



BITAMIN 2025-겨울 프로젝트

TEMPO 모델을 활용한 원달러 환율 변동 예측 및 환차익 전략 수립

Presented by:

시계열 1조: 문예림 박보희 양태원 이수민 이수아 이채연

CONTENT

1

프로젝트 배경

2

TEMPO 설명

3

데이터 설명

4

TEMPO

5

Prophet

6

환차의 전략 수립

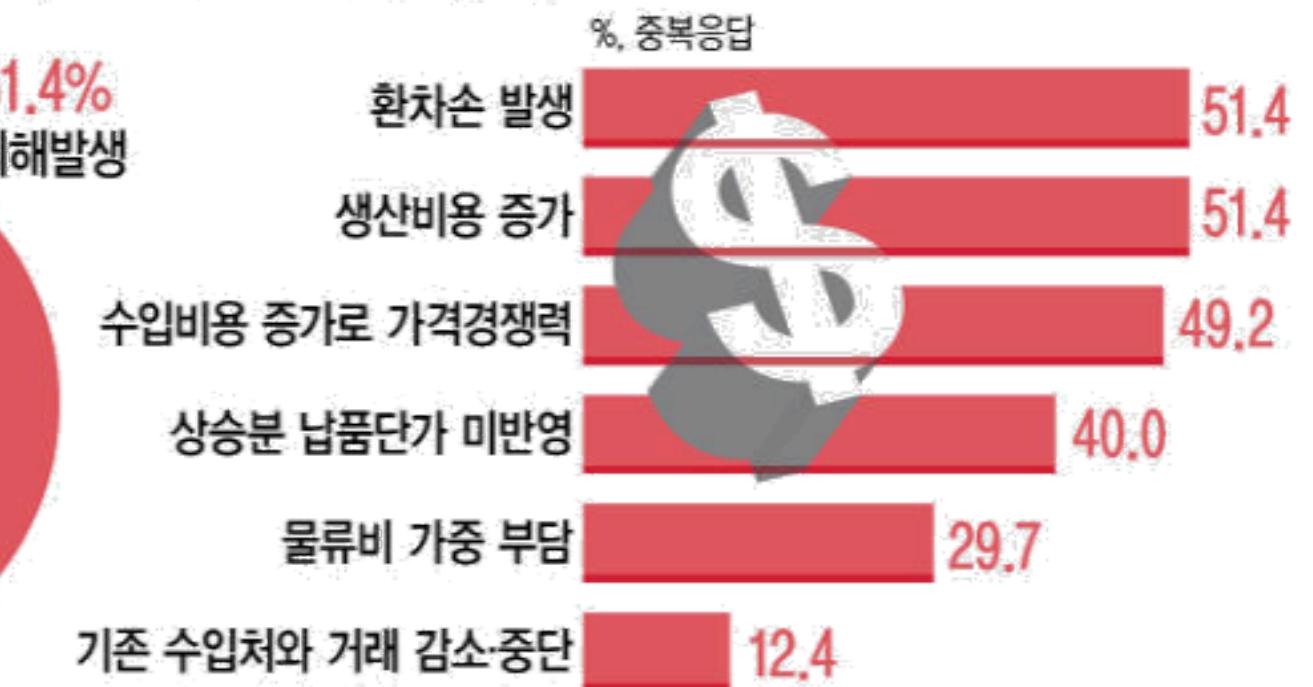
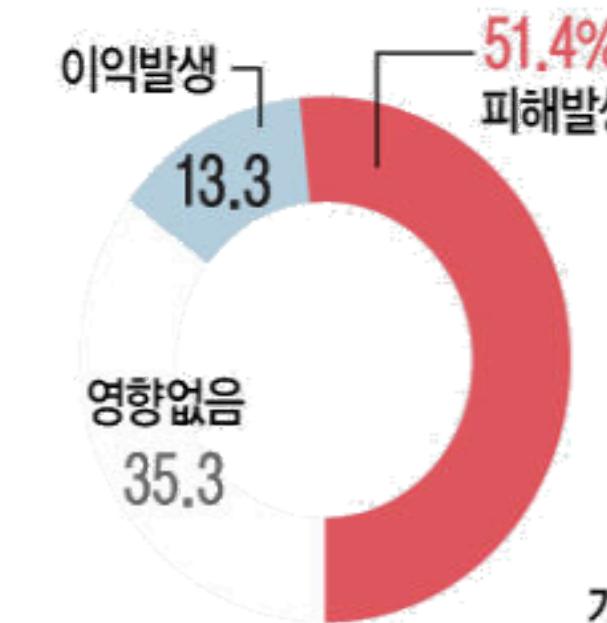
SECTION 01

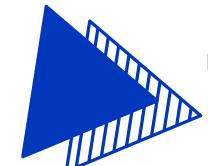
프로젝트 배경

▶ TEMPO 선정 이유



환율 급등에 따른 중소기업 피해 유형





TEMPO 선정 이유

1. GPT-2 기반
2. STL Decomposition을 활용한 시계열 분해
3. Soft Prompt Pool 도입
4. Zero-shot Forecasting & Multimodal Forecasting

| EBITDA Dataset | | | | | | | | | |
|----------------|--------------------|--------------------|-------------|-------------|--------------------|-------------|-------------|-------------|-------------|
| Sectors | TEMPO | LLaMA | GPT2 | Bert | T5 | Informer | PatchTST | Reformer | DLinear |
| CC | 32.27/33.48 | 33.13/34.31 | 33.77/35.37 | 33.42/35.33 | <u>32.65/33.83</u> | 41.12/43.17 | 41.44/43.18 | 37.23/39.09 | 33.53/35.65 |
| CD | 25.9/26.25 | <u>26.34/26.62</u> | 26.86/27.15 | 27.34/28.3 | <u>26.44/26.79</u> | 35.65/36.08 | 31.6/31.98 | 29.93/30.36 | 27.01/28.04 |
| Ind | 26.7/27.42 | <u>27.17/27.98</u> | 27.9/28.63 | 27.89/28.95 | 27.3/28.12 | 34.83/35.87 | 33.84/34.87 | 30.23/31.28 | 27.59/28.84 |
| RE | 29.46/30.11 | <u>29.63/30.48</u> | 30.62/31.21 | 30.62/31.66 | 30.1/30.64 | 36.4/37.22 | 37.63/38.31 | 31.23/31.69 | 29.95/30.92 |

그림 11. TEMPO^[4]와 Baseline 방법론들의 Multimodal 예측 성능 비교표

SECTION 02

TEMPO 설명

▶ TEMPO 모델이란?

Time sEries proMpt POol

- 시계열(Time-Series) + 텍스트(Text)를 함께 학습하는 멀티모달(Multi-modal) 예측 모델
- 기존 시계열 모델보다 장기 예측(Long-term Forecasting)에서 성능이 뛰어남
- 프롬프트를 입력값으로 받아서, 새로운 학습 없이도 다양한 예측이 가능

TEMPO Overview

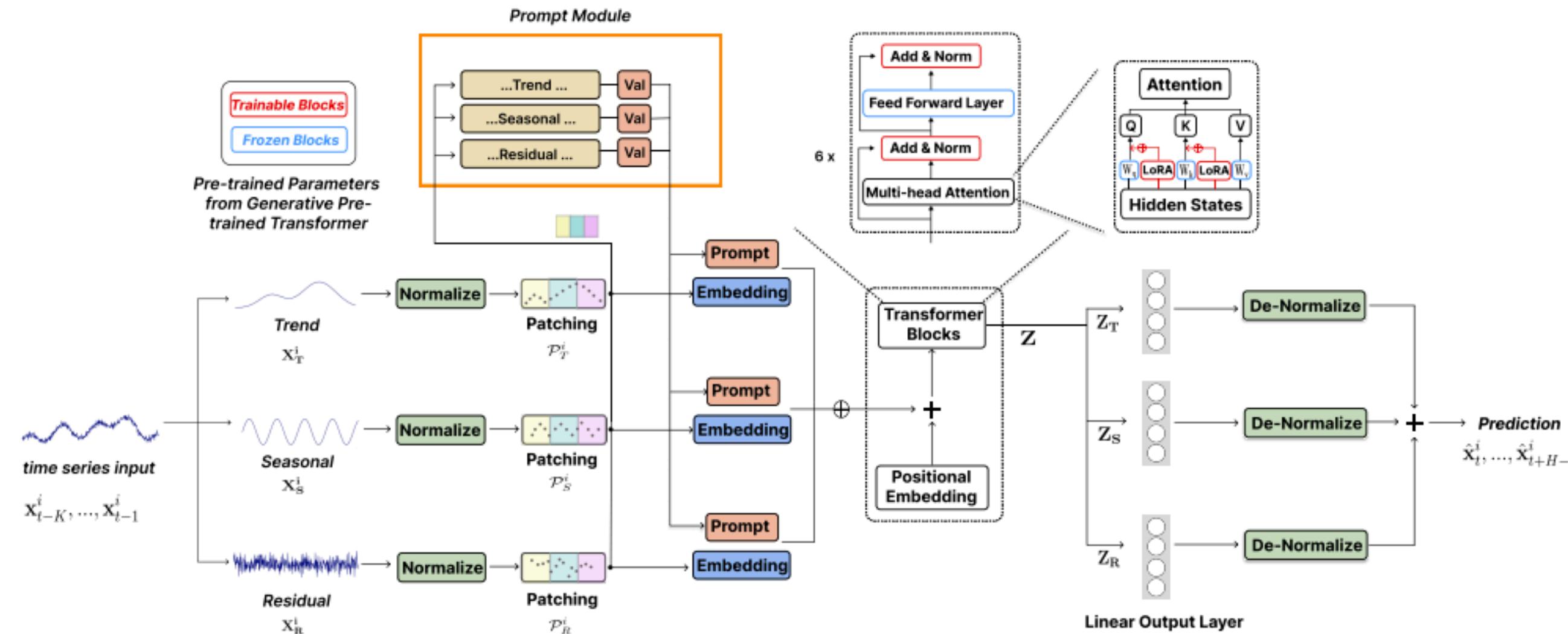


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.

