NOVA

AI / Data Engineering Track

Time-LLM: Time Series Forecasting By Reprogramming Large Language Models

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y.Zhang, Xiaming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, Qingsong Wen

ICLR 2024

숭실대학교 전기공학부 학술 소모임 NOVA
발표자 : AI 팀 (우상윤, 이다은)

Contents

01 Introduction

02 Related Work

03 Methodology

04 Main Results

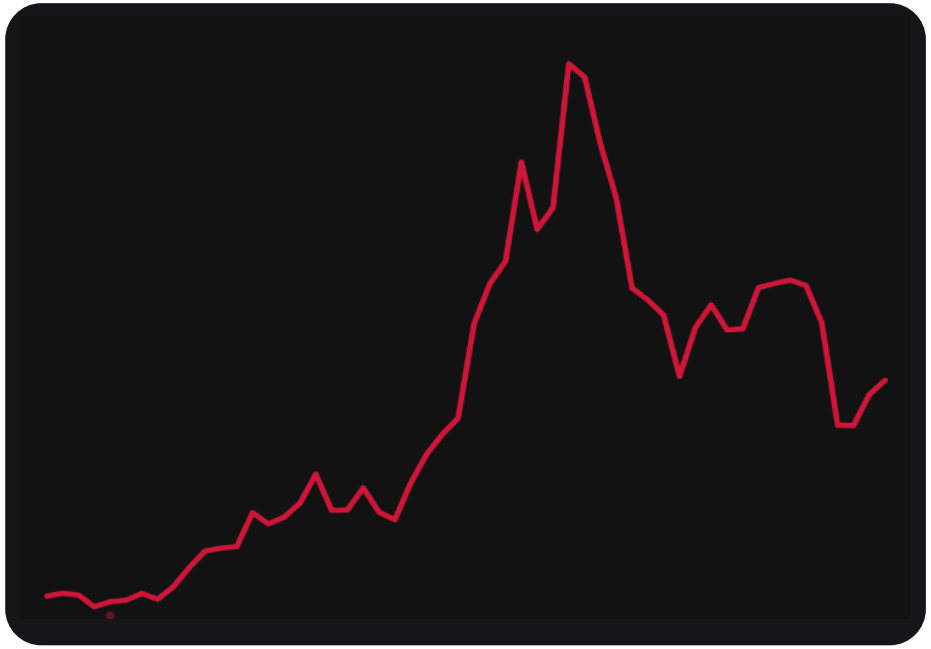
05 Conclusion and Future Work

06 Opinion

01 Introduction

시계열 데이터 : 시간의 흐름에 따라 기록된 데이터

E.g. 주식 가격 및 거래량, 시간대별 기온, 기업의 월별 매출액 등



	날짜	지점	평균기온(℃)	최저기온(℃)	최고기온(℃)
0	1907-10-01	108	13.5	7.9	20.7
1	1907-10-02	108	16.2	7.9	22.0
2	1907-10-03	108	16.2	13.1	21.3
3	1907-10-04	108	16.5	11.2	22.0
4	1907-10-05	108	17.6	10.9	25.4

