Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting

https://arxiv.org/abs/2106.13008

0. Introduction

- 장기 시계열 예측은 에너지, 트래픽, 경제, 날씨 등 다양한 분야에서 필수적이지만, 기존 방법은 장기 패턴 학습에 어려움이 있음.
- Transformer 기반 모델은 장기 종속성 파악에 강점이 있으나 계산 복잡도와 패턴 얽힘 문제 존재.
- 본 논문은 시계열 분해와 Auto-Correlation 메커니즘을 결합한 Autoformer를 제안함.

1. Overview

- 이 논문은 시계열 장기 예측 문제를 다룸
- 기존 Transformer 기반 시계열 모델은 복잡한 시간 패턴과 장기 의존성 학습에 한계가 있었음
- 이를 해결하기 위해 Autoformer라는 새로운 구조를 제안함
  - 시계열 데이터를 점진적으로 분해하여 복잡한 패턴을 단순한 구성 요소(추세 + 잔차)로 분리
  - Auto-Correlation 메커니즘을 통해 시계열 내 주기성과 반복성을 기반으로 종속성 탐색 및 정보 집계 수행
- 기존 self-attention 기반 방식과 달리 연산 복잡도는 O(LlogL)로 더 효율적이며, long sequence 처리 시 정보 병목 문제를 완화함
- 다양한 실제 시계열 데이터셋에서 기존 모델 대비 일관된 SOTA 성능을 기록함

2. Challenges

- 장기 시계열은 복잡한 주기와 잡음이 공존하며, 패턴이 얽혀있어 예측이 어려움.
- self-attention은 계산 비용이 높고, sparse attention은 포인트 단위 연결로 정보 손실 발생.
- 시계열 내 시리즈 수준의 종속성을 효과적으로 포착하는 기법 필요.