TEMPO Overview

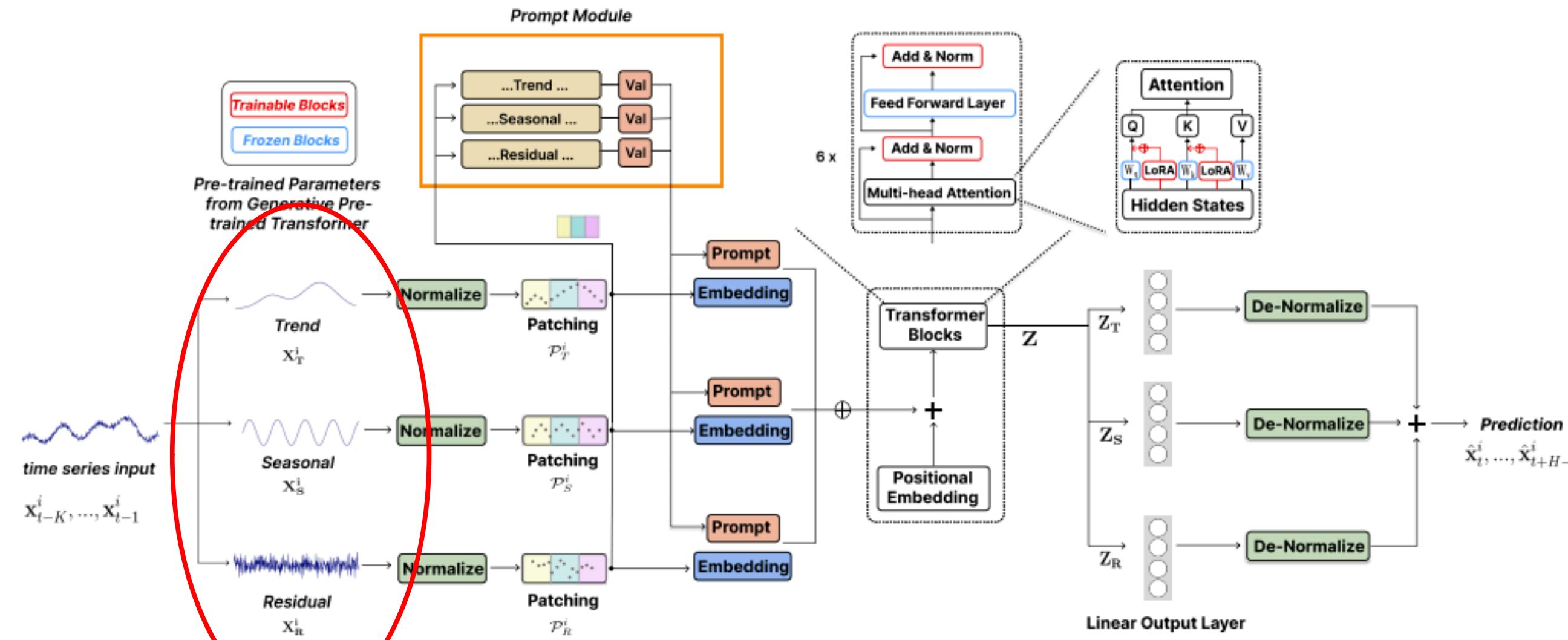
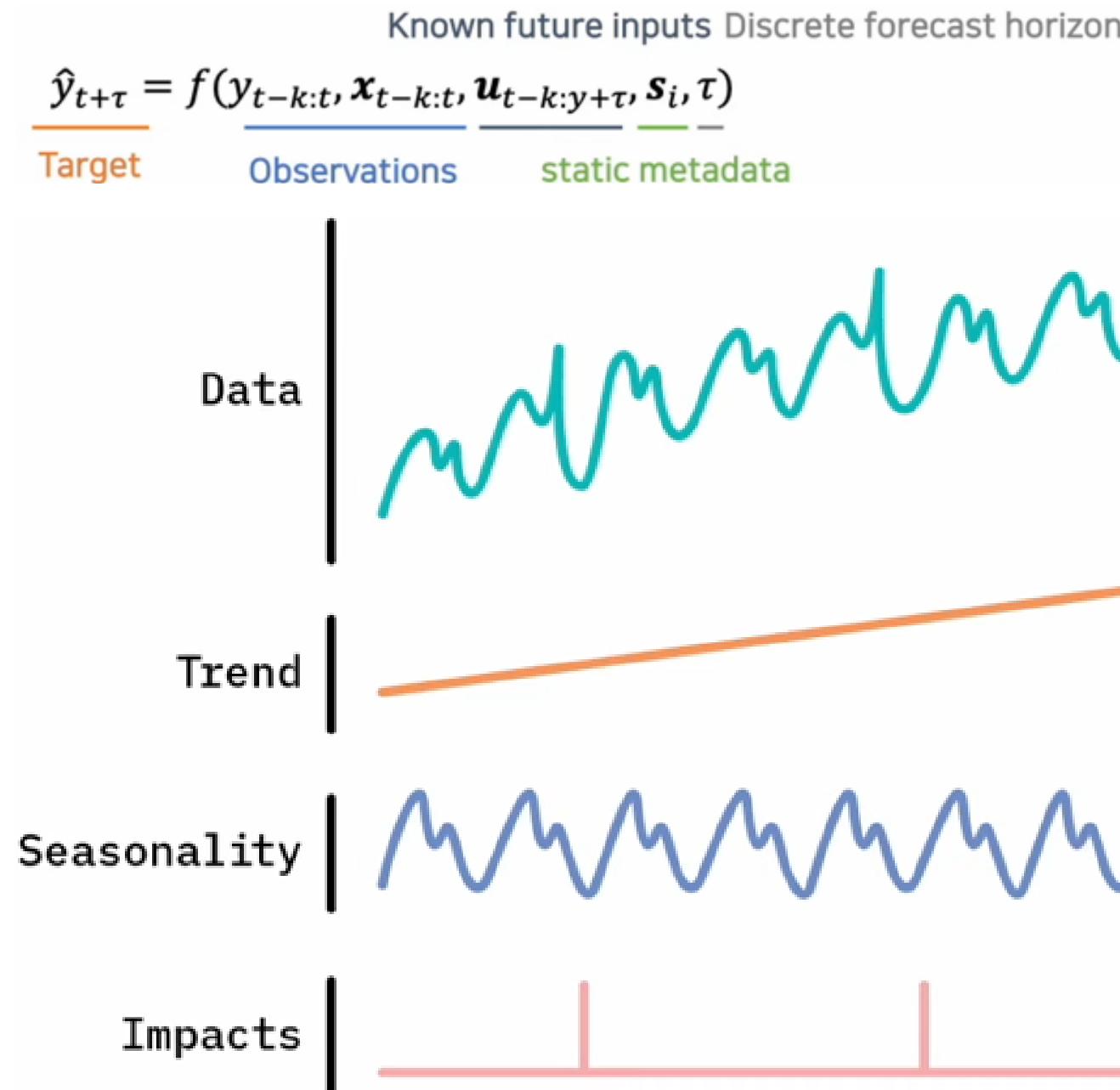


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.

▶ 시계열 데이터



시계열 데이터를 STL의 계절 및 추세 분해

- 추세(Trend): 시계열 데이터가 장기적으로 증가하거나 감소하는 것
- 계절성(Seasonality): 주, 월, 분기 단위 등 특정 시간의 주기로 나타나는 패턴
- 불규칙요소(Random, Residual): 설명될 수 없는 요인 또는 돌발적인 요인에 의해 일어나는 변화

TEMPO Overview

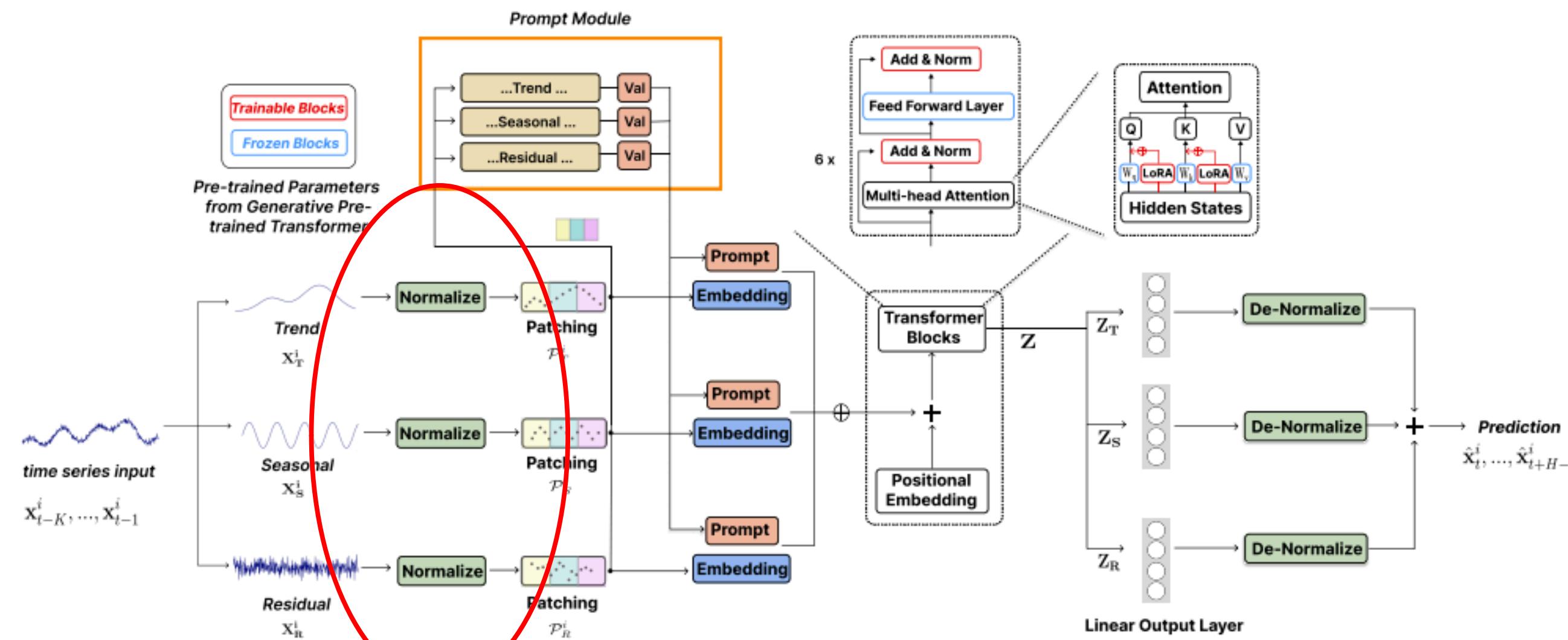
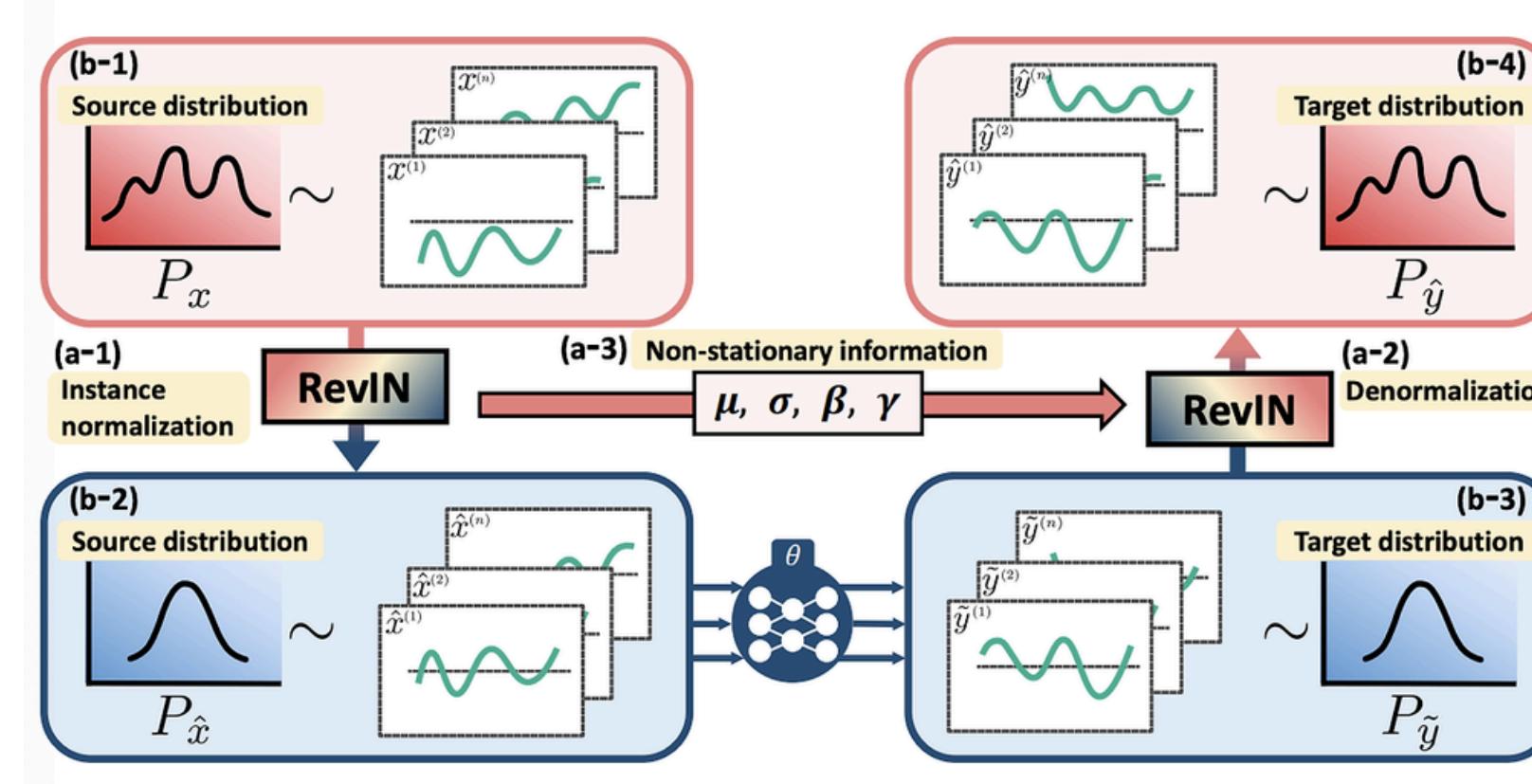


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.