01 Introduction

시계열 예측을 위해선 해당 분야에 대한 전문 지식이 필요하며,

작업별로 모델을 다르게 설계해야 한다는 단점이 존재함



주식 가격 예측 모델

- 변동성 및 불확실성
- 비정상성
- 지수, 환율, 금리 등

기온은 예측할 수 있지만, 주식 가격 예측은 못함

주식 가격은 예측할 수 있지만, 기온 예측은 못함



기온 예측 모델

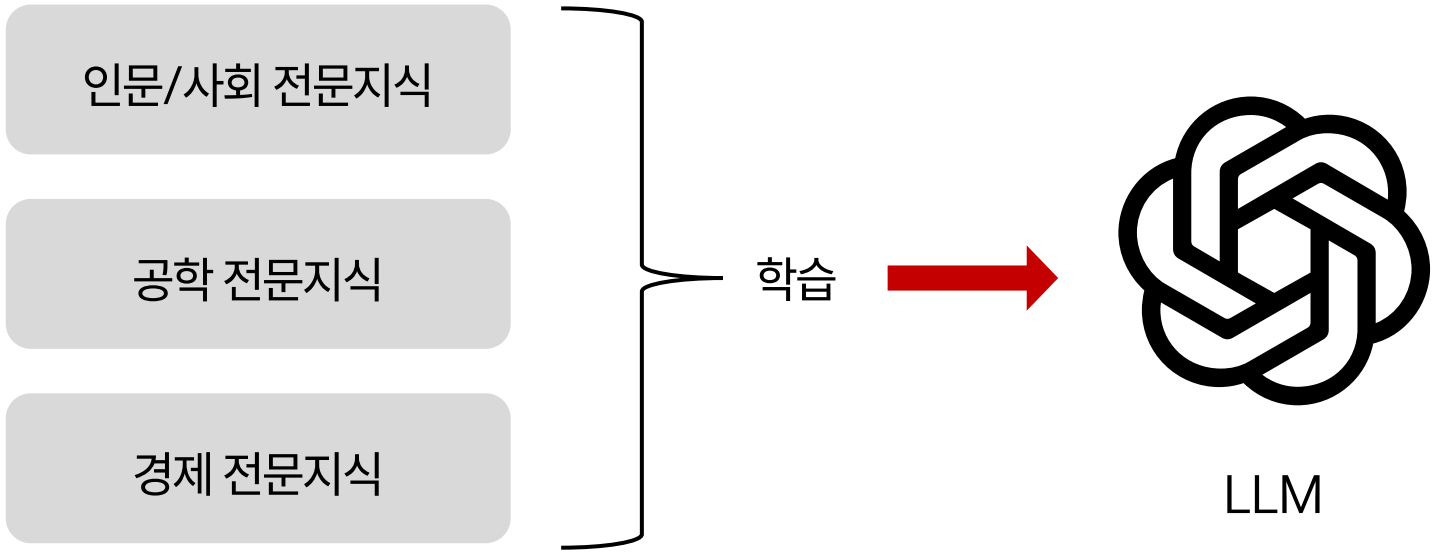
- 계절성 및 주기성
- 공간적 인접성
- 물리적 일관성

01 Introduction

최근 GPT-3, 4와 같은 대형 언어 모델(Large Language Model, LLM)이 등장함

LLM은 학습 과정에서 광범위한 분야의 지식을 학습시킴으로써 거의 모든 분야의 작업에 적용할 수 있으며,

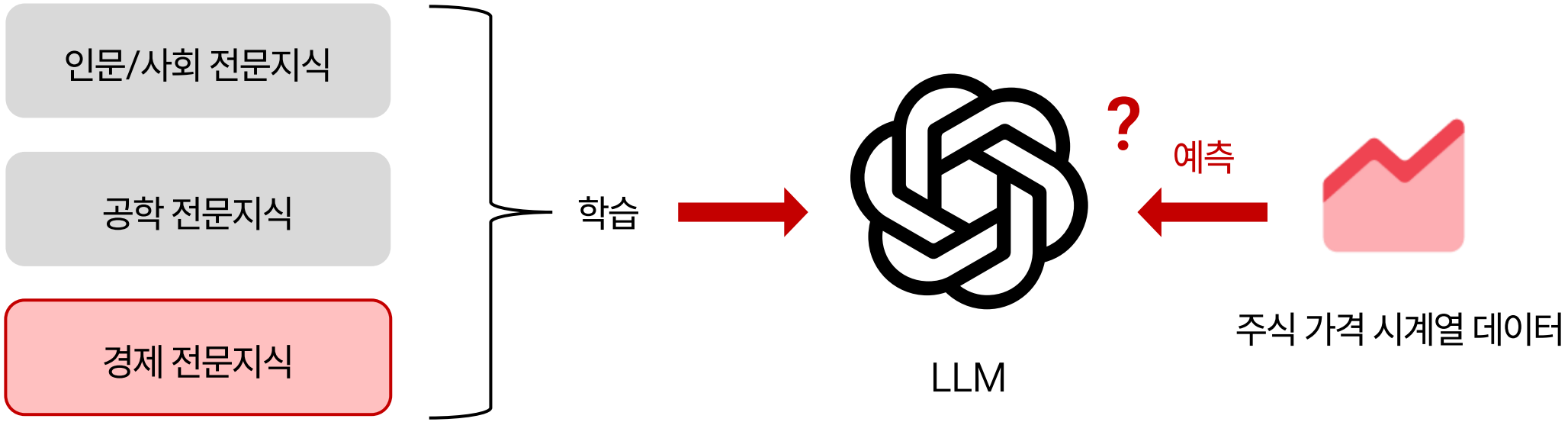
이로 인해 컴퓨터 비전과 자연어 처리 분야가 급격하게 성장하게 됨



01 Introduction

그렇다면, 이미 전문 지식들로 LLM을 학습시킨 거니까

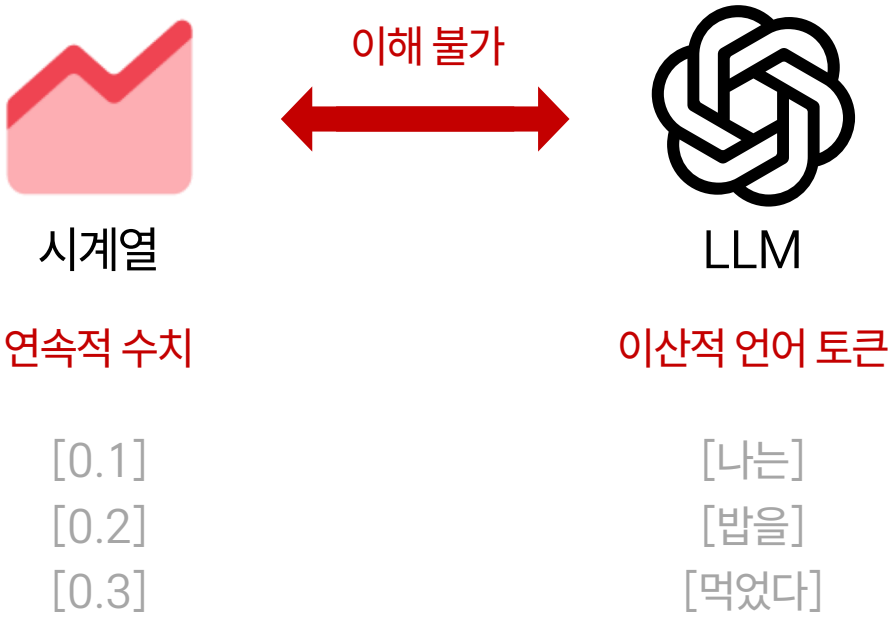
시계열 예측도 LLM을 사용해보면 되지 않을까?