3. Method

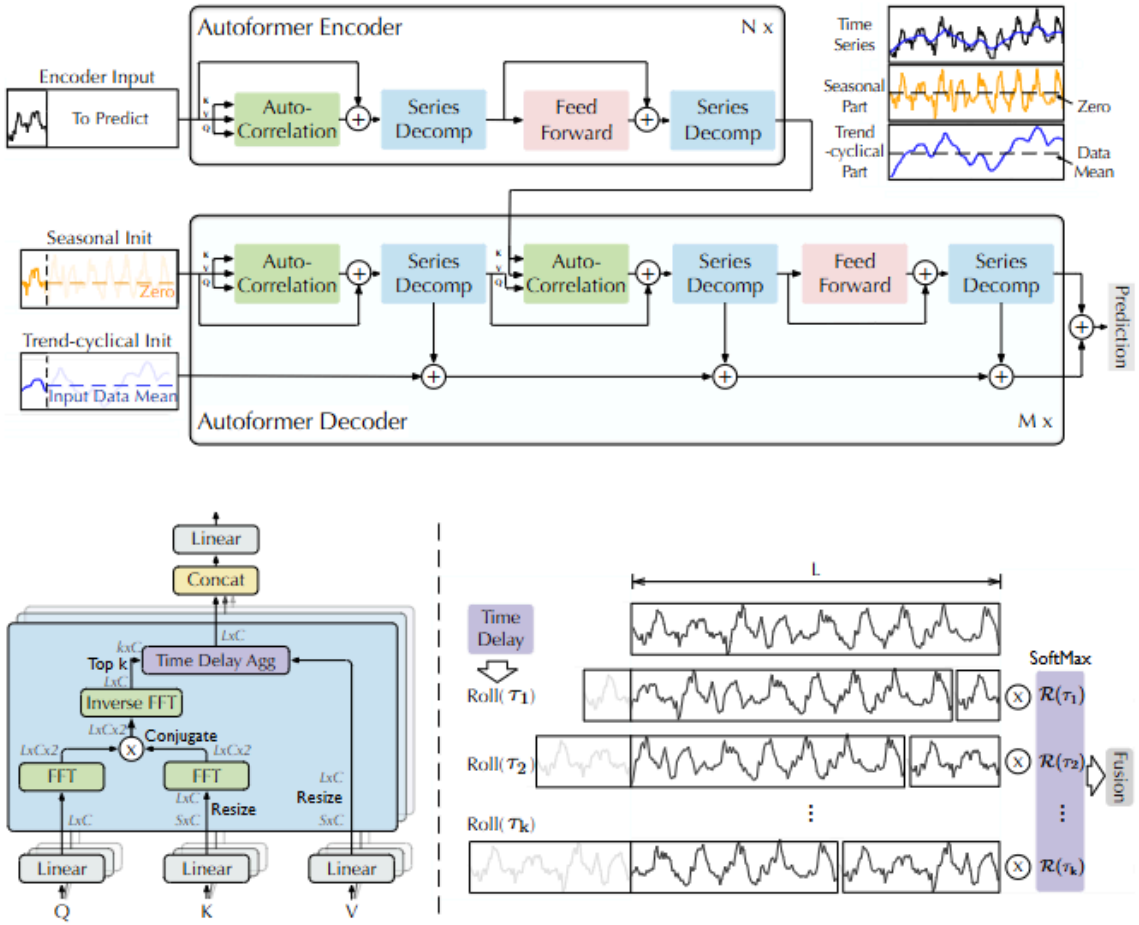


Figure 2: Auto-Correlation (left) and Time Delay Aggregation (right). We utilize the Fast Fourier Transform to calculate the autocorrelation $\mathcal{R}(\tau)$, which reflects the time-delay similarities. Then the similar sub-processes are rolled to the same index based on selected delay τ and aggregated by $\mathcal{R}(\tau)$.

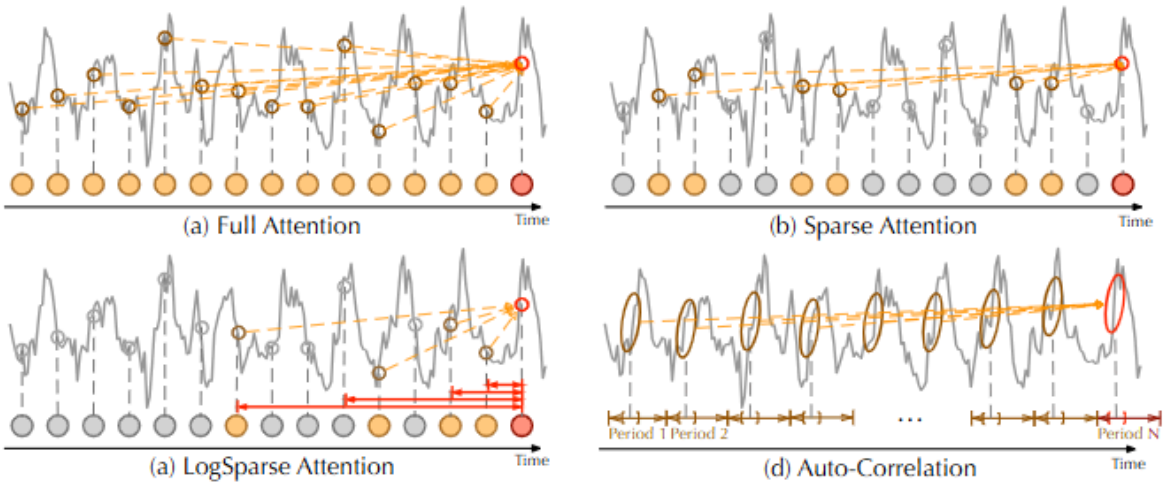


Figure 3: Auto-Correlation vs. self-attention family. Full Attention [41] (a) adapts the fully connection among all time points. Sparse Attention [23, 48] (b) selects points based on the proposed similarity metrics. LogSparse Attention [26] (c) chooses points following the exponentially increasing intervals. Auto-Correlation (d) focuses on the connections of sub-series among underlying periods.