▶ RevIN(Reversible Instance Normalization)

1. 데이터를 ‘일정한 크기’로 바꾸고(정규화), 모델이 더 쉽게 학습할 수 있게 함.
2. 모델의 결과 데이터를 다시 ‘원래 데이터의 크기와 분포’로 되돌림(역정규화)
3. 이때 단순히 평균과 표준편차로 돌리는 것이 아닌, 학습된 파라미터(β, γ)를 사용해 더 정확하고 유연하게 변환.



$$\mathbb{E}_t[x_{kt}^{(i)}] = \frac{1}{T_x} \sum_{j=1}^{T_x} x_{kj}^{(i)} \quad \text{Var}[x_{kt}^{(i)}] = \frac{1}{T_x} \sum_{j=1}^{T_x} (x_{kj}^{(i)} - \mathbb{E}_t[x_{kt}^{(i)}])^2$$

$$\hat{x}_{kt}^{(i)} = \gamma_k \left(\frac{x_{kt}^{(i)} - \mathbb{E}_t[x_{kt}^{(i)}]}{\sqrt{\text{Var}[x_{kt}^{(i)}] + \epsilon}} \right) + \beta_k,$$

$$\hat{y}_{kt}^{(i)} = \sqrt{\text{Var}[x_{kt}^{(i)}] + \epsilon} \cdot \left(\frac{\tilde{y}_{kt}^{(i)} - \beta_k}{\gamma_k} \right) + \mathbb{E}_t[x_{kt}^{(i)}].$$

TEMPO Overview

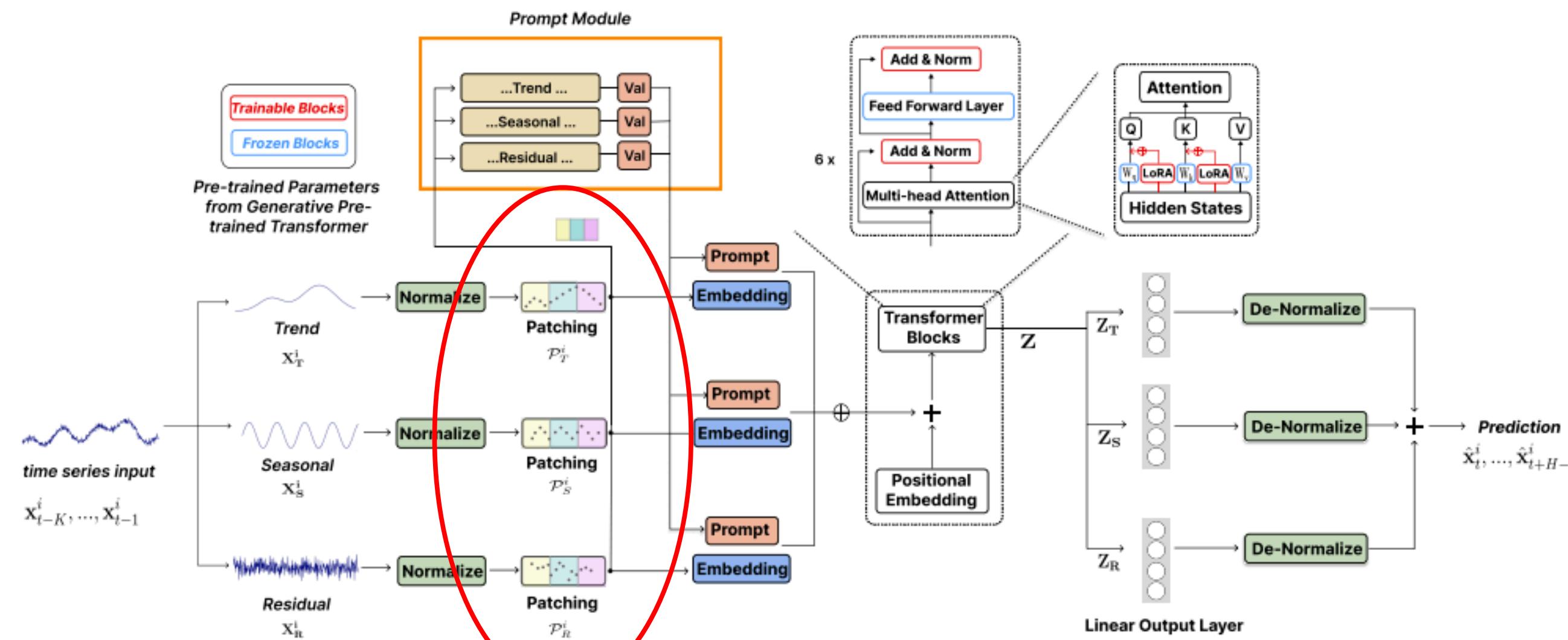
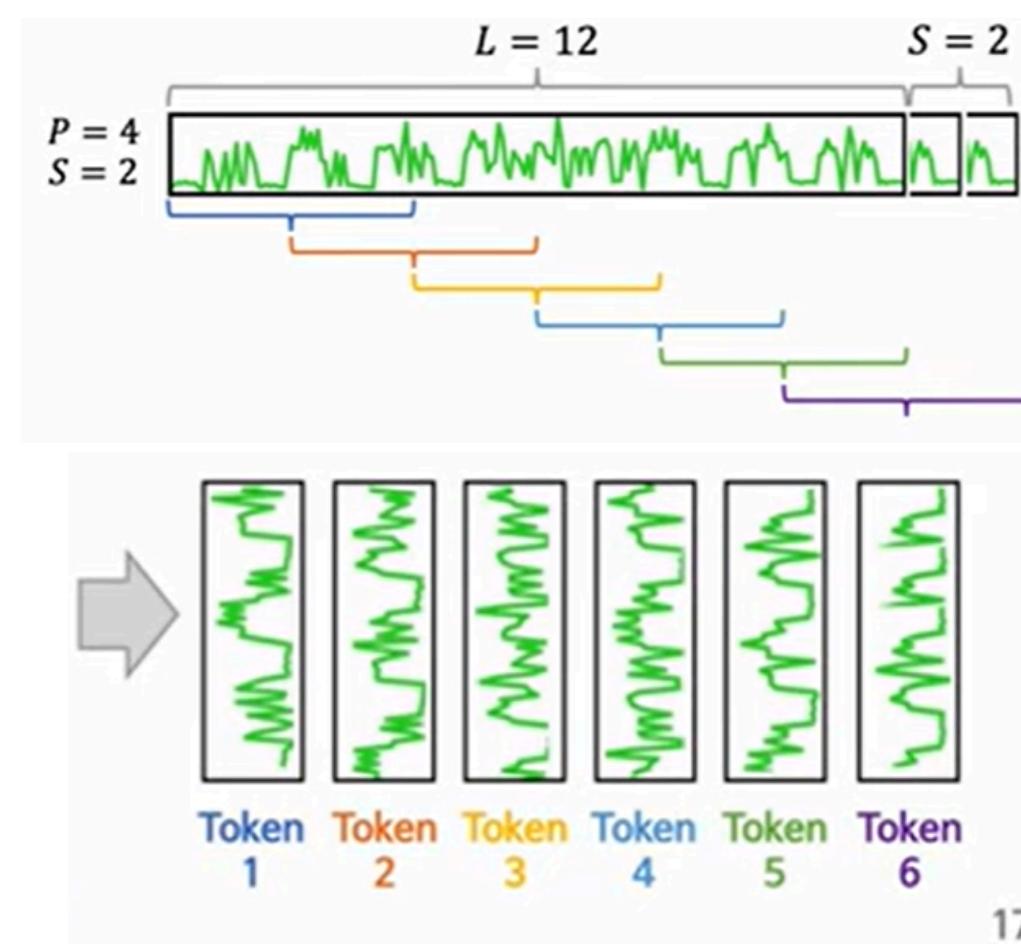


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.

Patching

Transformer 모델은 긴 문장을 처리할 때 토큰(Token)으로 나눠서 학습하고 TEMPO는 시계열 데이터를 Transformer에서 다루기 쉽게 sliding window 방식을 사용해 patching.



PatchTST에서 patching 도입의 장점

1. Point-wise time series 보다 많은 local semantic information 반영
2. Input token 개수의 감소로 인한 Transformer의 연산량과 메모리 사용량 감소
3. 동일한 GPU와 학습 시간 내에서 더 긴 길이의 sequence 활용 가능

TEMPO Overview

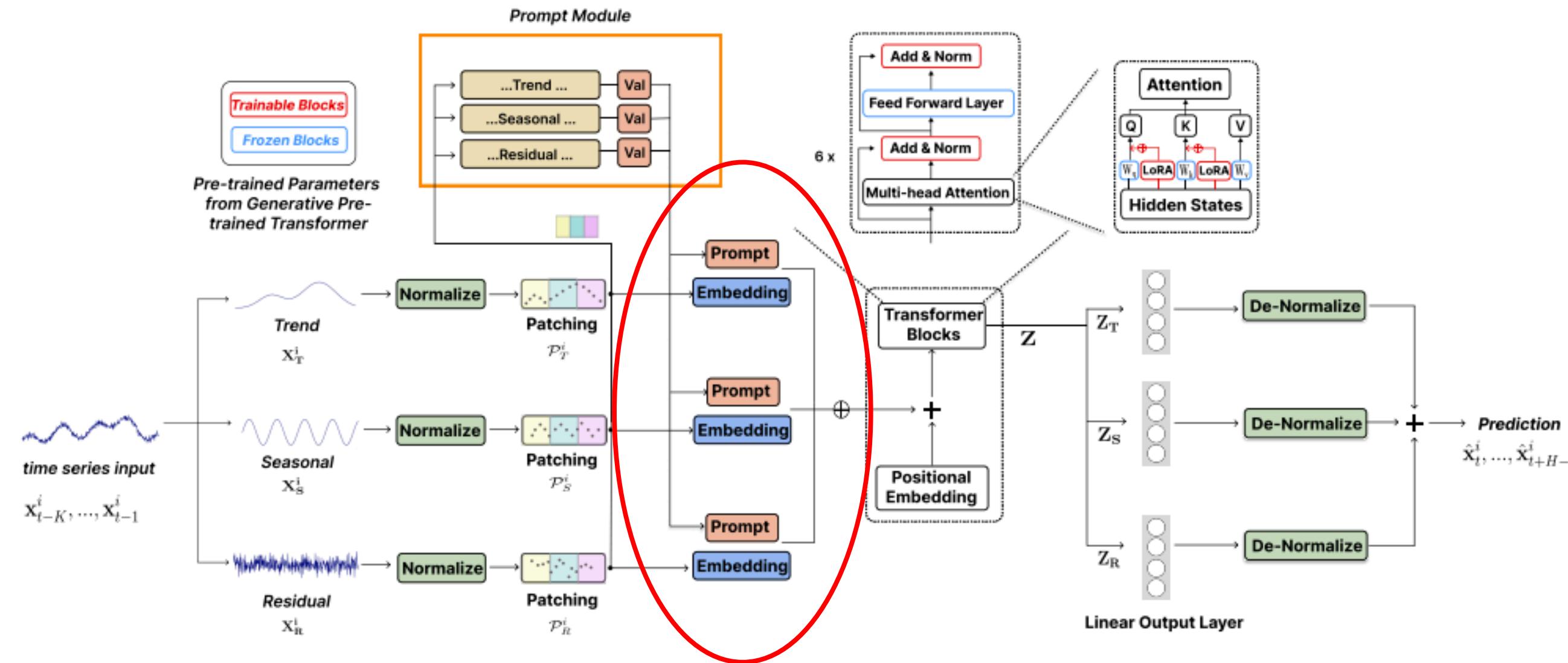


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.