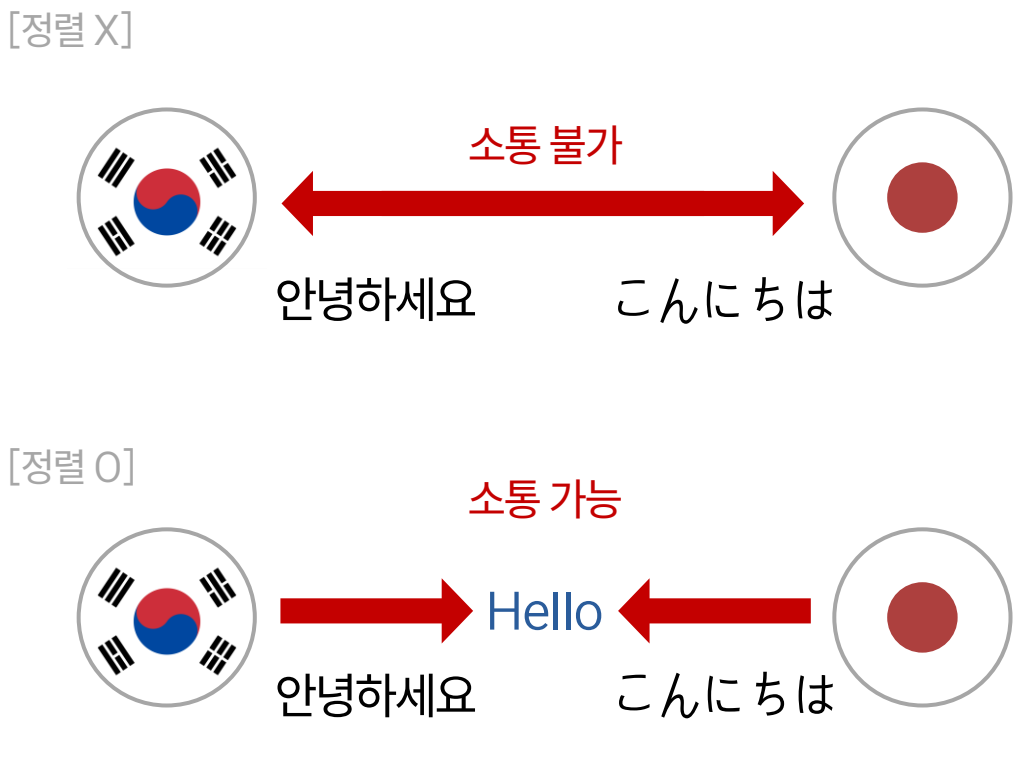


01 Introduction

하지만 무작정 LLM에 시계열 데이터를 입력하는 것은 바람직하지 않고,

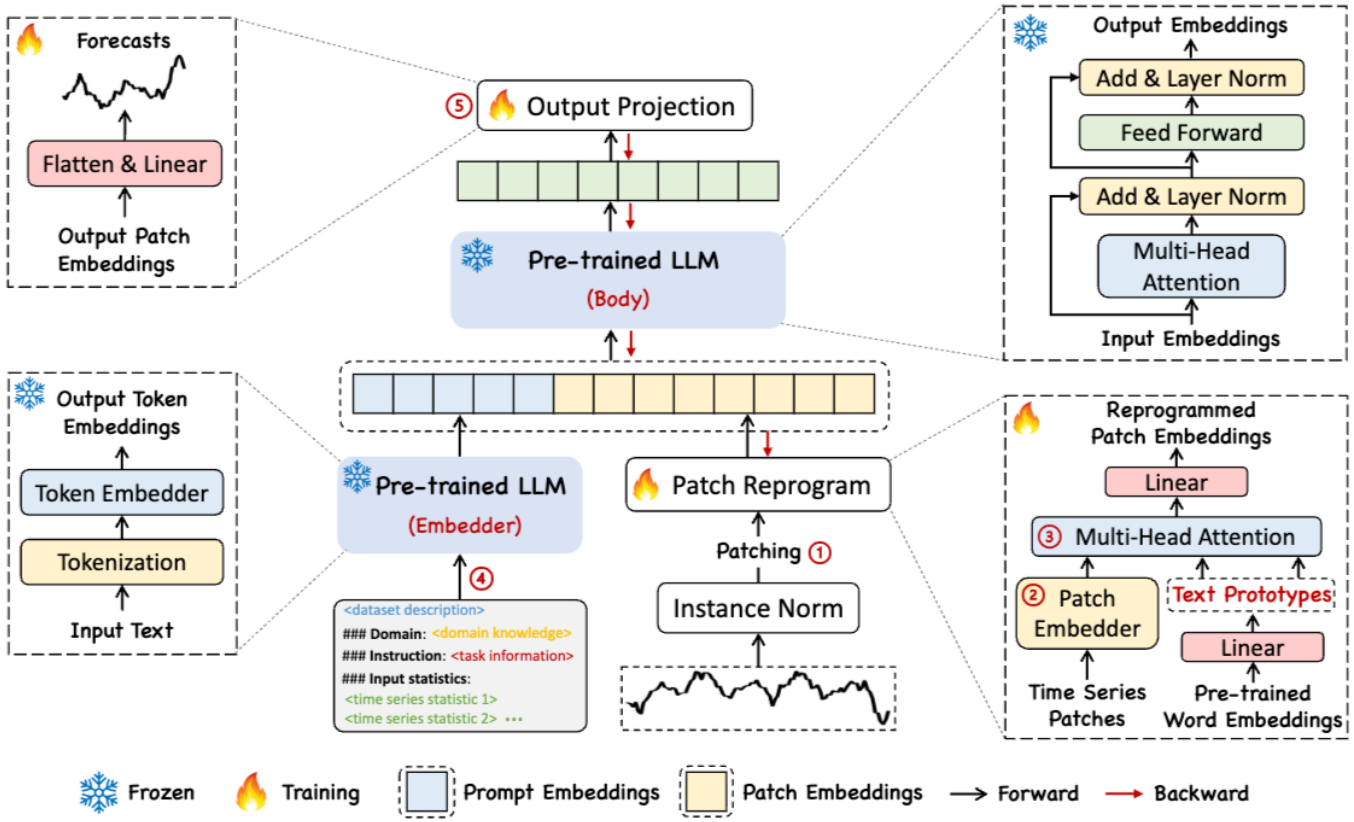
’정렬’이라는 과정을 거쳐야 효과적인 결과가 나타나게 됨





01 Introduction

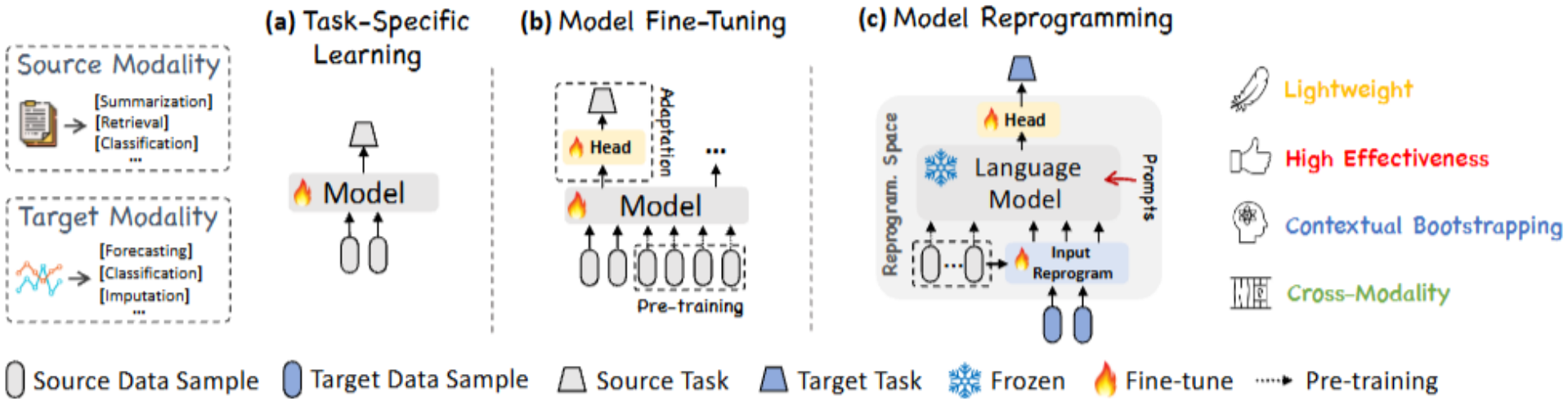
본 논문에서는 정렬을 효과적으로 수행하기 위해 TIME-LLM 이라는 프레임워크를 제안함



02 Related Work

시계열 예측을 위한 기존 연구 흐름은 크게 세 가지의 범주로 분류할 수 있음

- Task-specific Learning (과제 특화 학습)
- In-modality Adaption (내부 모달리티 적응)
- Cross-modality Adaption (교차 모달리티 적응)



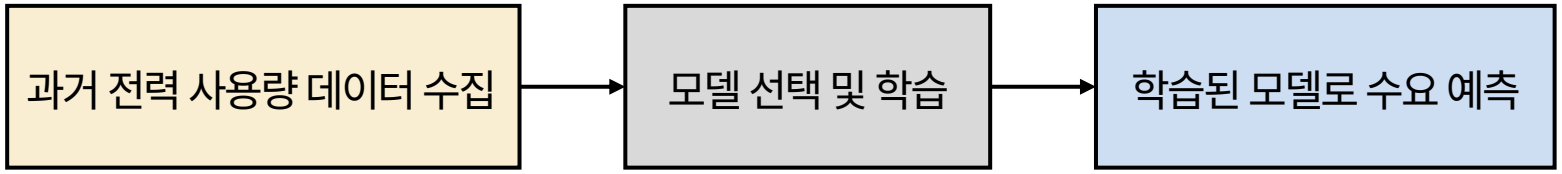
02 Related Work

Task-specific Learning (과제 특화 학습)

기존의 전통적인 시계열 모델들 (ARIMA, LSTM, Transformer 등)

특정 도메인이나 과제에 맞춰 정교하게 설계되었지만, 범용성이 부족하여 데이터가 바뀔 때마다 재학습 또는 구조 변경이 필요

만약 전력 수요 예측을 한다면:



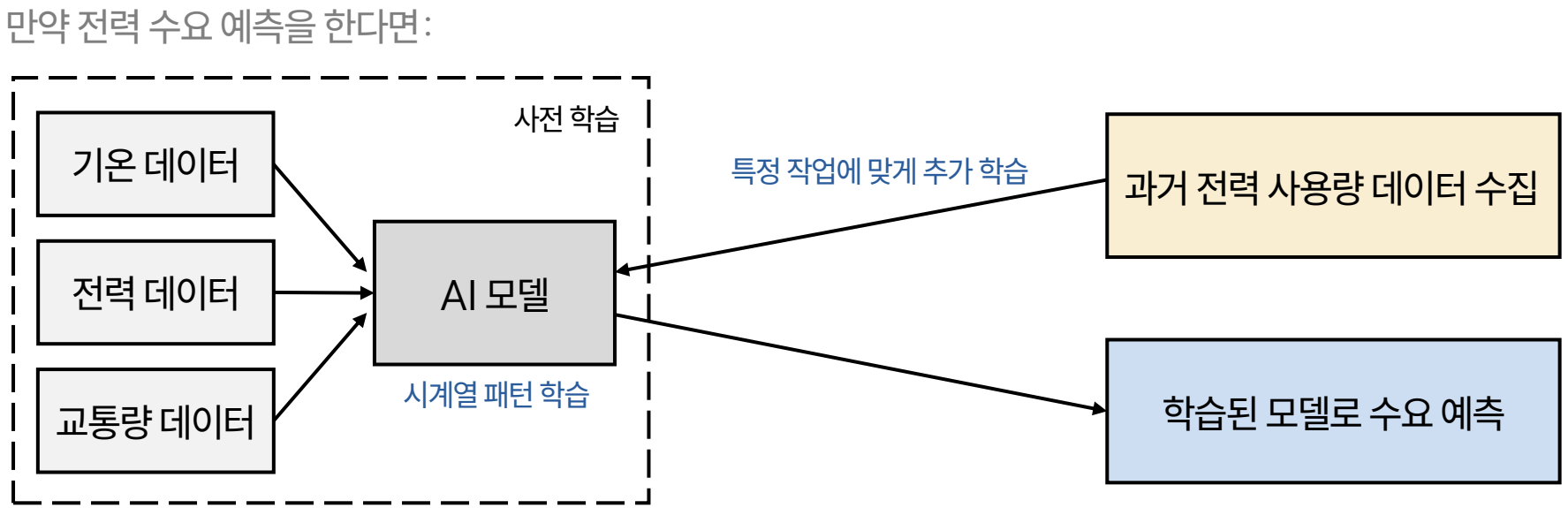
그러나 이 전력수요 예측 모델로 주가 예측은 수행할 수 없다.
(모델의 범용성이 부족)

