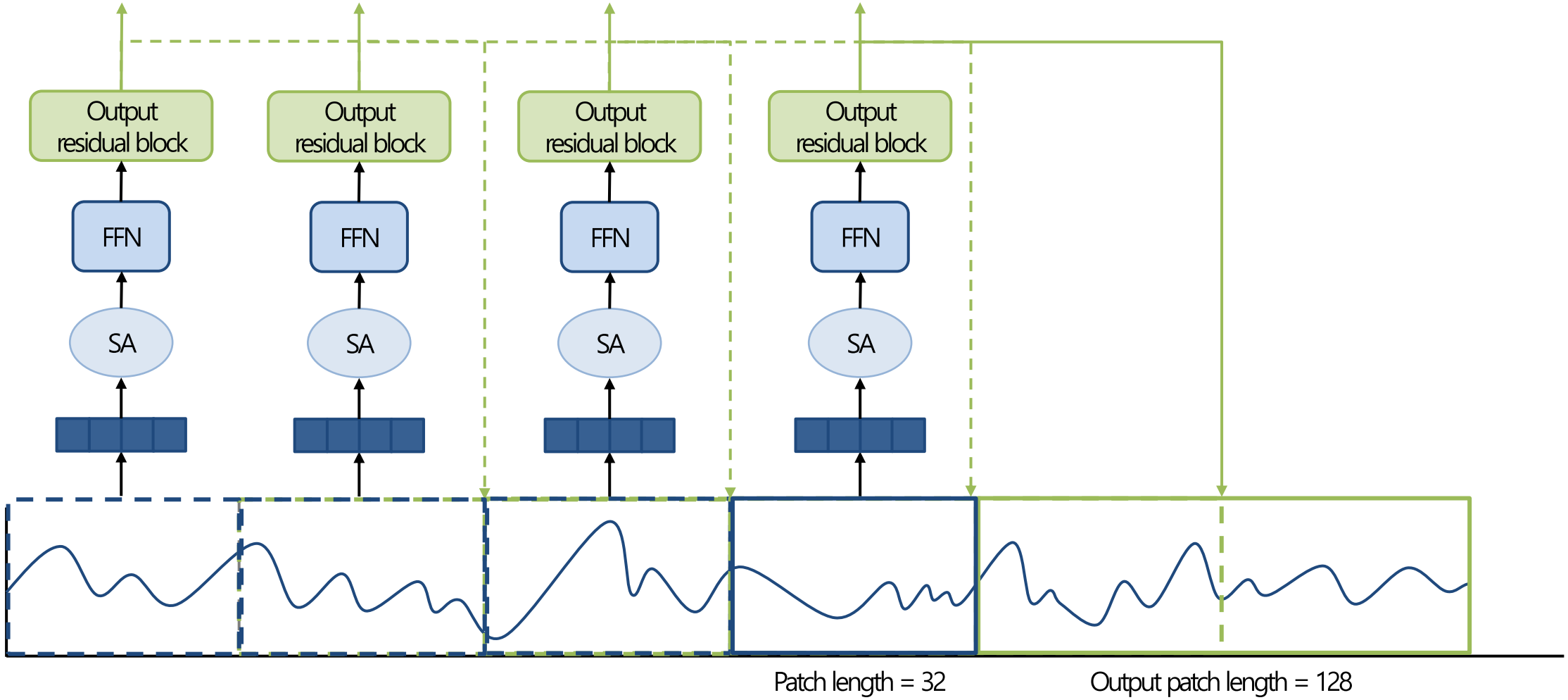
❖ **TimesFM은 어떤 구조를 가지고 있을까?**



Method

Foundation model

❖ **TimesFM은 어떤 구조를 가지고 있을까?**

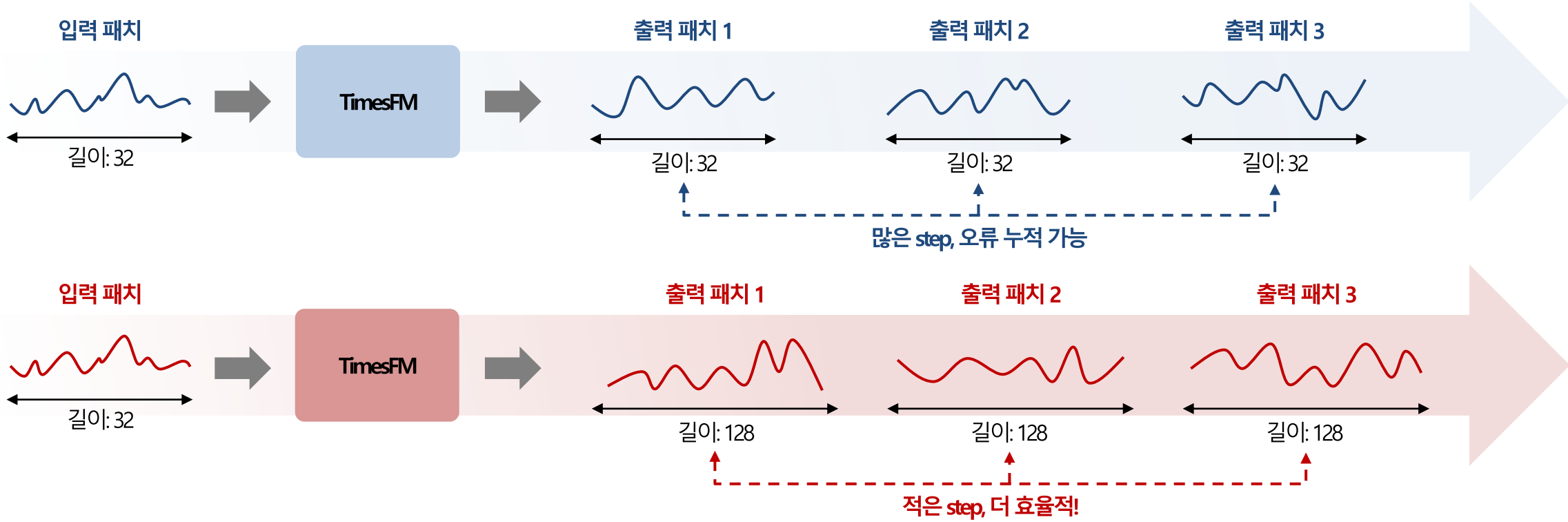


Method

Foundation model

❖ **입력 패치보다 긴 출력 패치를 사용하는 이유는 무엇일까?**

- 긴 horizon을 한 번에 더 큰 chunk로 예측해 autoregressive step 수를 줄임