▶ Embedding

패치된 시계열 데이터를 고차원 벡터로 변환

$P_T^i, P_S^i, P_R^i \in R^{L_p \times N}$ → 정규화(RevIN), patching 후의 각 요소들

$$N = \left[\frac{(L - L_P)}{S} \right] + 2 \rightarrow \text{패치 개수 계산식}$$

$P_T^i = f(P_T^i) \in R^{L_p \times L_E}$ → 언어모델이 이해할 수 있는 벡터로 변환

TEMPO Overview

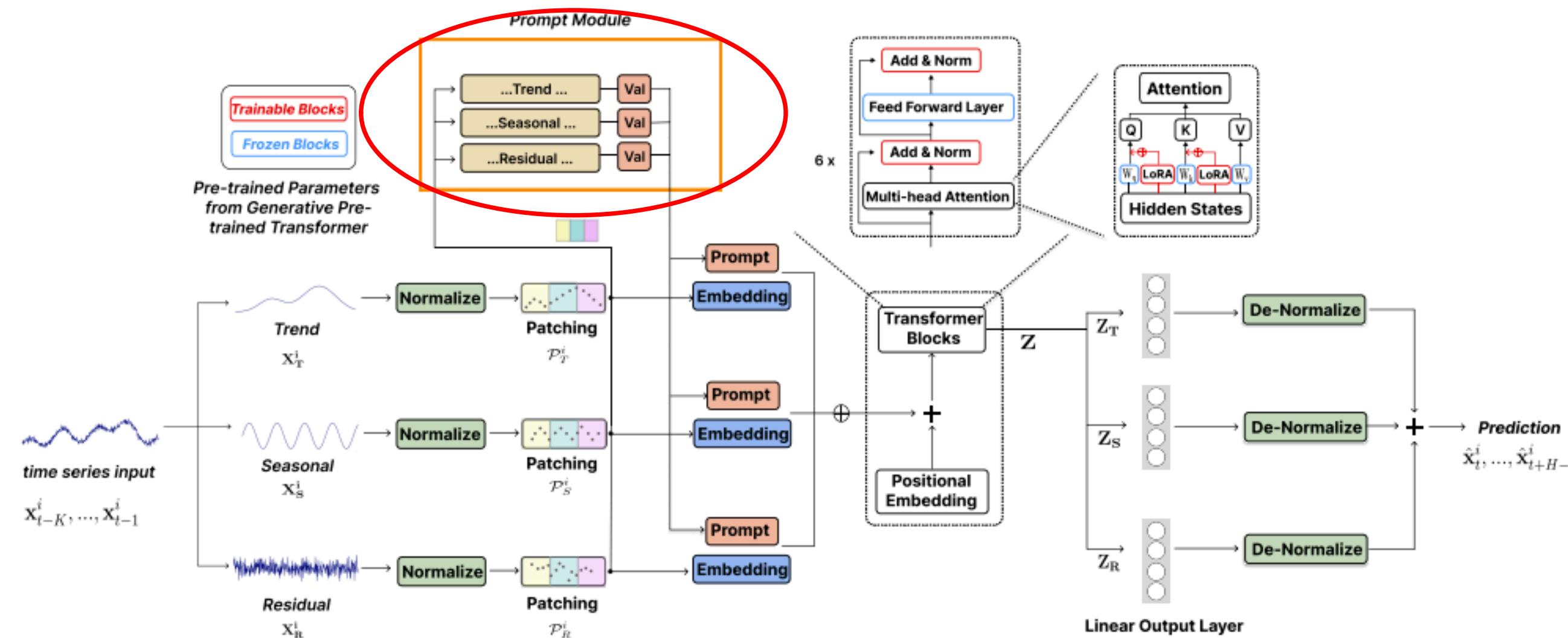
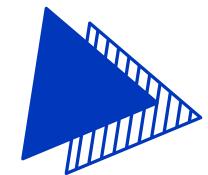


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.



Prompt Pool

고유한 key-value 쌍으로 저장된 prompt의 shared pool을 도입

- 유사한 입력 시계열이 pool에서 동일한 prompt 그룹을 검색하여 관련 과거 경험을 활용
- 다양한 데이터셋에서 학습된 패턴을 더 잘 인식하고 적용할 수 있게 되므로 모델링 효율성과 예측 성능을 향상시킬 수 있음.
- score 산출해 top k prompts를 시계열 데이터와 함께 input embedding으로 구성

Prompt key-value pairs

$$\mathbf{V_K} = \{(\mathbf{k}_1, V_1), (\mathbf{k}_2, V_2), \dots, (\mathbf{k}_M, V_M)\}$$

$$V_m \in \mathbb{R}^{L_p \times L_E} \quad \mathbf{k}_m \in \mathbf{K} = \{\mathbf{k}_m\}_{m=1}^M$$

Score matching function

$$\gamma(\mathcal{P}_T^i, \mathbf{k}_m) = \mathcal{P}_T^i \cdot \mathbf{k}_m / \|\mathcal{P}_T^i\| \|\mathbf{k}_m\|,$$

where $\gamma : \mathbb{R}^{L_E} \times \mathbb{R}^{L_E} \rightarrow \mathbb{R}$

Top k prompts 선택 후 input embedding

$$\mathbf{x}_T = [V_{s_1}; \dots; V_{s_K}; \mathcal{P}_T], \quad 1 \leq K \leq M$$

TEMPO Overview

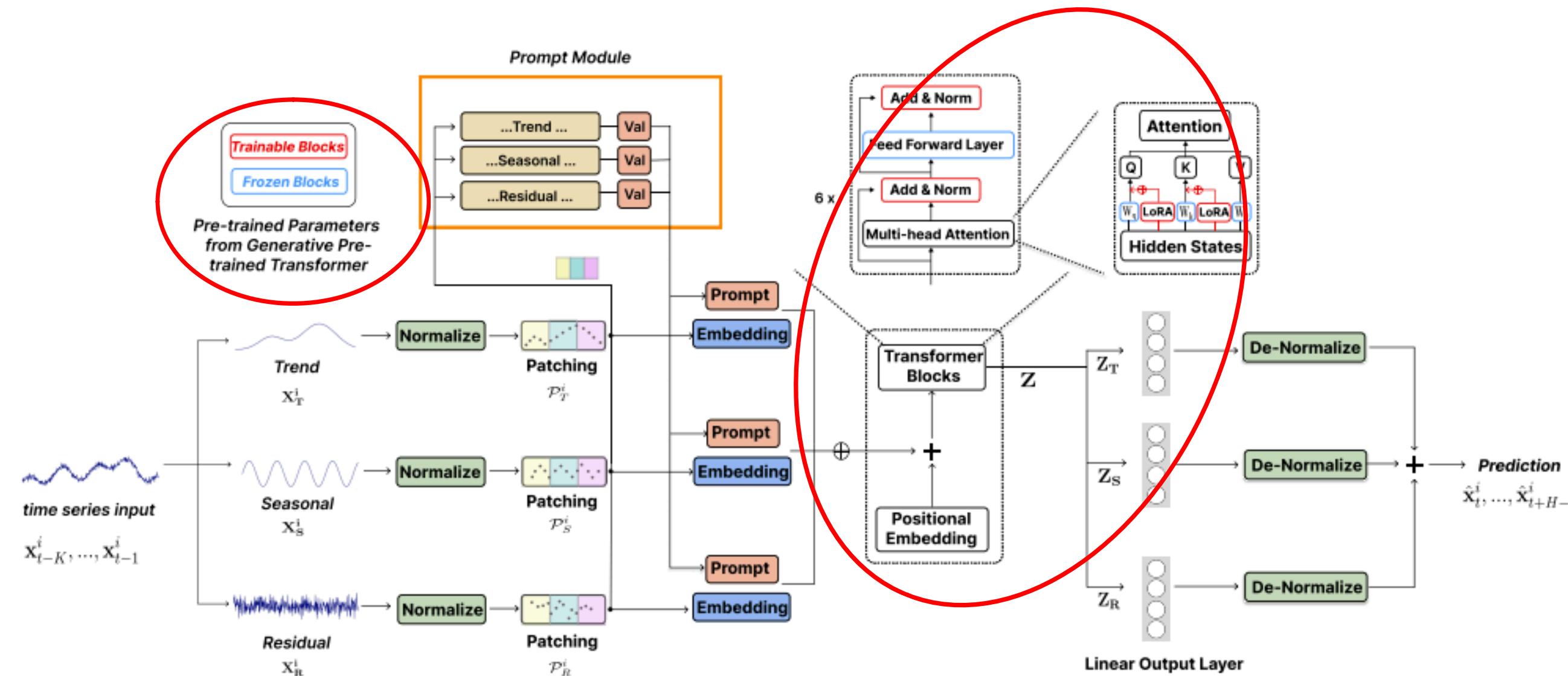


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.

GPT backbone

GPT가 문장의 흐름을 예측하는 것처럼, TEMPO는 GPT를 활용해 시계열 흐름을 예측

시계열 데이터에 맞게 조정

- 프롬프트와 다양한 시계열 요소(Trend, Seasonality, Residual)를 GPT 입력으로 변환
- 일부 구조를 Freeze하여 GPT의 지식을 유지

TEMPO Overview

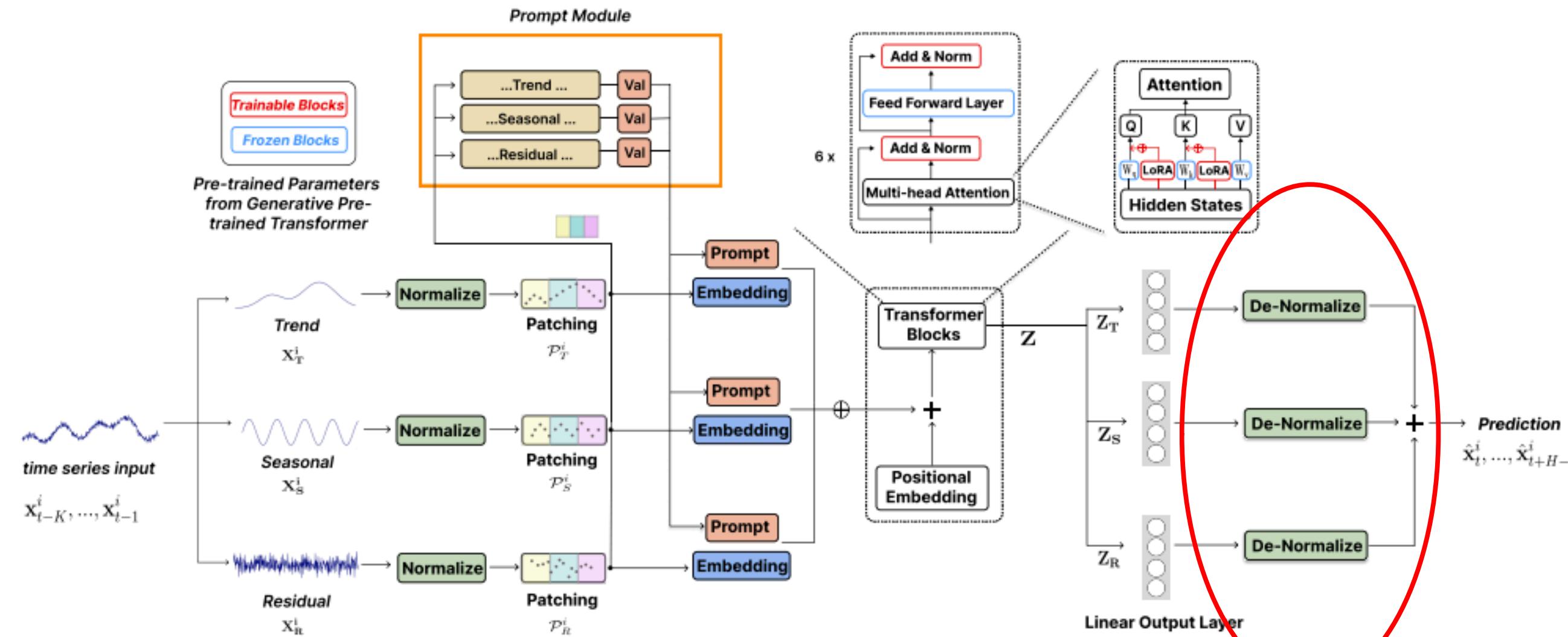


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.

TEMPO Overview

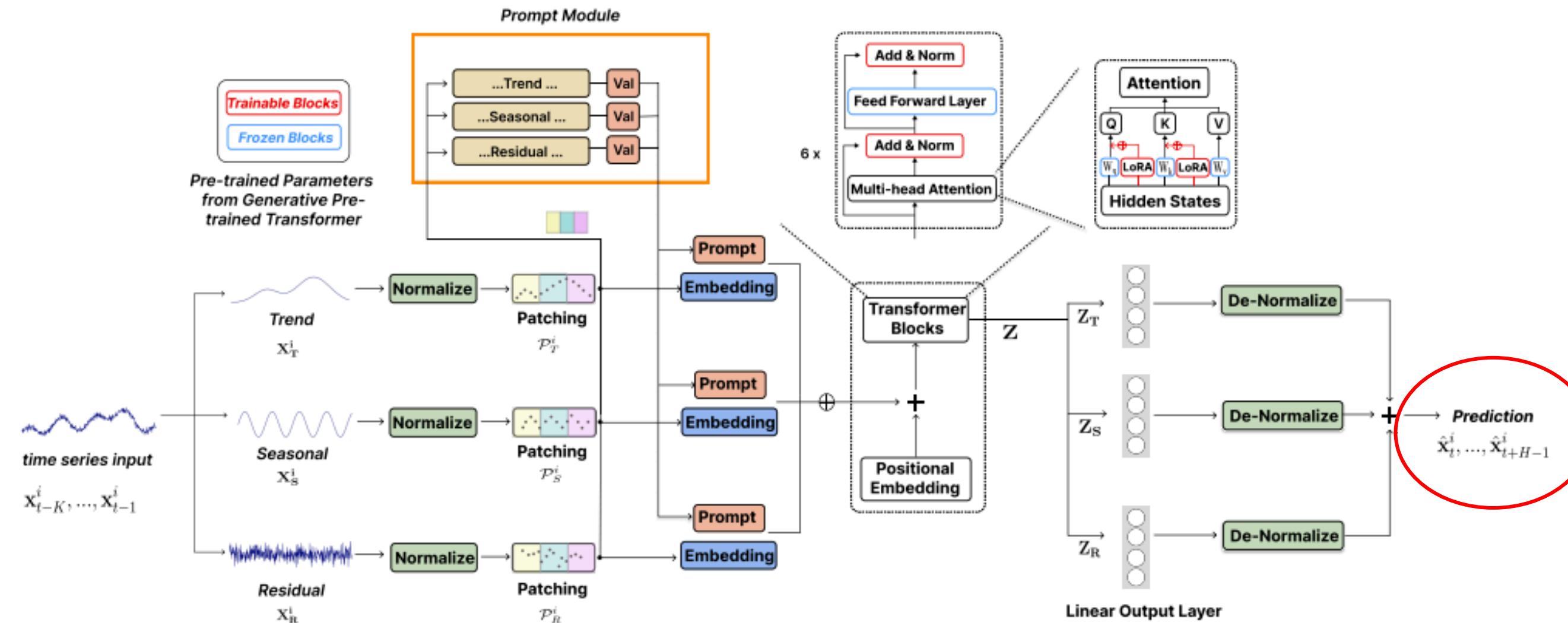
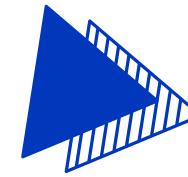


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.