02 Related Work

In-modality Adaption (내부 모달리티 적응)

다른 인공지능 분야의 사전 학습-미세 조정 성공 사례를 시계열 분야에 도입

시계열 전용 사전 학습 모델을 만든 뒤 하위 과제에 맞게 미세 조정하지만, 데이터 부족 문제로 사전 구축이 어려움

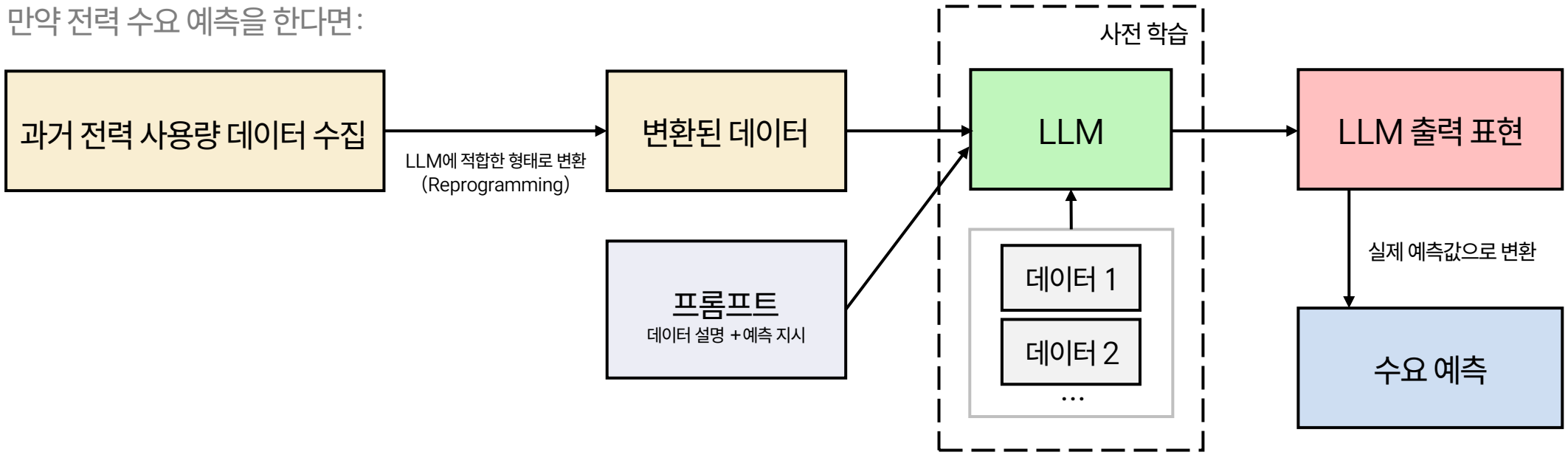


02 Related Work

Cross-modality Adaption (교차 모달리티 적응)

이미 사전학습이 완료된 대형 언어 모델(LLM)을 그대로 두고,

데이터를 모델이 처리할 수 있는 형태로 바꾸는 방법을 학습



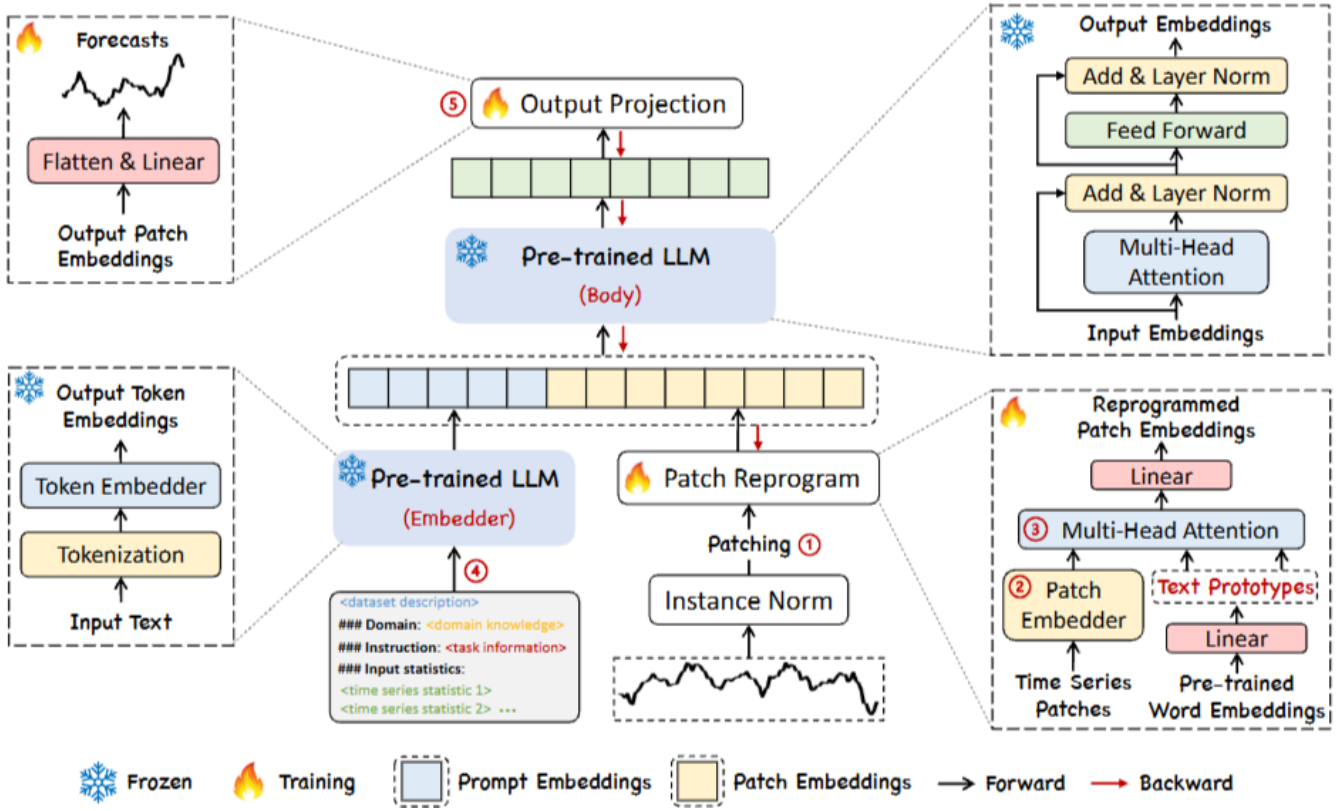
03 Methodology

목표: 1. LLM가 시계열 데이터를 이해할 수 있게 Reprogramming 하면서

2. 정답과 예측값 사이의 평균 제곱 오차를 최소화

03 Methodology

Time-LLM 전체 구조



① Patching
입력 시계열을 작은 구간으로 분할

② Customized Embedding Layer
각 Patch를 벡터로 변환

③ Patch Reprogramming
시계열과 언어의 표현 공간 정렬

④ Prompt-as-Prefix
Prompt를 추가해 LLM 입력에 문맥과 작업 지시 제공

⑤ Output Projection
LLM에서 나온 Patch 표현으로 미래 시계열 예측값 생성

전력 수요 예측을 예시로 모델을 이해해 봅시다.