- One-shot 예측과 token-by-token 예측 사이의 절충안
- Forecast horizon이 다양한 zero-shot 환경에 더 유연하게 대응이 가능함

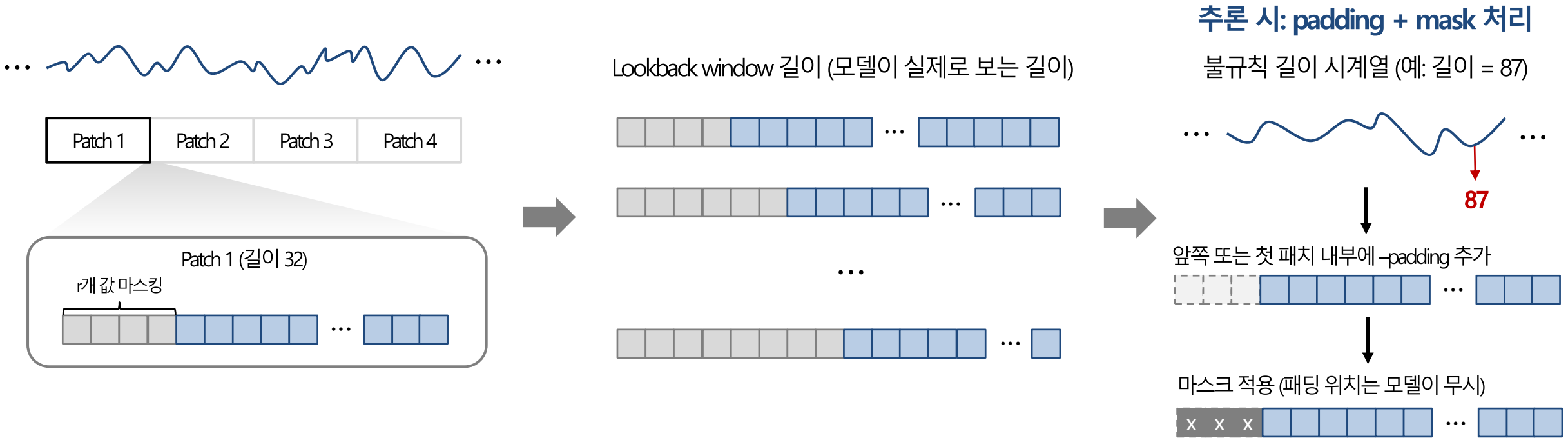


Method

Foundation model

❖ **모델이 모든 입력 길이에 적응할 수 있는 이유는 무엇일까?**

- Patch 단위 학습만 하면 특정 patch 배수 길이에만 익숙해질 수 있음
- Random masking으로 다양한 context length를 학습
- Padding과 mask로 길이가 맞지 않는 입력도 처리