Contextual Information(텍스트 데이터)

TEMPO 모델은 시계열 데이터의 복잡한 패턴을 분석하고 예측 정확도를 높이기 위해 STL 분해와 텍스트 데이터를 함께 사용

(1) STL 분해

- Trend: 장기적인 방향성
- Seasonal: 주기적인 패턴
- Residual: 예측할 수 없는 변동성

(2) Contextual Information

- 뉴스, 경제 보고서 등의 텍스트 데이터를 함께 입력
- GPT 기반의 텍스트 임베딩
 - 텍스트 벡터는 시계열 데이터와 함께 학습됨.

▶ Contextual Information 생성

Prompt:

Suppose you are living in {year}. Please summarize the most important news events from {year}-{month_name} related to {country} and their impact on the exchange rate between {currency1} and {currency2}. Provide a concise summary in 2-3 sentences.

OpenAI GPT API를 활용하여 환율 예측에 필요한 정보를 보완

출력 예시:

"February 2023: North Korea's missile tests increased geopolitical risks, leading to a weaker KRW. Additionally, a strong US job market reinforced expectations of Fed tightening, strengthening the USD."

▶ Contextual Information + 시계열 데이터

결합 과정

1. STL 분해를 통해 X_T , X_S , X_R 로 나눈 후 정규화(RevIN) 수행
2. 텍스트 데이터를 변환해 텍스트 임베딩 벡터(Text_embedding) 생성
3. 생성된 텍스트 임베딩과 STL 분해된 시계열 데이터를 함께 모델에 입력
4. GPT 기반의 Prompt Pool을 이용하여 모델이 데이터의 의미적 구조를 학습하도록 유도

$$x = \text{Text_embedding} \oplus X_T \oplus X_S \oplus X_R$$

TEMPO Overview

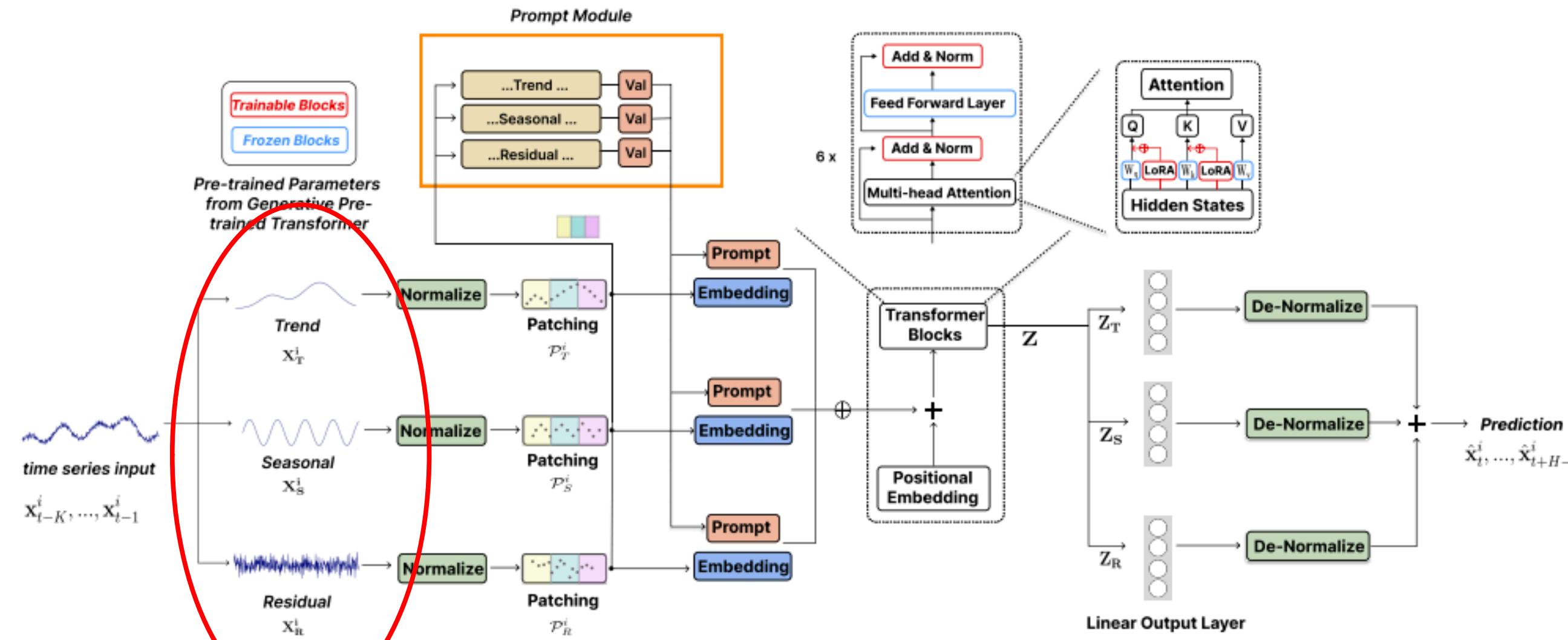
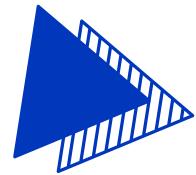


Figure 1: The architecture of proposed TEMPO-GPT. The trend X_T , seasonal X_S and residual X_R components are treated as different semantic inductive biases to feed into the pre-trained transformer.

SECTION 03

데이터 설명

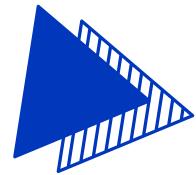


1. 환율 데이터

- 한국은행 경제통계시스템(ECOS)에서 원/미국 달러 환율 다운로드

The screenshot shows the '통계항목' (Statistical Item) section of the ECOS system. On the left, there is a search bar with placeholder text '검색어를 입력해주세요.' and a magnifying glass icon. Below it is a dropdown menu with various statistical items, including '2.4. 국민대차대조표(2020년 기준년)', '2.5. 국제수지표', '2.6. 국제투자대조표/대외채권채무', '3. 환율/통관수출입/외환보유액', '3.1. 환율', and '3.1.1. 일일환율'. Under '3.1.1. 일일환율', three specific items are listed: '3.1.1.1. 주요국 통화의 대원화환율', '3.1.1.2. 주요국 통화의 대미달러환율', and '3.1.1.3. 원화의 대미달러, 원화의 대위안/대엔 환율'. The first item is highlighted with a yellow background. On the right, there is another search bar and a list of currency pairs. The '계정항목' tab is selected, and the '원/미국달러(매매기준율)' option is checked and highlighted with a blue border. Other options include '원/위안(매매기준율)', '원/일본엔(100엔)', '원/유로', '원/독일마르크', '원/프랑스프랑', and '원/이태리리라(100리라)'. A '전체선택' (Select All) button is also present.

| 항목 | 설명 |
|----------------------------------|-------------------------|
| 3.1.1.1. 주요국 통화의 대원화환율 | 주요국 통화의 대원화환율 |
| 3.1.1.2. 주요국 통화의 대미달러환율 | 주요국 통화의 대미달러환율 |
| 3.1.1.3. 원화의 대미달러, 원화의 대위안/대엔 환율 | 원화의 대미달러, 원화의 대위안/대엔 환율 |



1. 환율 데이터

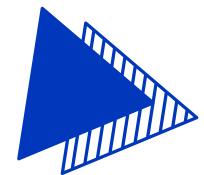
```
exchange_rate_df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/BITAmin/[2025-겨울] 프로젝트/2022-01-03~2022-01-20.csv')
exchange_rate_df['변환'] = pd.to_datetime(exchange_rate_df['변환'])
exchange_rate_df['원자료'] = exchange_rate_df['원자료'].str.replace(',', '').astype(float)
exchange_rate_df = exchange_rate_df.rename(columns={'변환': 'date', '원자료': 'exchange_rate'})
```



```
exchange_rate_df = exchange_rate_df.sort_values('date').reset_index(drop=True)
```

- CSV 파일에서 환율 데이터 불러오기
- 날짜 형식 변환 (pd.to_datetime)
- 데이터 정리 및 쉼표 제거 후 float 변환
- 정렬 및 인덱스 초기화

| | A | B |
|------|------------|----------|
| 1 | 변환 | 원자료 |
| 1720 | 2022-01-03 | 1,185.50 |
| 1721 | 2022-01-04 | 1,190.80 |
| 1722 | 2022-01-05 | 1,194.30 |
| 1723 | 2022-01-06 | 1,198.20 |
| 1724 | 2022-01-07 | 1,199.60 |
| 1725 | 2022-01-10 | 1,202.40 |
| 1726 | 2022-01-11 | 1,200.00 |
| 1727 | 2022-01-12 | 1,195.20 |
| 1728 | 2022-01-13 | 1,189.90 |
| 1729 | 2022-01-14 | 1,188.20 |
| 1730 | 2022-01-17 | 1,187.60 |
| 1731 | 2022-01-18 | 1,192.30 |
| 1732 | 2022-01-19 | 1,188.90 |
| 1733 | 2022-01-20 | 1,191.80 |



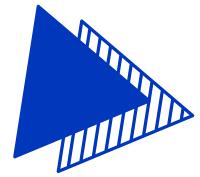
2. 뉴스 데이터

- OpenAI API 키 설정

```
monthly_news = {}  
country = "Korea"  
year_start = 2022  
currency1 = "KRW"  
year_end = 2024  
currency2 = "USD"
```

```
for year in tqdm(range(year_start, year_end + 1), desc="Year Progress"):  
    for month in range(1, 13):  
        month_name = f"{month:02d}"  
        prompt = f"""  
        Suppose you are living in {year}. Please summarize the most important news events from {year}-{month_name}  
        related to {country} and their impact on the exchange rate between {currency1} and {currency2}.  
        Provide a concise summary in 2-3 sentences.  
        """
```

- monthly_news: 뉴스 데이터를 저장할 딕셔너리
- year_start = 2022, year_end = 2024
: 2022년~2024년 뉴스 수집
- 한국(KRW) - 미국(USD): 환율 관련 뉴스 추출



2. 뉴스 데이터

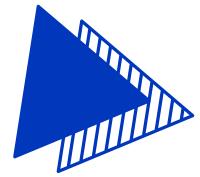
```
# API 호출 및 응답 받기
for attempt in range(3): # 최대 3번 재시도
    try:
        response = client.chat.completions.create(
            model="gpt-4-turbo",
            messages=[
                {"role": "system", "content": "You are an expert in financial markets."},
                {"role": "user", "content": prompt}
            ],
            temperature=0.7,
            max_tokens=200
        )

        # 응답 데이터 추출
        news_summary = response.choices[0].message.content.strip()
        break # 성공하면 루프 종료
    except Exception as e:
        print(f"⚠️ 오류 발생: {e}. 재시도 ({attempt + 1}/3)...")
        time.sleep(5) # 5초 대기 후 재시도
    else:
        news_summary = "Failed to retrieve data after multiple attempts."
# 데이터 저장
monthly_news[f"{year}-{month_name}"] = news_summary
print(f"✓ {year}-{month_name}: {news_summary}")

time.sleep(5) # API 요청 간격 유지
```

```
# 5. 뉴스 데이터를 JSON 파일로 저장
#####
news_json_path = "news_data_openai.json"
with open(news_json_path, "w", encoding="utf-8") as f:
    json.dump(monthly_news, f, indent=4, ensure_ascii=False)
```

- 최대 3번 재시도하여 응답 오류를 방지
- 수집된 뉴스 데이터 `monthly_news` 딕셔너리에 저장
- 수집된 뉴스 데이터를 JSON 파일로 저장