03 Methodology

1. Input Embedding (1) — Normalization

정규화(Normalization) : 데이터의 크기를 일정한 기준에 맞추는 작업

도시 1



전력 수요 약 100MW

도시 2



전력 수요 약 10000MW

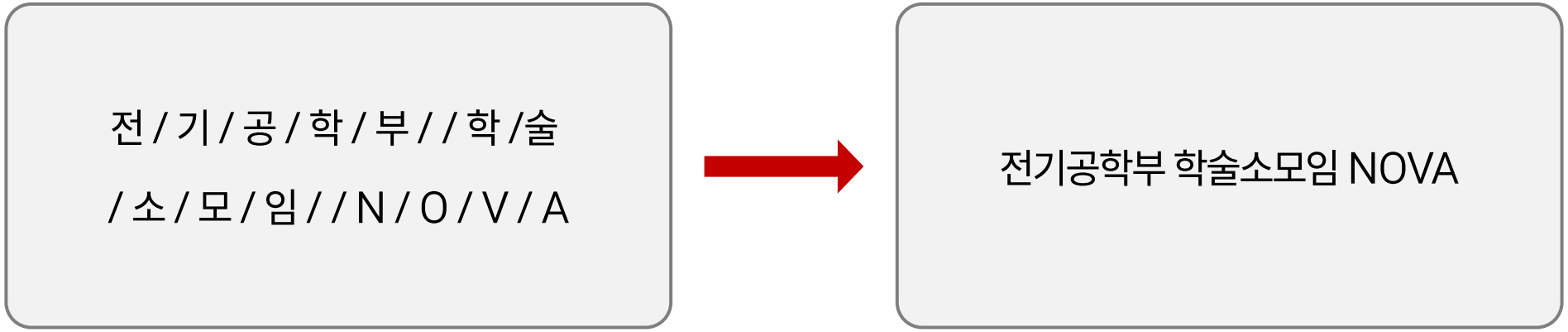
두 도시의 전력 수요는 약 100배 차이, 그러나 비슷한 상승/하락 패턴을 지닐 수 있음

모델이 데이터의 규모 차이에 영향을 덜 받도록 평균과 표준편차를 이용해 크기를 조절하는 과정

03 Methodology

1. Input Embedding (1) — Patching

패치 분할 : 긴 시계열 데이터를 일정한 구간 단위로 나누는 작업



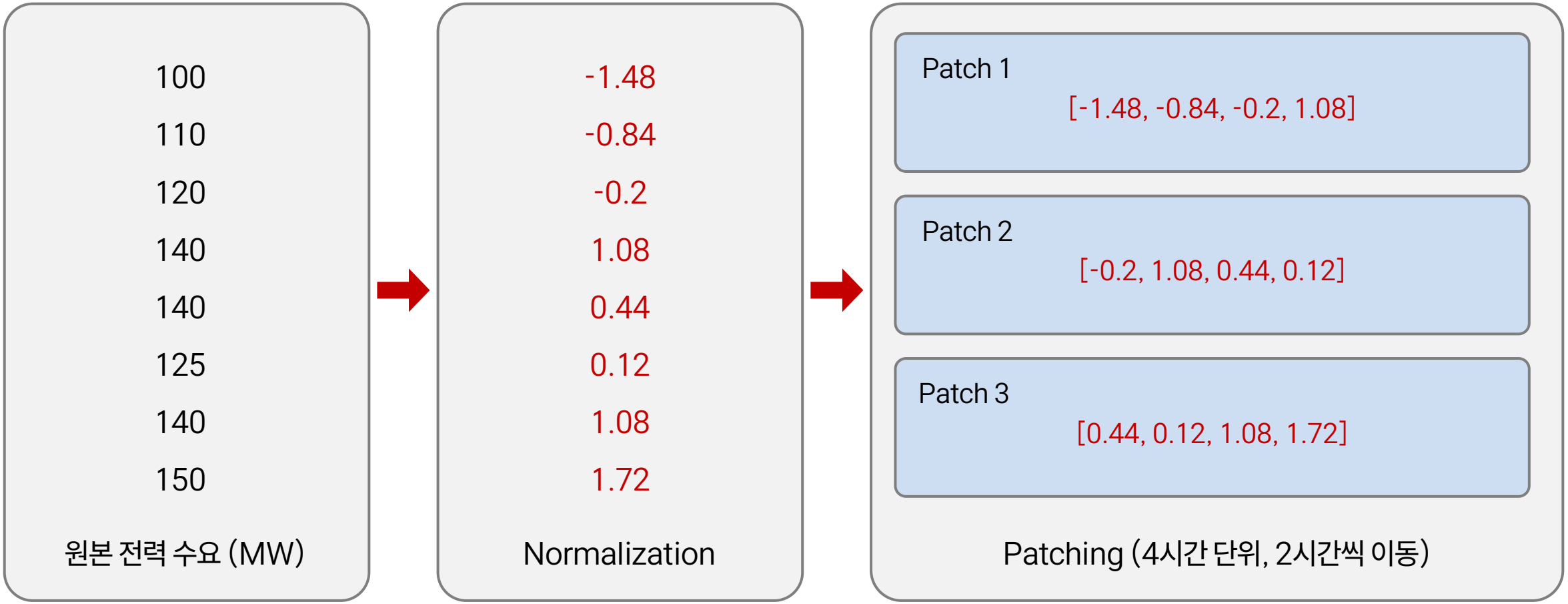
글자를 하나씩 읽는 것 보다, 문장이나 단어 단위로 묶어서 읽으면 더 이해하기 쉽다

시계열 데이터를 일정 구간 단위로 나눔으로써

변화 패턴을 더욱 쉽게 파악할 수 있음

03 Methodology

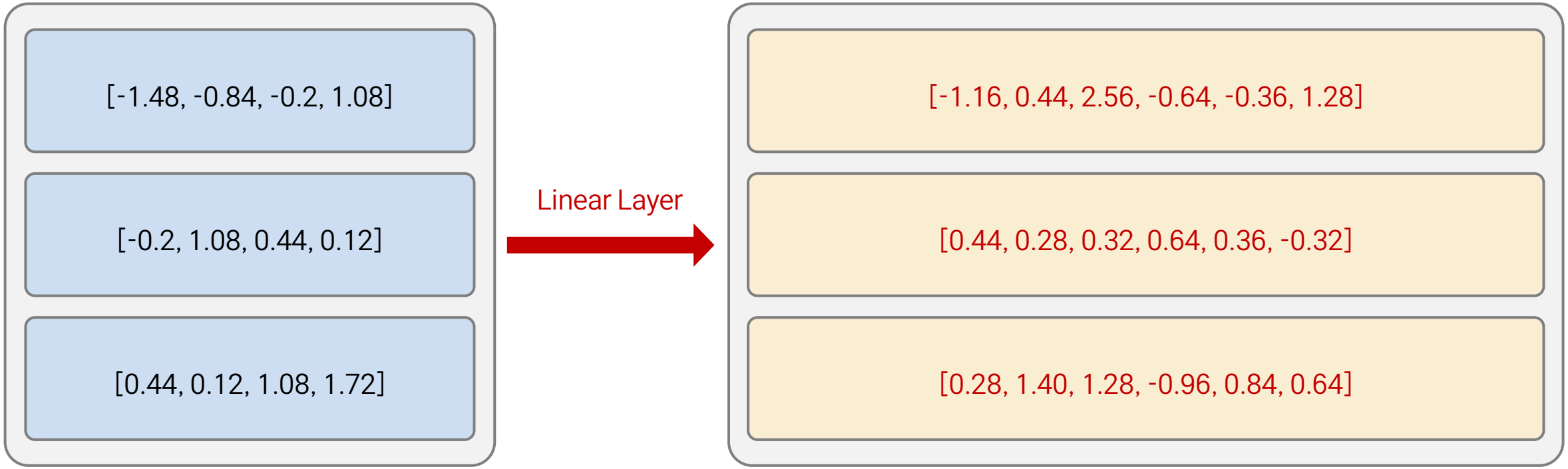
1. Input Embedding (1) 예시



03 Methodology

2. Input Embedding (2) — Patch Embedding

각 Patch를 Linear Layer에 통과시켜 벡터로 변환



현재는 단순히 시간별 전력 숫자의 묶음이므로,
신경망이 계산할 수 있는 형태로 변환이 필요

각 전력 수요 구간을
신경망이 계산할 수 있는 벡터로 변환

03 Methodology

3. Patch Reprogramming (1) — Text Prototypes

벡터를 LLM이 익숙한 표현 공간과 연결하기 위해 Text Prototypes를 생성

LLM이 학습했던 것:

자연어 텍스트 (O)

전력 수요 데이터 (X)

전력 수요 데이터를
기존 자연어 텍스트의 벡터로 표현해야 함

기존 자연어 토큰들의 벡터

Up : [1, 2, 0]