- Series Decomposition Block
  - 입력 시계열을 trend(추세)와 seasonal(계절성) 성분으로 분해하여 처리.
  - trend는 moving average로 추출하고, seasonal은 나머지 구성요소로 간주.
  - 이 과정을 통해 복잡한 시계열을 단순 구성 요소로 분해하여 안정적인 학습 유도.
- Auto-Correlation Mechanism
  - 기존 self-attention 대신 사용되는 핵심 모듈로, 시계열 내 주기성을 기반으로 유사한 시점 간 종속성을 포착.
  - 두 단계로 구성:
    1. 상관성 탐색(Search Phase): 주기적인 시점 간 유사도를 기반으로 관련 시점을 선택.
    2. 정보 집계(Aggregation Phase): 유사한 시점들의 정보를 통합하여 표현 강화.
  - 계산 복잡도는 O(L log L)로, 기존 self-attention의 $O(L^2)$보다 효율적.
- Encoder-Decoder 구조
  - Encoder: 입력 시계열을 분해 후 seasonal, trend 성분 각각에 대해 Auto-Correlation 적용.
  - Decoder: 이전 time step에서 예측된 trend는 누적되고, seasonal은 반복 예측.
  - 예측된 trend + seasonal 성분을 합산하여 최종 예측 결과 생성.

- Embedding 없이 Raw Input 사용
    - 시계열의 연속성과 스케일 유지가 중요하기 때문에, 일반적인 position embedding은 사용하지 않고, 입력값을 그대로 사용.

4. Experiments

5. Results

- 5가지 주요 시계열 예측 응용 분야를 포함하는, 총 6가지 benchmark에서 평가 (에너지, 교통, 경제, 날씨, 질병 등)

Models		**Autoformer**		Informer[48]		LogTrans[26]		Reformer[23]		LSTNet[25]		LSTM[17]		TCN[4]	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT*	96	**0.255**	**0.339**	0.365	0.453	0.768	0.642	0.658	0.619	3.142	1.365	2.041	1.073	3.041	1.330
	192	**0.281**	**0.340**	0.533	0.563	0.989	0.757	1.078	0.827	3.154	1.369	2.249	1.112	3.072	1.339
	336	**0.339**	**0.372**	1.363	0.887	1.334	0.872	1.549	0.972	3.160	1.369	2.568	1.238	3.105	1.348
	720	**0.422**	**0.419**	3.379	1.388	3.048	1.328	2.631	1.242	3.171	1.368	2.720	1.287	3.135	1.354
Electricity	96	**0.201**	**0.317**	0.274	0.368	0.258	0.357	0.312	0.402	0.680	0.645	0.375	0.437	0.985	0.813
	192	**0.222**	**0.334**	0.296	0.386	0.266	0.368	0.348	0.433	0.725	0.676	0.442	0.473	0.996	0.821
	336	**0.231**	**0.338**	0.300	0.394	0.280	0.380	0.350	0.433	0.828	0.727	0.439	0.473	1.000	0.824
	720	**0.254**	**0.361**	0.373	0.439	0.283	0.376	0.340	0.420	0.957	0.811	0.980	0.814	1.438	0.784
Exchange	96	**0.197**	**0.323**	0.847	0.752	0.968	0.812	1.065	0.829	1.551	1.058	1.453	1.049	3.004	1.432
	192	**0.300**	**0.369**	1.204	0.895	1.040	0.851	1.188	0.906	1.477	1.028	1.846	1.179	3.048	1.444
	336	**0.509**	**0.524**	1.672	1.036	1.659	1.081	1.357	0.976	1.507	1.031	2.136	1.231	3.113	1.459
	720	**1.447**	**0.941**	2.478	1.310	1.941	1.127	1.510	1.016	2.285	1.243	2.984	1.427	3.150	1.458
Traffic	96	**0.613**	**0.388**	0.719	0.391	0.684	0.384	0.732	0.423	1.107	0.685	0.843	0.453	1.438	0.784
	192	**0.616**	**0.382**	0.696	0.379	0.685	0.390	0.733	0.420	1.157	0.706	0.847	0.453	1.463	0.794
	336	**0.622**	**0.337**	0.777	0.420	0.733	0.408	0.742	0.420	1.216	0.730	0.853	0.455	1.479	0.799
	720	**0.660**	**0.408**	0.864	0.472	0.717	0.396	0.755	0.423	1.481	0.805	1.500	0.805	1.499	0.804
Weather	96	**0.266**	**0.336**	0.300	0.384	0.458	0.490	0.689	0.596	0.594	0.587	0.369	0.406	0.615	0.589
	192	**0.307**	**0.367**	0.598	0.544	0.658	0.589	0.752	0.638	0.560	0.565	0.416	0.435	0.629	0.600
	336	**0.359**	**0.395**	0.578	0.523	0.797	0.652	0.639	0.596	0.597	0.587	0.455	0.454	0.639	0.608
	720	**0.419**	**0.428**	1.059	0.741	0.869	0.675	1.130	0.792	0.618	0.599	0.535	0.520	0.639	0.610
ILI	24	**3.483**	**1.287**	5.764	1.677	4.480	1.444	4.400	1.382	6.026	1.770	5.914	1.734	6.624	1.830
	36	**3.103**	**1.148**	4.755	1.467	4.799	1.467	4.783	1.448	5.340	1.668	6.631	1.845	6.858	1.879
	48	**2.669**	**1.085**	4.763	1.469	4.800	1.468	4.832	1.465	6.080	1.787	6.736	1.857	6.968	1.892
	60	**2.770**	**1.125**	5.264	1.564	5.278	1.560	4.882	1.483	5.548	1.720	6.870	1.879	7.127	1.918