2. 뉴스 데이터

• GPT-4 API가 월별로 요약한 원달러 환율 관련 뉴스

- ✓ 2022-02: In February 2022, geopolitical tensions in Eastern Europe, specifically the Russia-Ukraine conflict, ha
 - ✓ 2022-03: In March 2022, North Korea's missile tests significantly affected geopolitical tensions in the Korean P
 - ✓ 2022-04: In April 2022, South Korea faced several impactful events that influenced the KRW-USD exchange rate. Fi
 - ✓ 2022-05: In May 2022, geopolitical tensions on the Korean Peninsula escalated as North Korea conducted a series
 - ✓ 2022-06: In June 2022, geopolitical tensions on the Korean Peninsula, particularly surrounding North Korea's mis
 - ✓ 2022-07: In July 2022, tensions on the Korean Peninsula escalated due to North Korea's missile tests, which incl
 - ✓ 2022-08: In August 2022, several key events in Korea influenced the KRW-USD exchange rate. Tensions surrounding I
 - ✓ 2022-09: In September 2022, the geopolitical tensions on the Korean Peninsula escalated, particularly due to Nor
 - ✓ 2022-10: In October 2022, North Korea's missile tests significantly influenced geopolitical tensions in the Kore
 - ✓ 2022-11: In November 2022, South Korea faced several key events that impacted the KRW-USD exchange rate. Tension
 - ✓ 2022-12: In December 2022, significant events in Korea included the announcement of increased fiscal spending by
- Year Progress: 33% | | 1/3 [01:43<03:27, 103.85s/it] 2023-01: In January 2023, the geopolitical tension
- ✓ 2023-02: As of February 2023, tensions on the Korean Peninsula have influenced the exchange rate between the Kor
 - ✓ 2023-03: As of March 2023, tensions in the Korean Peninsula escalated due to North Korea's continued missile tes
 - ✓ 2023-04: As of 2023, significant developments in Korea include heightened geopolitical tensions on the Korean Pe
 - ✓ 2023-05: As of May 2023, the significant news events in Korea primarily revolved around geopolitical tensions an
 - ✓ 2023-06: As of June 2023, tensions escalated on the Korean Peninsula following North Korea's renewed missile tes
 - ✓ 2023-07: As of July 2023, tensions on the Korean Peninsula influenced the KRW-USD exchange rate significantly. N
 - ✓ 2023-08: As of August 2023, heightened geopolitical tensions on the Korean Peninsula, particularly due to North I

SECTION 04

TEMPO

▶ 주요 코드

[STEP 1] GitHub 저장소 클론

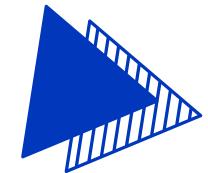
```
[ ] !git clone https://github.com/DC-research/TEMPO.git
```

```
→ Cloning into 'TEMPO'...
remote: Enumerating objects: 263, done.
remote: Counting objects: 100% (93/93), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 263 (delta 69), reused 49 (delta 49), pack-reused 170 (from 1)
Receiving objects: 100% (263/263), 1.88 MiB | 13.37 MiB/s, done.
Resolving deltas: 100% (140/140), done.
```

```
▶ import pandas as pd
import numpy as np
from datetime import datetime, timedelta
import google.generativeai as genai
import time
from sklearn.preprocessing import MinMaxScaler
import torch
import os
import matplotlib.pyplot as plt
```

- TEMPO GitHub 저장소 클론

- 필요한 라이브러리 불러오기



주요 코드

[STEP 2] 수집한 데이터 저장 및 정규화

```
# Min-Max 정규화 수행
scaler = MinMaxScaler()
exchange_rate_df['scaled_rate'] = scaler.fit_transform(exchange_rate_df[['exchange_rate']])

# 환율 데이터 저장 (exchange_rate.csv)
exchange_rate_save_path = f'{data_dir}/exchange_rate.csv'
exchange_rate_df[['date', 'scaled_rate']].to_csv(exchange_rate_save_path, index=False)

print(f"✓ 환율 데이터가 '{exchange_rate_save_path}' 파일로 저장되었습니다.")
```

- Min-Max 정규화 수행하여 0~1 범위로 변환
- CSV 파일로 저장

[STEP 3] TEMPO 모델로 시계열 예측 수행

```
# 예측 수행
with torch.no_grad():
    predicted_values = model.predict(input_data, pred_length=365)
```

- **torch.no_grad()**

: PyTorch의 그래디언트 연산 비활성화
(메모리 절약 및 속도 향상)

: 모델이 학습이 아닌 예측만 수행하는 단계

- **model.predict(input_data,pred_length=365)**

: input_data를 기반으로 향후 365일 환율 예측

▶ 주요 코드

[STEP 3] TEMPO 모델로 시계열 예측 수행

```
# 예측값 범위를 0~1 사이로 강제 제한하여 비정상적인 값 방지  
predicted_values = np.clip(predicted_values, 0, 1)
```

- 예측값이 0~1 사이를 벗어나지 않도록 강제 조정
- 시계열 예측 모델이 훈련 중 이상값을 학습했을 경우를 대비

```
# MinMaxScaler를 기준보다 넓은 범위로 적용하여 비정상적인 변환 방지  
scaler = MinMaxScaler(feature_range=(0, 1))  
scaler.fit(exchange_rate_df[['exchange_rate']])  
predicted_exchange_rates = scaler.inverse_transform(predicted_values.reshape(-1, 1)).flatten()
```

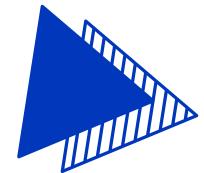
- MinMaxScaler를 적용하여 예측된 정규화 데이터(0~1 범위)를 실제 환율 값으로 변환
- `scaler.inverse_transform(...)`을 사용하여 정규화 이전의 원래 값으로 되돌림

▶ 주요 코드

[STEP 3] TEMPO 모델로 시계열 예측 수행

```
# 최근 변동성을 반영한 랜덤 노이즈 추가 (강도를 낮춤)
recent_mean = exchange_rate_df['exchange_rate'].iloc[-30:].mean()
recent_std = exchange_rate_df['exchange_rate'].iloc[-30:].std()
predicted_exchange_rates += np.random.normal(0, recent_std * 0.1, len(predicted_exchange_rates))
```

- 모델이 예측한 값은 완전히 고정된 값이므로 실제 시장의 변동성을 반영하기 위해 노이즈를 추가
 - : 최근 30일 동안의 환율 평균 값을 구함
 - : 최근 30일 동안의 환율 표준 편차(변동성)를 구함
 - : 변동성을 반영한 랜덤 노이즈(정규분포 기반) 추가
 - : 표준 편차의 10% 수준의 변동을 부여하여 급격한 변동을 방지

 주요 코드

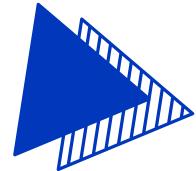
[STEP 3] TEMPO 모델로 시계열 예측 수행

```
# 변동성 범위 조정 ( $\pm 40\%$ 로 축소)
predicted_exchange_rates = np.clip(predicted_exchange_rates,
                                     recent_mean * (1 - 0.4),
                                     recent_mean * (1 + 0.4))

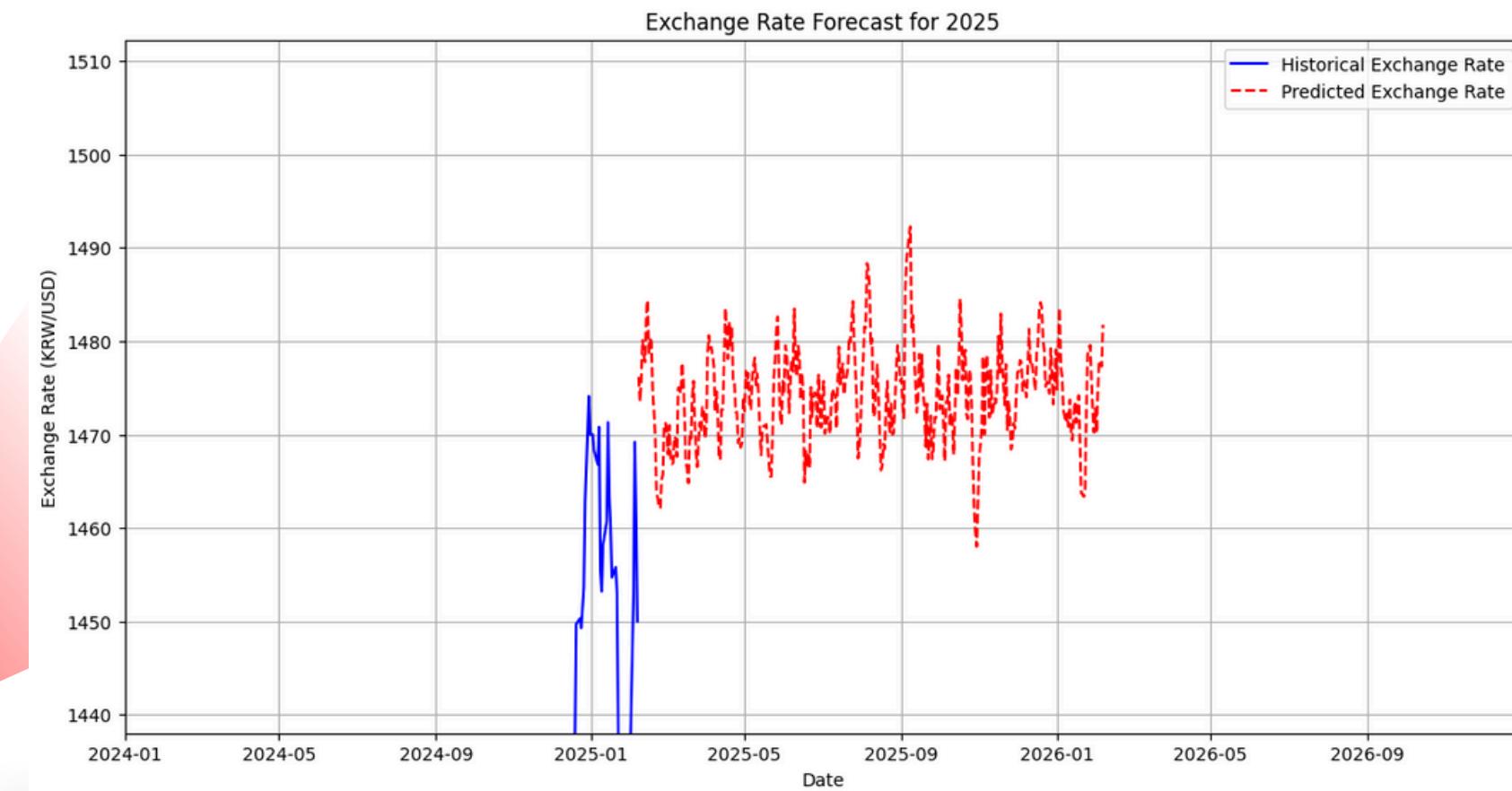
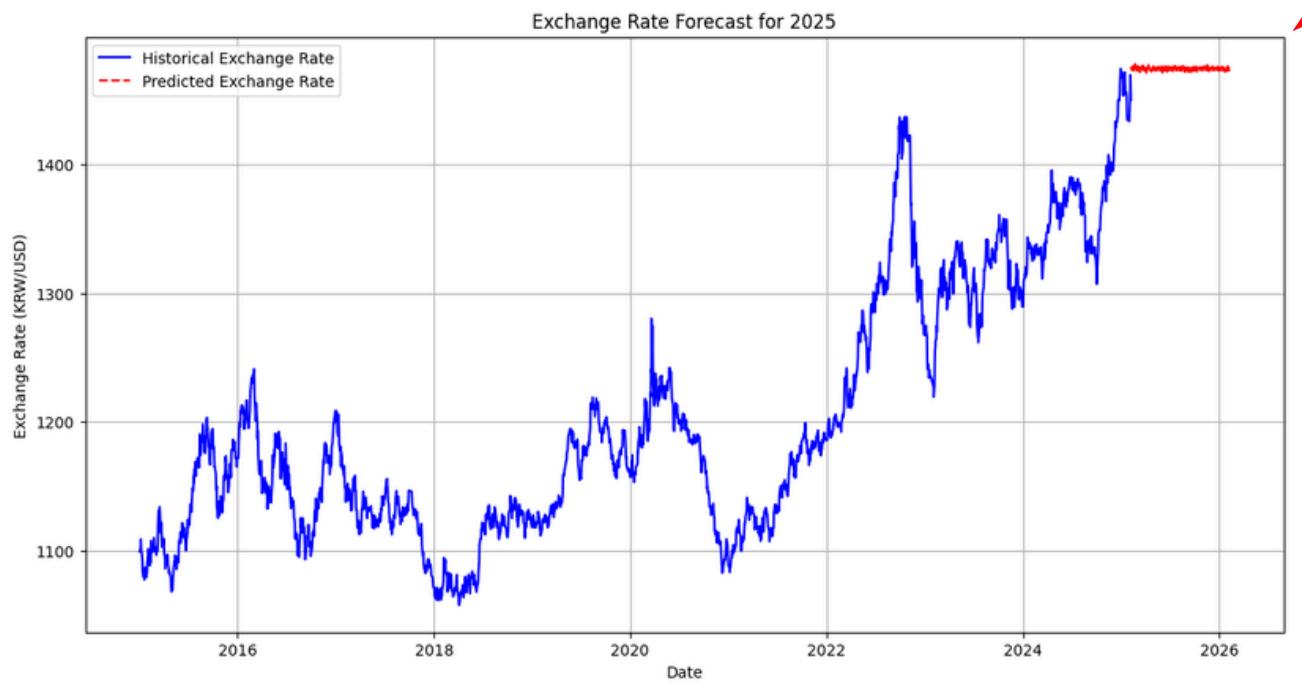
# 최종 예측값 확인
print("Final Adjusted Predicted Values (First 5):")
print(predicted_exchange_rates[:5])
```