Down : [-1, 0, 2]

Steady : [0, 1, 1]

기존 벡터들을 조합하여 Text Prototype 생성

Text Prototype 1 (Up 60%, Down 10%, Steady 30%)

$P_1 = 0.6[1, 2, 0] + 0.1[-1, 0, 2] + 0.3[0, 1, 1]$
$= [0.5, 1.5, 0.5]$

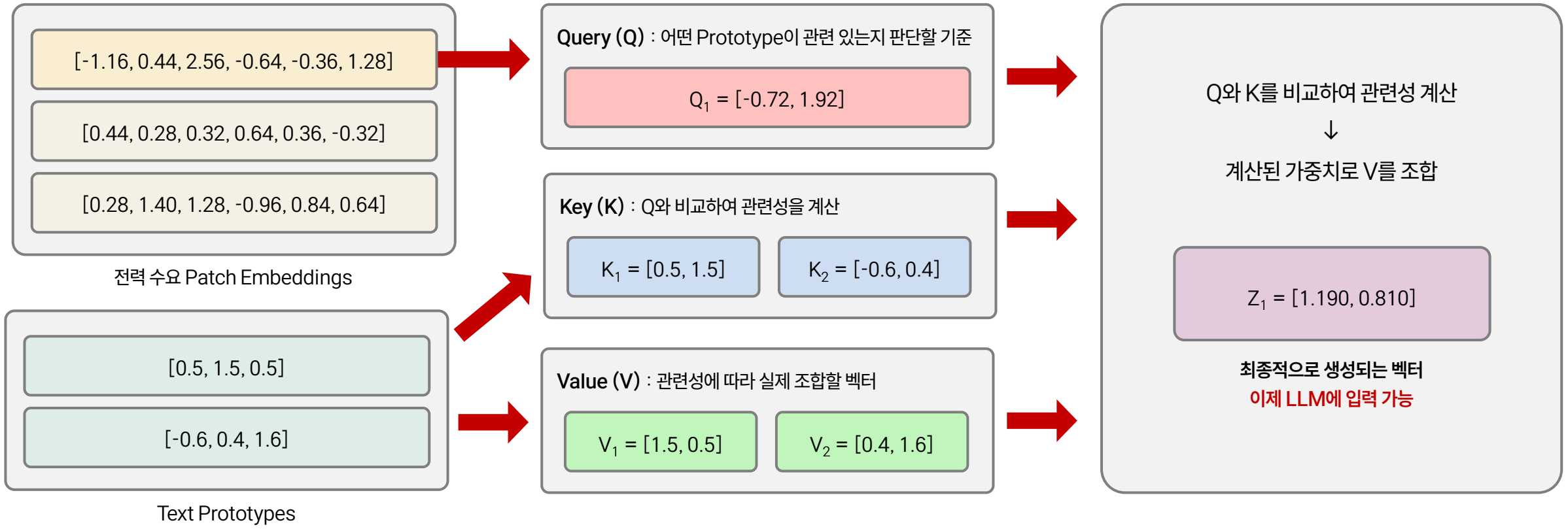
Text Prototype 2 (Up 10%, Down 70%, Steady 20%)

$P_2 = 0.1[1, 2, 0] + 0.7[-1, 0, 2] + 0.2[0, 1, 1]$
$= [-0.6, 0.4, 1.6]$

03 Methodology

4. Patch Reprogramming (2) — Cross-Attention

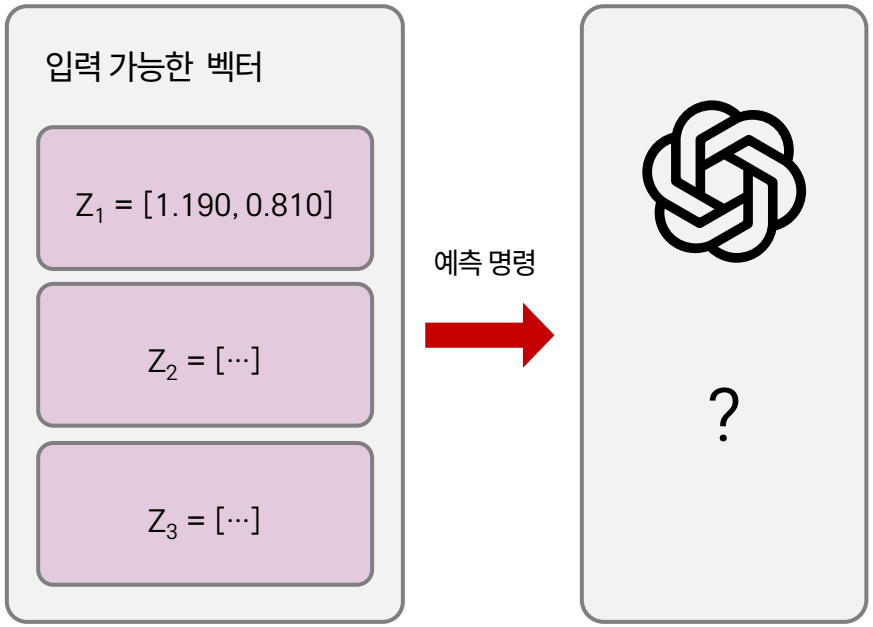
각 전력 수요 Patch가 Text Prototypes을 참고하여 새로운 벡터를 생성