* *ETT* means the ETTm2. See Appendix A for the **full benchmark** of ETTh1, ETTh2, ETTm1.

- 시계열 분해 모듈 통해 계절성, 피크, 트렌드 등 주요 요소 효과적으로 학습함
- 시간 지연 기반으로 주기성 탐지 → roll된 시계열 집계 유도하여 의존성 학습 진행함

Models		**Autoformer**		N-BEATS[29]		Informer[48]		LogTrans[26]		Reformer[23]		DeepAR[34]		Prophet[39]		ARIMA[1]	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT	96	**0.065**	**0.189**	0.082	0.219	0.088	0.225	0.082	0.217	0.131	0.288	0.099	0.237	0.287	0.456	0.211	0.362
	192	**0.118**	**0.256**	0.120	0.268	0.132	0.283	0.133	0.284	0.186	0.354	0.154	0.310	0.312	0.483	0.261	0.406
	336	**0.154**	**0.305**	0.226	0.370	0.180	0.336	0.201	0.361	0.220	0.381	0.277	0.428	0.331	0.474	0.317	0.448
	720	**0.182**	**0.335**	0.188	0.338	0.300	0.435	0.268	0.407	0.267	0.430	0.332	0.468	0.534	0.593	0.366	0.487
Exchange	96	0.241	0.387	0.156	0.299	0.591	0.615	0.279	0.441	1.327	0.944	0.417	0.515	0.828	0.762	**0.112**	**0.245**
	192	**0.273**	**0.403**	0.669	0.665	1.183	0.912	1.950	1.048	1.258	0.924	0.813	0.735	0.909	0.974	0.304	0.404
	336	**0.508**	**0.539**	0.611	0.605	1.367	0.984	2.438	1.262	2.179	1.296	1.331	0.962	1.304	0.988	0.736	0.598
	720	**0.991**	**0.768**	1.111	0.860	1.872	1.072	2.010	1.247	1.280	0.953	1.894	1.181	3.238	1.566	1.871	0.935

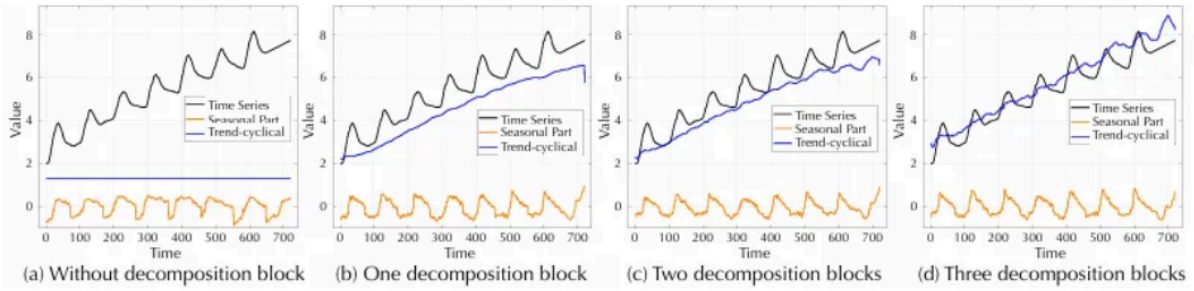
- 단변량 설정에서도 Autoformer가 다른 모델보다 long-term robustness 뛰어남
- 96→96 예측에서는 ARIMA가 더 나은 성능 보였으나, long-term 예측에서는 성능 급격히 저하됨
- ARIMA가 non-stationary한 경제 데이터 처리에는 강점 있으나, 실제 시계열의 복잡한 temporal pattern 학습에는 한계 있음으로 보임