- 모델이 예측한 값이 현실적인 범위를 벗어나지 않도록 보정
- $\text{recent_mean} * (1 - 0.4) \sim \text{recent_mean} * (1 + 0.4)$
 - : 최근 평균 환율의 $\pm 40\%$ 범위 내에서 값 유지.
 - : 급격한 환율 급등/급락을 방지하여 보다 현실적인 예측값을 얻음

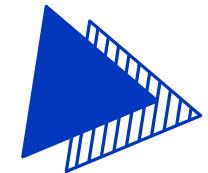
Predicted Values (Before Clipping) – Min/Max Check:
Min: 2.7336137294769287, Max: 10.939007759094238
Final Adjusted Predicted Values (First 5):
[1475.1439 1473.5558 1475.3884 1474.4551 1474.339]



결과(2025년 환율 예측)



- 2025년 환율이 **1400원대 후반**에서만 변동할 것으로 예측
- 최근 환율이 급등한 추세를 유지하려는 경향이 강함.
- 뉴스 데이터가 ‘한국 환율이 지속적으로 약세’를 보일 것 전망
→ 경향이 예측 결과에 반영되었을 가능성



결과(2025년 환율 예측)

저번주부터 실제 환율과 비교

| date | predicted_exchange_rate | 실제값 | predicted_exchange_rate - 실제값 |
|---------------------|-------------------------|--------|-------------------------------|
| 2025-02-07 00:00:00 | 1473.925537 | 1447.1 | 26.82553711 |
| 2025-02-08 00:00:00 | 1471.008789 | - | - |
| 2025-02-09 00:00:00 | 1472.708252 | - | - |
| 2025-02-10 00:00:00 | 1475.915894 | 1447.4 | 28.51589355 |
| 2025-02-11 00:00:00 | 1474.18103 | 1453.4 | 20.78103027 |
| 2025-02-12 00:00:00 | 1472.685547 | 1453.5 | 19.18554688 |
| 2025-02-13 00:00:00 | 1474.670654 | 1452.6 | 22.0706543 |
| 2025-02-14 00:00:00 | 1474.651001 | 1450.5 | 24.15100098 |
| 2025-02-15 00:00:00 | 1471.555176 | - | - |
| 2025-02-16 00:00:00 | 1473.843872 | - | - |
| 2025-02-17 00:00:00 | 1475.730591 | 1441.2 | 34.53059082 |
| 2025-02-18 00:00:00 | 1474.115601 | 1440.7 | 33.41560059 |
| 2025-02-19 00:00:00 | 1474.150146 | 1440.9 | 33.25014648 |
| 2025-02-20 00:00:00 | 1474.044678 | 1438.2 | 35.84467773 |

- 평균 오차: 약 27.66원
- 단기 외환 트레이딩 측면: 오차(19~35원)로 인해, 몇 원의 변동성도 영향이 큰 실시간 외환 시장에서 활용하기에 부족
- 장기적인 환율 전망 리포트 측면: 변동성을 과소평가하는 경향이 있어 조정 필요하나, 환율 방향성은 어느 정도 맞출 수 있어 일부 활용 가능
- 기업 환리스크 관리 측면: 기업이 몇 개월 뒤의 환율을 참고 하는 용도로 활용 가능

TMEPO 논문 저자와의 컨택

Defu Cao(University of Southern California)

Inquiry About the TEMPO from a Korean University Student



Dear Dr. Cao,

I hope this message finds you well. My name is Chaeyeon Lee, a student at Sookmyung Women's University in South Korea. I am currently studying big data as a member of BITAmin, a university big data club in Korea.

It is truly an honor to reach out to you. First, I would like to thank you for sharing such detailed and valuable resources on your GitHub. Your work has been incredibly helpful for my studies, and I truly appreciate the effort you put into making your research accessible.

I am writing to you because my team and I are very interested in your paper, "TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting." We are currently studying the paper and utilizing your GitHub repository to conduct a related project. Specifically, we are applying the TEMPO model to forecast the KRW-USD exchange rate for 2025, using South Korean news text data and historical exchange rate data. This idea was inspired by your work with the TETS dataset.

However, during the implementation, we encountered some challenges. Despite multiple attempts and code modifications, we are getting either unrealistically high exchange rate predictions or the same repeated values. I have attached the code that produced predicted exchange rates consistently around 1,474 KRW, along with the exchange rate dataset we used. If you could spare a moment to review it, even a brief suggestion or advice would be tremendously helpful.

We understand you must be very busy with your research, but even a short reply would mean a great deal to us. As students, some of us from non-technical backgrounds, we recognize that there may be many shortcomings in our code, but we are eager to learn and improve through this experience. Your insights would greatly contribute to our understanding.

Lastly, I would like to express my deep respect for your impressive research. Thank you once again for your time and consideration.

Best regards,
Chaeyeon Lee

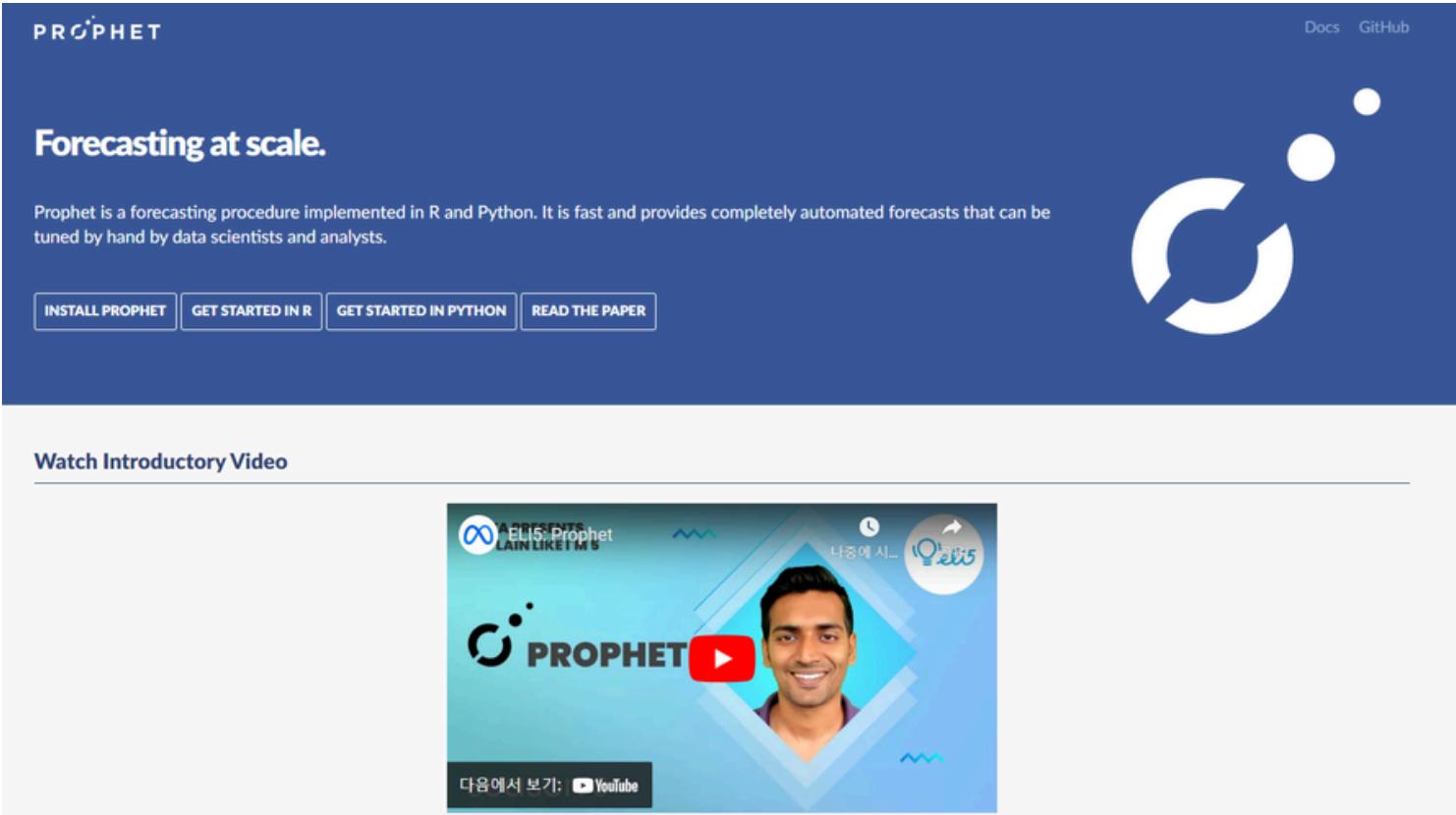
첨부파일 2개 • Gmail에서 스캔함 ①



SECTION 05

PROPHET

모델 설명



참고 링크 : <https://facebook.github.io/prophet/>

- 시계열 데이터의 예측을 위해 Facebook에서 개발한 오픈소스 라이브러리
- 자동화된 이상 탐지 및 계절성 분석 지원
- 특히, 매출, 사용자 수, 웹 트래픽과 같은 비즈니스 예측에 유용

모델 설명



1. 추세 변동

선형 및 비선형 추세 모델링을 지원하여 장기적인 추세를 모델링할 수 있다.

2. 계절성

연간 계절성과 주간 계절성을 포함하며 데이터의 주기적인 변동을 모델링할 수 있다.

3. 휴일 효과

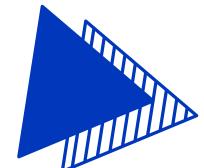
공휴일 및 특별 이벤트에 대한 효과를 반영할 수 있다.

4. 결측치 처리

자동으로 결측치를 처리하고 불규칙한 시계열 데이터를 다룰 수 있다.

5. 이상치 처리

이상치에 강건한 모델링을 제공한다.