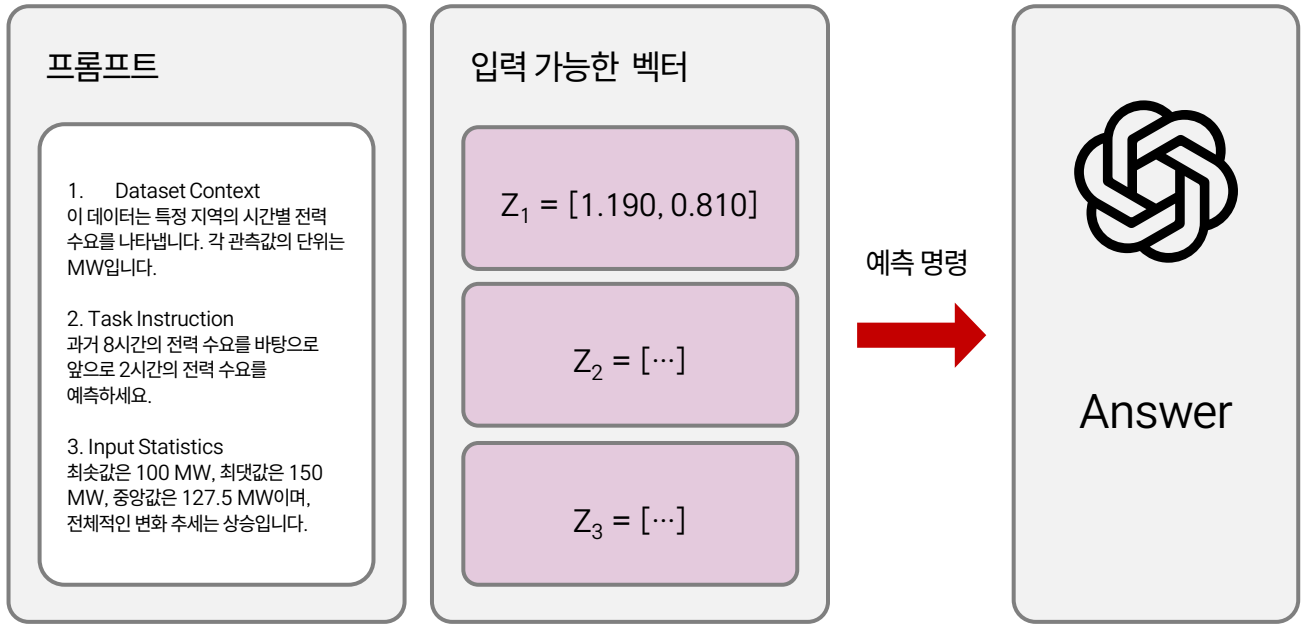
03 Methodology

5. Prompt-as-Prefix (PaP)

데이터에 관한 설명과 목표 등을 프롬프트로 제공



데이터만 제공할 경우,
이 데이터가 무엇인지, 무엇을 예측해야 하는지 등에 대한 정보가 부족함

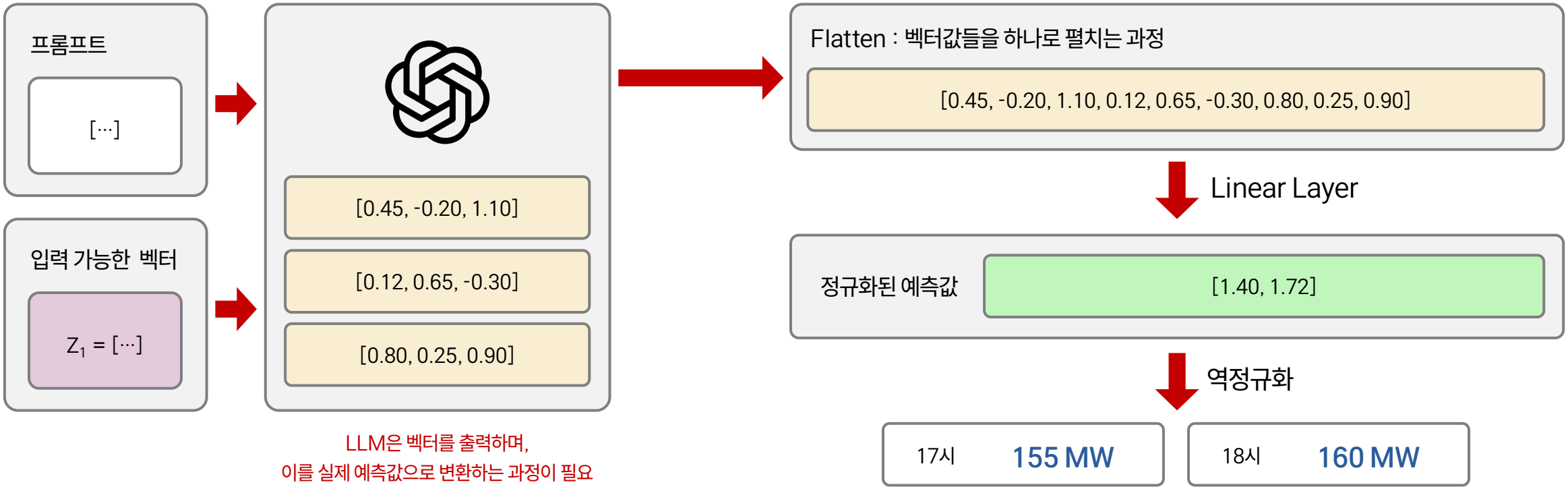


프롬프트를 활용하여
데이터에 대한 설명, 예측 작업에 대한 지시, 입력 데이터의 통계 정보 등을 제공

03 Methodology

6. Output Projection

LLM의 출력은 곧바로 예측값으로 나오는 것이 아니며, 최종적으로 변환이 필요



04 Main Results

본 논문은 변압기 온도 및 부하, 전력 사용량, 교통량, 기상 관측 데이터 등을 사용해 실험을 진행









04 Main Results

Long-term Forecasting (장기 예측 성능)

과거 데이터를 이용해 비교적 먼 미래까지 예측하는 작업

Table 1: Long-term forecasting results. All results are averaged from four different forecasting horizons: $H \in \{24, 36, 48, 60\}$ for ILI and $\{96, 192, 336, 720\}$ for the others. A lower value indicates better performance. **Red**: the best, Blue: the second best. Our full results are in Appendix D.

Methods	TIME-LLM (Ours)		GPT4TS (2023a)		DLinear (2023)		PatchTST (2023)		TimesNet (2023)		FEDformer (2022)		Autoformer (2021)		Stationary (2022)		ETSformer (2022)		LightTS (2022a)		Informer (2021)		Reformer (2020)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.408**	**0.423**	0.465	0.455	0.422	0.437	0.413	0.430	0.458	0.450	0.440	0.460	0.496	0.487	0.570	0.537	0.542	0.510	0.491	0.479	1.040	0.795	1.029	0.805
ETTh2	0.334	0.383	0.381	0.412	0.431	0.446	**0.330**	**0.379**	0.414	0.427	0.437	0.449	0.450	0.459	0.526	0.516	0.439	0.452	0.602	0.543	4.431	1.729	6.736	2.191
ETTm1	**0.329**	**0.372**	0.388	0.403	0.357	0.378	0.351	0.380	0.400	0.406	0.448	0.452	0.588	0.517	0.481	0.456	0.429	0.425	0.435	0.437	0.961	0.734	0.799	0.671
ETTm2	**0.251**	**0.313**	0.284	0.339	0.267	0.333	0.255	0.315	0.291	0.333	0.305	0.349	0.327	0.371	0.306	0.347	0.293	0.342	0.409	0.436	1.410	0.810	1.479	0.915
Weather	**0.225**	**0.257**	0.237	0.270	0.248	0.300	**0.225**	0.264	0.259	0.287	0.309	0.360	0.338	0.382	0.288	0.314	0.271	0.334	0.261	0.312	0.634	0.548	0.803	0.656
ECL	**0.158**	**0.252**	0.167	0.263	0.166	0.263	0.161	**0.252**	0.192	0.295	0.214	0.327	0.227	0.338	0.193	0.296	0.208	0.323	0.229	0.329	0.311	0.397	0.338	0.422
Traffic	**0.388**	0.264	0.414	0.294	0.433	0.295	0.390	**0.263**	0.620	0.336	0.610	0.376	0.628	0.379	0.624	0.340	0.621	0.396	0.622	0.392	0.764	0.416	0.741	0.422
ILI	**1.435**	0.801	1.925	0.903	2.169	1.041	1.443	**0.797**	2.139	0.931	2.847	1.144	3.006	1.161	2.077	0.914	2.497	1.004	7.382	2.003	5.137	1.544	4.724	1.445
1st Count	7		0		0		5		0		0		0		0		0		0		0		0	

04 Main Results

Short-term Forecasting (단기 예측 성능)

과거 데이터를 이용해 비교적 가까운 미래를 예측하는 작업

Table 2: Short-term time series forecasting results on M4. The forecasting horizons are in [6, 48] and the three rows provided are weighted averaged from all datasets under different sampling intervals. A lower value indicates better performance. **Red**: the best, Blue: the second best. More results are in Appendix D.

Methods		TIME-LLM (Ours)	GPT4TS (2023a)	TimesNet (2023)	PatchTST (2023)	N-HiTS (2023b)	N-BEATS (2020)	ETSformer (2022)	LightTS (2022a)	DLinear (2023)	FEDformer (2022)	Stationary (2022)	Autoformer (2021)	Informer (2021)	Reformer (2020)
Average	SMAPE	**11.983**	12.69	12.88	12.059	12.035	12.25	14.718	13.525	13.639	13.16	12.780	12.909	14.086	18.200
	MASE	**1.595**	1.808	1.836	1.623	1.625	1.698	2.408	2.111	2.095	1.775	1.756	1.771	2.718	4.223
	OWA	**0.859**	0.94	0.955	0.869	0.869	0.896	1.172	1.051	1.051	0.949	0.930	0.939	1.230	1.775

04 Main Results

Few-shot Forecasting (학습 데이터가 부족한 상황)

전체 데이터셋의 일부 (Table 3의 경우 10%)만을 사용했을 때 성능이 어떻게 나오는지 확인

Table 3: Few-shot learning on 10% training data. We use the same protocol in Tab. 1. All results are averaged from four different forecasting horizons: $H \in \{96, 192, 336, 720\}$. Our full results are in Appendix E.