Input-96	Transformer[41]			Informer[48]			LogTrans[23]			Reformer[26]			Promotion	
Predict-O	Origin	Sep	Ours	Origin	Sep	Ours	Origin	Sep	Ours	Origin	Sep	Ours	Sep	Ours
96	0.604	0.311	**0.204**	0.365	0.490	**0.354**	0.768	0.862	**0.231**	0.658	0.445	**0.218**	0.069	0.347
192	1.060	0.760	**0.266**	0.533	0.658	**0.432**	0.989	0.533	**0.378**	1.078	0.510	**0.336**	0.300	0.562
336	1.413	0.665	**0.375**	1.363	1.469	**0.481**	1.334	0.762	**0.362**	1.549	1.028	**0.366**	0.434	1.019
720	2.672	3.200	**0.537**	3.379	2.766	**0.822**	3.048	2.601	**0.539**	2.631	2.845	**0.502**	0.079	2.332

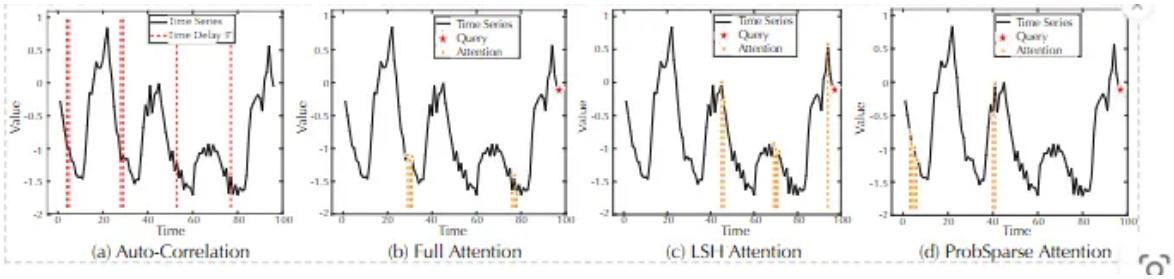
- Autoformer의 progressive decomposition 구조 적용 시, 다른 모델도 성능 향상됨
- 예측 길이 O가 길어질수록 성능 향상 효과 더 뚜렷해짐
- progressive decomposition이 다양한 모델에 일반화 가능, 다른 의존성 학습 메커니즘의 학습 능력 확장 + 복잡한 패턴에서 오는 distraction 완화 효과 있음

Input Length I		96			192			336	
Prediction Length O	336	720	1440	336	720	1440	336	720	1440
Auto-Correlation	MSE	**0.339**	**0.422**	**0.555**	**0.355**	**0.429**	**0.503**	**0.361**	**0.425**	**0.574**
	MAE	**0.372**	**0.419**	**0.496**	**0.392**	**0.430**	**0.484**	**0.406**	**0.440**	**0.534**
Full Attention[41]	MSE	0.375	0.537	0.667	0.450	0.554	-	0.501	0.647	-
	MAE	0.425	0.502	0.589	0.470	0.533	-	0.485	0.491	-
LogSparse Attention[26]	MSE	0.362	0.539	0.582	0.420	0.552	0.958	0.474	0.601	-
	MAE	0.413	0.522	0.529	0.450	0.513	0.736	0.474	0.524	-
LSH Attention[23]	MSE	0.366	0.502	0.663	0.407	0.636	1.069	0.442	0.615	-
	MAE	0.404	0.475	0.567	0.421	0.571	0.756	0.476	0.532	-
ProbSparse Attention[48]	MSE	0.481	0.822	0.715	0.404	1.148	0.732	0.417	0.631	1.133
	MAE	0.472	0.559	0.586	0.425	0.654	0.602	0.434	0.528	0.691