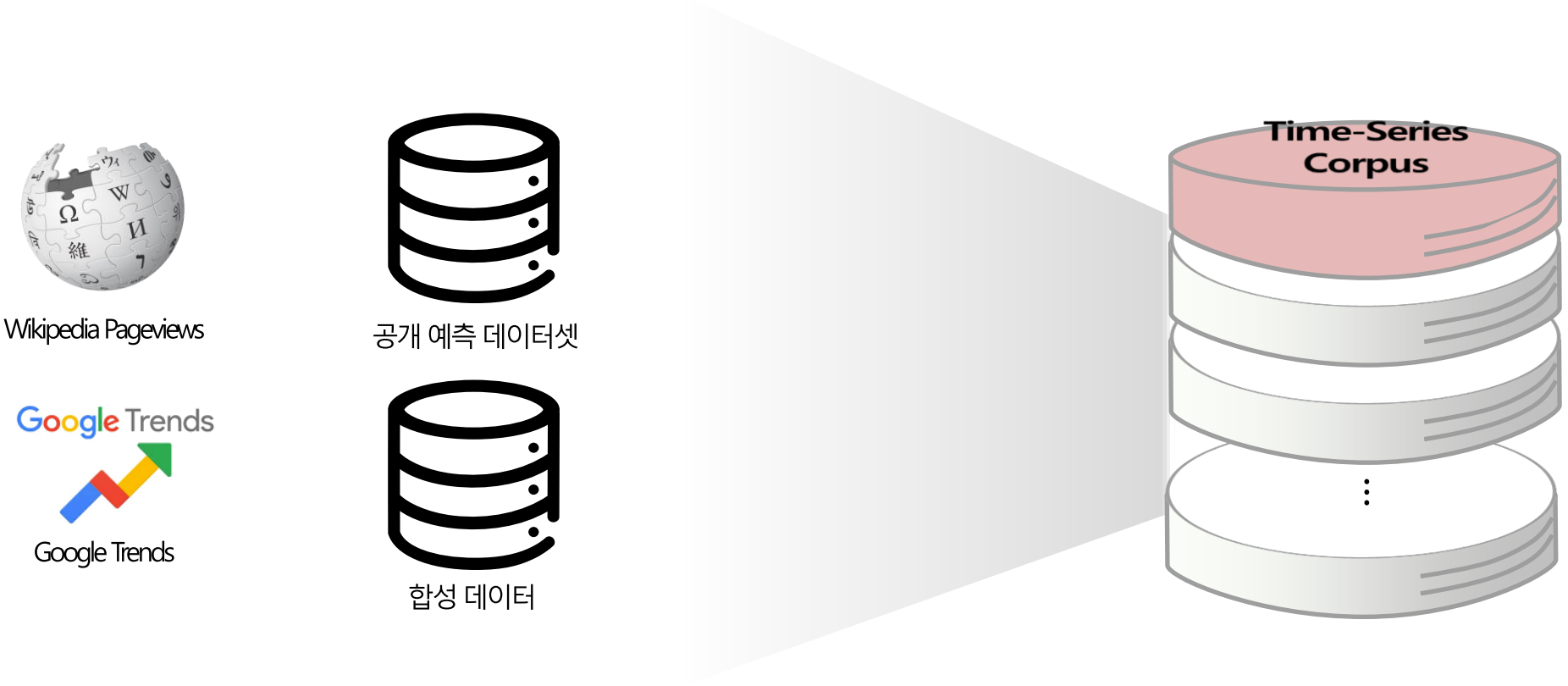


Method

Foundation model

❖ **1000억 개 이상의 timepoints는 어떻게 구성했을까?**

- Wiki Pageviews와 Google Trends로 대규모 real-world 시계열 확보
- 공개 forecasting dataset으로 domain 다양성 보완
- Synthetic data로 부족한 패턴과 granularity를 추가 학습

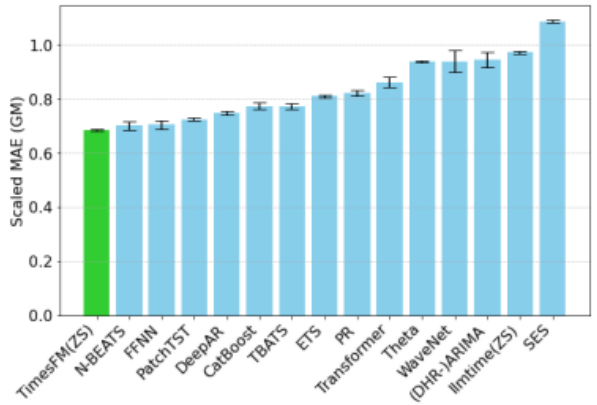


Experiment

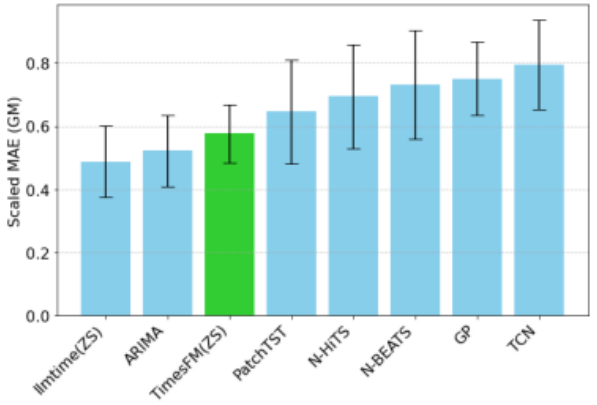
Foundation model

❖ 추가 학습 없이 좋은 예측 성능을 보였나?