Methods	TIME-LLM (Ours)		GPT4TS (2023a)		DLinear (2023)		PatchTST (2023)		TimesNet (2023)		FEDformer (2022)		Autoformer (2021)		Stationary (2022)		ETSformer (2022)		LightTS (2022a)		Informer (2021)		Reformer (2020)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.556**	**0.522**	0.590	0.525	0.691	0.600	0.633	0.542	0.869	0.628	0.639	0.561	0.702	0.596	0.915	0.639	1.180	0.834	1.375	0.877	1.199	0.809	1.249	0.833
ETTh2	**0.370**	**0.394**	0.397	0.421	0.605	0.538	0.415	0.431	0.479	0.465	0.466	0.475	0.488	0.499	0.462	0.455	0.894	0.713	2.655	1.160	3.872	1.513	3.485	1.486
ETTm1	**0.404**	**0.427**	0.464	0.441	0.411	0.429	0.501	0.466	0.677	0.537	0.722	0.605	0.802	0.628	0.797	0.578	0.980	0.714	0.971	0.705	1.192	0.821	1.426	0.856
ETTm2	**0.277**	**0.323**	0.293	0.335	0.316	0.368	0.296	0.343	0.320	0.353	0.463	0.488	1.342	0.930	0.332	0.366	0.447	0.487	0.987	0.756	3.370	1.440	3.978	1.587
Weather	**0.234**	**0.273**	0.238	0.275	0.241	0.283	0.242	0.279	0.279	0.301	0.284	0.324	0.300	0.342	0.318	0.323	0.318	0.360	0.289	0.322	0.597	0.495	0.546	0.469
ECL	**0.175**	0.270	0.176	**0.269**	0.180	0.280	0.180	0.273	0.323	0.392	0.346	0.427	0.431	0.478	0.444	0.480	0.660	0.617	0.441	0.489	1.195	0.891	0.965	0.768
Traffic	**0.429**	0.306	0.440	0.310	0.447	0.313	0.430	**0.305**	0.951	0.535	0.663	0.425	0.749	0.446	1.453	0.815	1.914	0.936	1.248	0.684	1.534	0.811	1.551	0.821
1st Count	**7**		1		0		1		0		0		0		0		0		0		0		0	

04 Main Results

Zero-shot Forecasting (처음 보는 데이터셋에서 예측을 수행)

한 데이터셋에서 학습한 모델을 다른 데이터셋에 적용했을 때 추가 학습 없이도 잘 예측하는지 확인

Table 5: Zero-shot learning results. **Red**: the best, Blue: the second best. Appendix E shows our detailed results.

Methods	TIME-LLM (**Ours**)		GPT4TS (2023a)		LLMTime (2023)		DLinear (2023)		PatchTST (2023)		TimesNet (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh1 \rightarrow ETTh2$	**0.353**	**0.387**	0.406	0.422	0.992	0.708	0.493	0.488	0.380	0.405	0.421	0.431
$ETTh1 \rightarrow ETTm2$	**0.273**	**0.340**	0.325	0.363	1.867	0.869	0.415	0.452	0.314	0.360	0.327	0.361
$ETTh2 \rightarrow ETTh1$	**0.479**	**0.474**	0.757	0.578	1.961	0.981	0.703	0.574	0.565	0.513	0.865	0.621
$ETTh2 \rightarrow ETTm2$	**0.272**	**0.341**	0.335	0.370	1.867	0.869	0.328	0.386	0.325	0.365	0.342	0.376
$ETTm1 \rightarrow ETTh2$	**0.381**	**0.412**	0.433	0.439	0.992	0.708	0.464	0.475	0.439	0.438	0.457	0.454
$ETTm1 \rightarrow ETTm2$	**0.268**	**0.320**	0.313	0.348	1.867	0.869	0.335	0.389	0.296	0.334	0.322	0.354
$ETTm2 \rightarrow ETTh2$	**0.354**	**0.400**	0.435	0.443	0.992	0.708	0.455	0.471	0.409	0.425	0.435	0.443
$ETTm2 \rightarrow ETTm1$	**0.414**	**0.438**	0.769	0.567	1.933	0.984	0.649	0.537	0.568	0.492	0.769	0.567

04 Main Results

구성 요소 중 일부가 존재하지 않는다면?

B.1 : 시계열 데이터를 LLM이 인식할 수 있도록 벡터로 변환하는 과정

B.2 : 프롬프트를 통해 데이터 정보와 문맥을 제공하는 과정

C.1, C.2, C.3 : 프롬프트 내에서의 문맥 / 작업 지침 / 통계적 정보

Table 6: Ablations on ETTh1 and ETTm1 in predicting 96 and 192 steps ahead (MSE reported). **Red**: the best.

Variant	Long-term Forecasting				Few-shot Forecasting			
	ETTh1-96	ETTh1-192	ETTm1-96	ETThm1-192	ETTh1-96	ETTh1-192	ETTm1-96	ETThm1-192
A.1 Llama (**Default**; 32)	**0.362**	**0.398**	**0.272**	**0.310**	**0.448**	**0.484**	**0.346**	**0.373**
A.2 Llama (8)	0.389	0.412	0.297	0.329	0.567	0.632	0.451	0.490
A.3 GPT-2 (12)	0.385	0.419	0.306	0.332	0.548	0.617	0.447	0.509
A.4 GPT-2 (6)	0.394	0.427	0.311	0.342	0.571	0.640	0.468	0.512
B.1 w/o Patch Reprogramming	0.410	0.412	0.310	0.342	0.498	0.570	0.445	0.487
B.2 w/o Prompt-as-Prefix	0.398	0.423	0.298	0.339	0.521	0.617	0.432	0.481
C.1 w/o Dataset Context	0.402	0.417	0.298	0.331	0.491	0.538	0.392	0.447
C.2 w/o Task Instruction	0.388	0.420	0.285	0.327	0.476	0.529	0.387	0.439
C.3 w/o Statistical Context	0.391	0.419	0.279	0.347	0.483	0.547	0.421	0.461

04 Main Results

시계열을 벡터로 변환하는 과정 (Reprogramming Network)은 비교적 가벼운 편이며,

LLM의 자체 연산이 전체 연산량에서 큰 비중을 차지하고 있음

Table 7: Efficiency analysis of TIME-LLM on ETTh1 in forecasting different steps ahead.

Length	ETTh1-96			ETTh1-192			ETTh1-336			ETTh1-512		
Metric	Param. (M)	Mem. (MiB)	Speed(s/iter)	Param. (M)	Mem. (MiB)	Speed(s/iter)	Param. (M)	Mem. (MiB)	Speed(s/iter)	Param. (M)	Mem.(MiB)	Speed(s/iter)
D.1 LLama (32)	3404.53	32136	0.517	3404.57	33762	0.582	3404.62	37988	0.632	3404.69	39004	0.697
D.2 LLama (8)	975.83	11370	0.184	975.87	12392	0.192	975.92	13188	0.203	976.11	13616	0.217
D.3 w/o LLM	6.39	3678	0.046	6.42	3812	0.087	6.48	3960	0.093	6.55	4176	0.129

05 Conclusion and Future Work

Time-LLM은 LLM을 활용하여 시계열 데이터를 예측하는 새로운 방식을 제시하였으며,

향후 연구에서는 최적의 Reprogramming 방식을 탐구하고 기타 양식 전반에 걸친 더 범용적인 모델을 구축해야 함

06 Opinion

흔히 사용하는 GPT와 같은 LLM을 통해 데이터를 예측하는 방식은

전력, 제어 등 다양한 분야에서 사용 가능하므로 향후 졸업설계 등에서 사용될 수 있는 가능성 또한 존재함

그러나 일부 데이터셋에서는 성능이 상대적으로 밀리는 모습을 보이며,

연산 속도가 LLM의 크기로 결정되기 때문에 실시간 예측 작업에서는 사용이 어려울 것으로 보이기에

SLM(소규모 언어 모델)을 활용한 방식도 연구해볼 가치가 있어 보임

NOVA

Thank you

숭실대학교 전기공학부 학술 소모임 NOVA