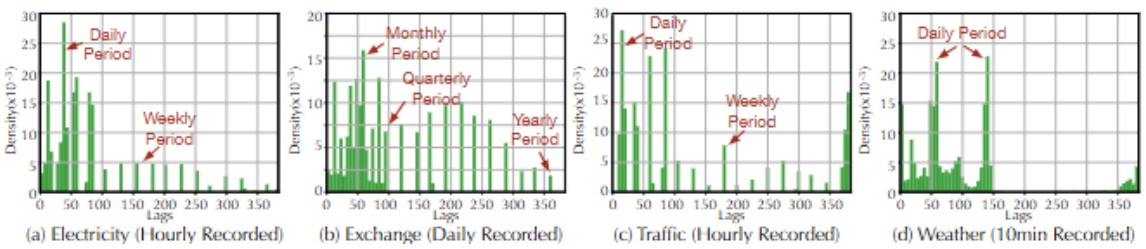
- auto-correlation은 다양한 input-I, predict-O 설정에서 self-attention 기반 방법들보다 더 좋은 성능 보임
- O 길이가 1440일 때 self-attention 계열 모델은 out-of-memory 발생했지만, auto-correlation은 그렇지 않음
- auto-correlation이 성능뿐 아니라 메모리 효율성 면에서도 우수함을 보여줌



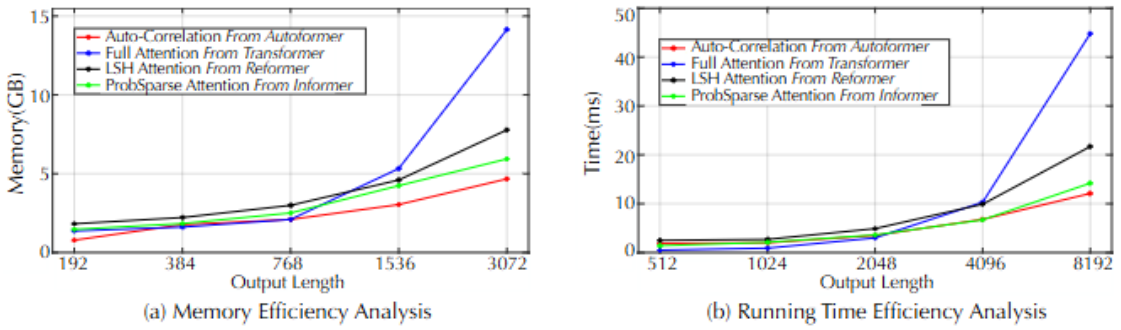
- series decomposition block 없으면 예측 모델이 증가하는 trend나 seasonal part의 peak를 제대로 포착하지 못함
- series decomposition block 추가 시 trend-cyclical part를 잘 정제하고 집계할 수 있음
- seasonal part의 peak와 trough도 더 잘 학습하게 됨



- (a)에서 표시된 time delay 크기는 가장 가능성 높은 주기
- 학습한 periodicity는 Roll(X, τi)로부터 모델이 같은 또는 인접한 phase of periods의 sub-series를 합치도록 가이드
- 마지막 step에서 auto-correlation은 유사한 sub-series를 완전히 활용하는 반면, self-attention 기반은 그렇지 않음 확인 가능



- Autoformer가 deep representations에서 학습한 lag는 원본 시계열의 실제 seasonality를 나타냄



- Auto-correlation과 self-attention 기반 모델 학습 시 실행 메모리 및 시간 비교 결과, auto-correlation(red)이 메모리와 시간 면에서 더 효율적임을 확인 가능

6. Insight

- 시계열 분해와 Auto-Correlation 메커니즘을 구조에 자연스럽게 녹였다는 점
- 특히 복잡한 패턴을 trend와 seasonal로 분리해 안정적인 학습을 유도하고, 주기성 기반으로 종속성을 찾는 방식이 직관적이면서도 효과적이라는 느낌
- self-attention 구조의 한계를 극복하고도 계산 효율성과 메모리 효율성까지 확보한 것도 인상적임.
- 실제 구현 시 어떤 방식으로 trend와 seasonal을 나누는지에 대한 디테일 부족
- 기존 transformer 계열 모델과 직접적인 비교를 통해 얻은 인사이트가 논문에 많지만, 왜 그 구조가 더 잘 작동하는지에 대한 이론적 설명은 조금 부족하게 느껴짐.
- 이후 Auto-Correlation 부분이 어떻게 작동하는지, FFT 기반으로 계산되는 로직 등을 PyTorch 관점에서 살펴볼 계획
- 더 나아가 논문에서 언급한 progressive decomposition을 다른 모델(LSTM, Informer 등)에 적용해보는 것도 실험해볼만하다고 생각함.