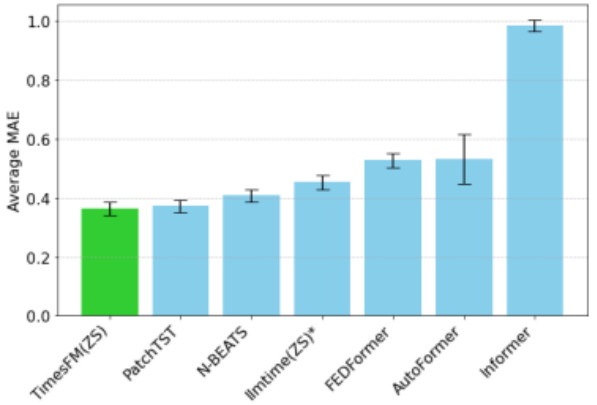
- 평가 데이터셋은 pretraining에서 제외된 unseen datasets
- TimesFM은 zero-shot setting에서도 supervised baseline과 경쟁
- Monash, Darts, ETT에서 범용 예측 성능 확인



(a) Monash Archive [GBW+21]



(b) Darts [HLP+22]



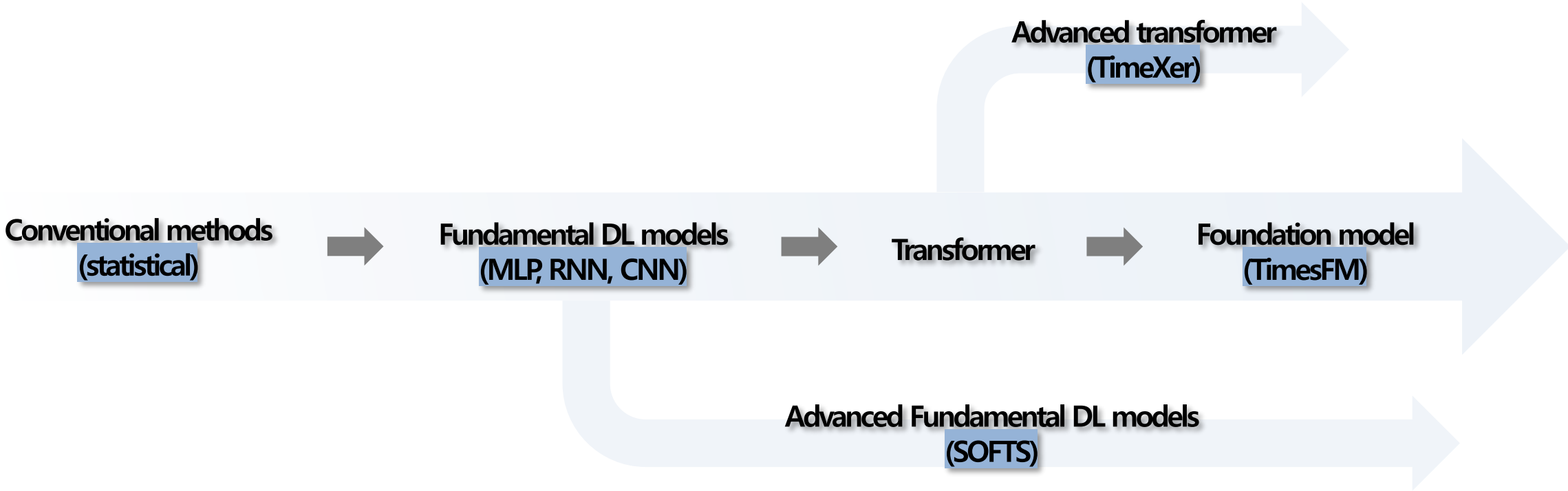
(c) ETT (Horizons 96 and 192) [ZZP+21]

Conclusion

Evolution of Time Series Forecasting Models

❖ **시계열 예측 분야는 어떻게 발전해왔을까?**

- 통계적 가정 기반 모델에서 데이터 기반 표현 학습으로 이동
- Transformer, DLinear, patching, channel strategy로 확장
- CI의 안정성과 CD의 상호작용을 함께 활용하는 방향으로 발전
- 데이터셋별 학습에서 대규모 사전학습 기반 zero-shot forecasting으로 이동



고맙습니다