주요 코드

[STEP 1] 데이터 준비

```
import pandas as pd

data = pd.read_csv("/content/drive/MyDrive/BITAmin/Project/시계열1조/주요국 통화의 대원화환율_06183901.csv")
data['변환'] = pd.to_datetime(data['변환']).dt.normalize()
data['원자료'] = data['원자료'].str.replace(',', '').astype(float)
data.columns = ['ds', 'y']
data.head()
```

| | ds | y |
|---|------------|--------|
| 0 | 2015-01-02 | 1099.2 |
| 1 | 2015-01-05 | 1102.6 |
| 2 | 2015-01-06 | 1108.7 |
| 3 | 2015-01-07 | 1103.9 |
| 4 | 2015-01-08 | 1099.7 |

- Prophet에서 요구하는 데이터 형식에 맞춰 datetime을 ds로, value를 y로 설정

[STEP 2] Prophet 모델 학습 및 예측

```
!pip install prophet

import pandas as pd
from prophet import Prophet

m = Prophet()
m.fit(data)
future = m.make_future_dataframe(periods=365)

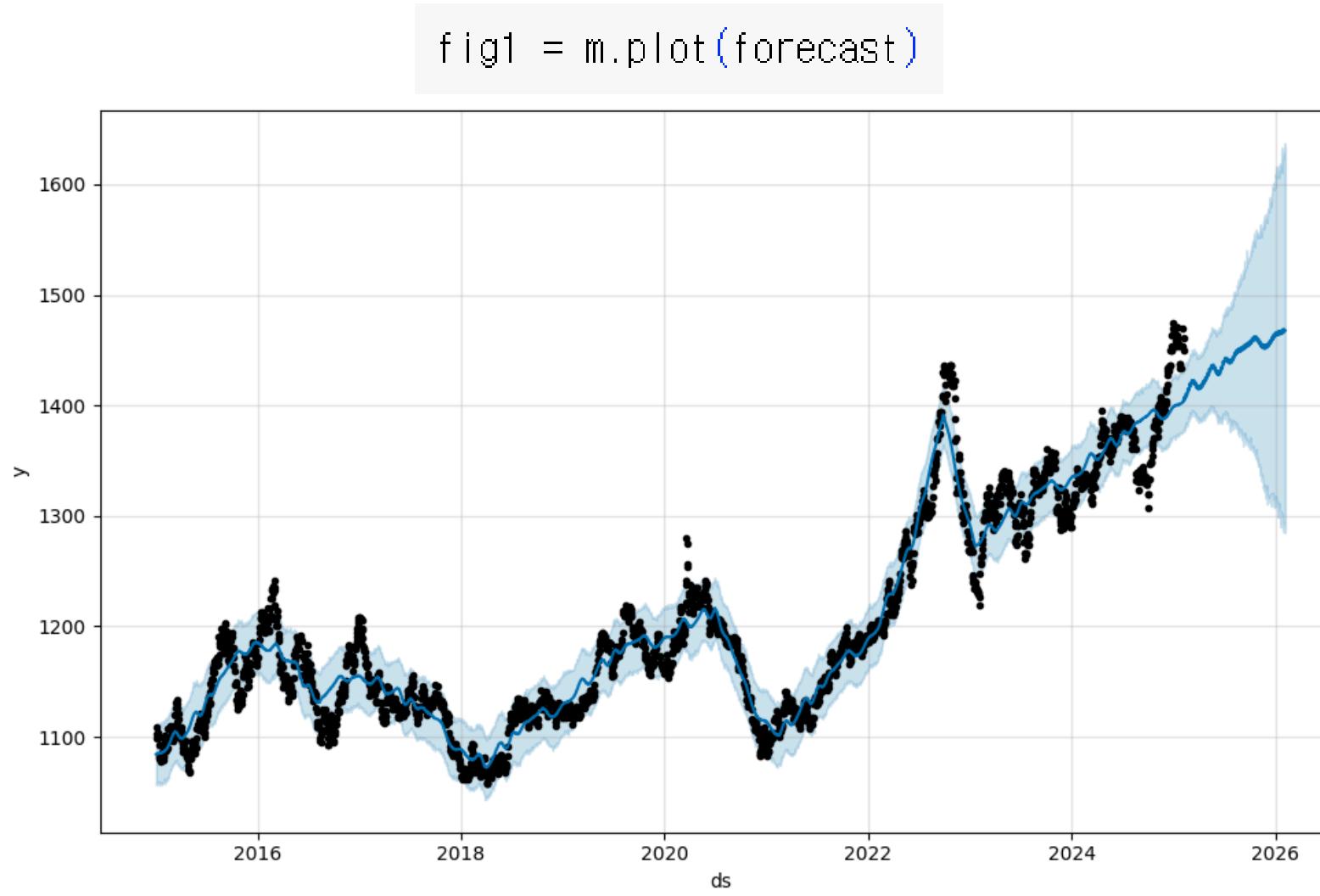
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

| | ds | yhat | yhat_lower | yhat_upper |
|------|------------|-------------|-------------|-------------|
| 2850 | 2026-02-02 | 1466.403354 | 1287.509718 | 1635.768130 |
| 2851 | 2026-02-03 | 1467.295836 | 1293.491868 | 1637.493413 |
| 2852 | 2026-02-04 | 1467.201015 | 1290.564101 | 1638.496207 |
| 2853 | 2026-02-05 | 1467.422639 | 1285.364472 | 1635.950124 |
| 2854 | 2026-02-06 | 1467.739371 | 1291.184848 | 1639.883445 |

- Prophet 모델 불러온 후, 시계열 데이터를 학습하고 향후 1년 동안의 값을 예측

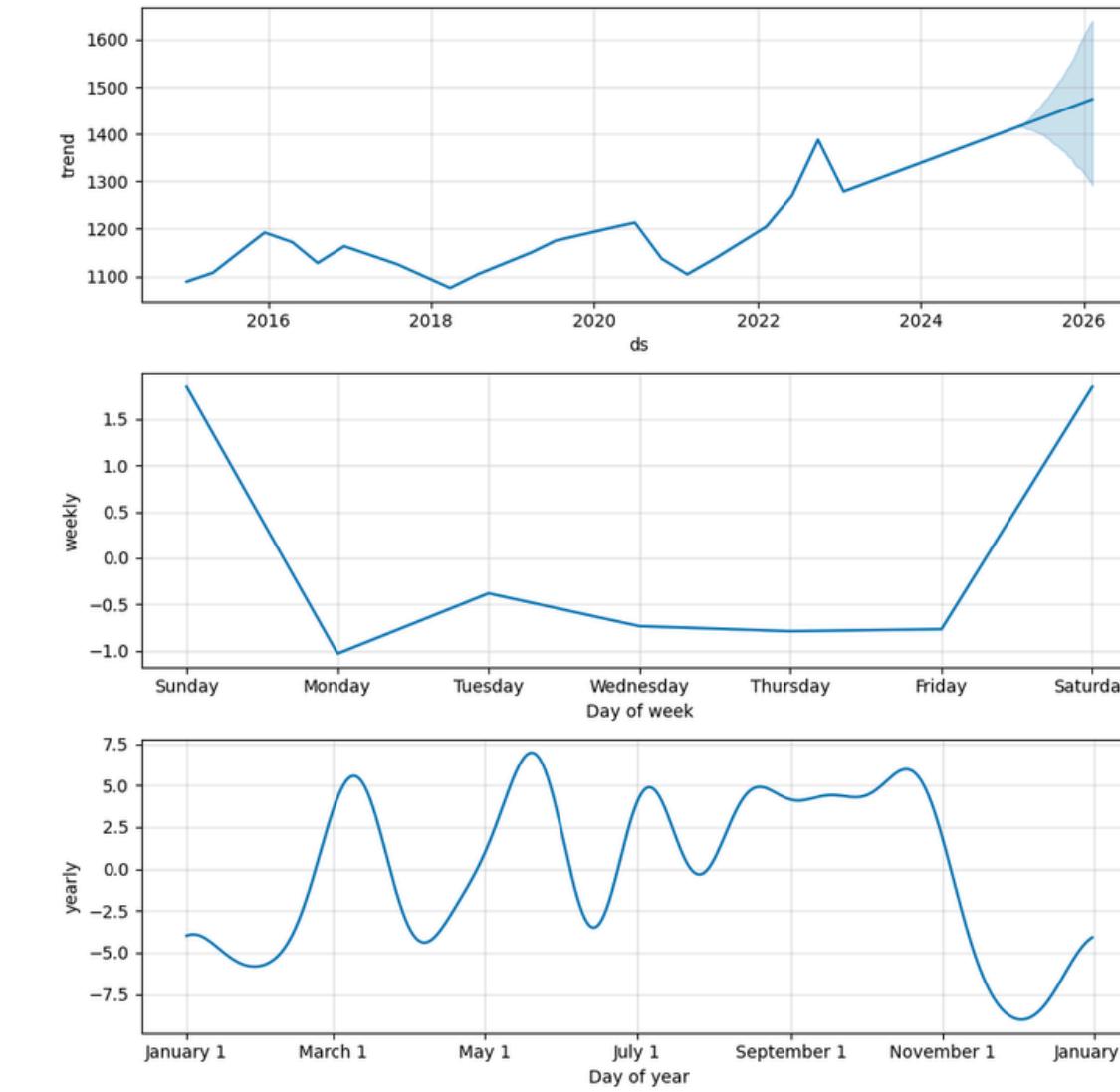
▶ 주요 코드

[STEP 3] 예측 결과 시각화



- 실제값을 검정점, 예측값을 파란선으로 신뢰구간과 함께 시각화

```
fig2 = m.plot_components(forecast)
```



- Prophet이 예측한 주요 구성 요소
(트렌드, 주기성, 연간 변동성 등)를 시각화

▶ 주요 코드

[STEP 4] 모델 성능 평가

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

# 실제 값과 예측 값 추출
y_true = data['y'] # 학습 데이터의 실제 값
y_pred = forecast['yhat'].iloc[:-365] # 예측값에서 마지막 365일 제외한 부분

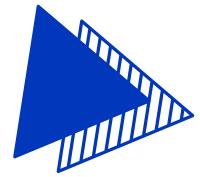
# MAE 계산
mae = mean_absolute_error(y_true, y_pred)

# RMSE 계산
rmse = mean_squared_error(y_true, y_pred)** 0.5 # RMSE는 MSE의 제곱근

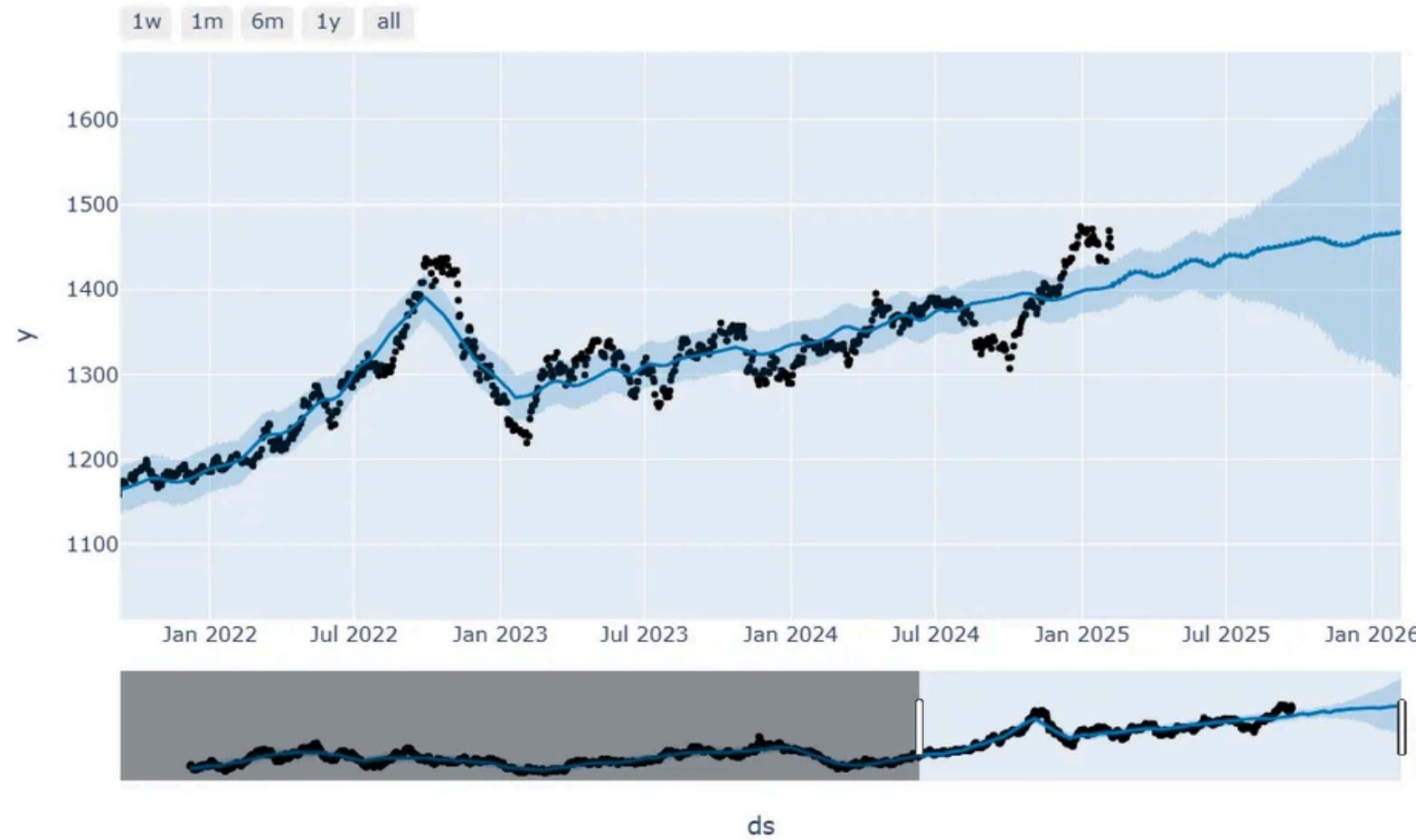
print("Mean Absolute Error (MAE):", mae)
print("Root Mean Squared Error (RMSE):", rmse)
```

- MEAN ABSOLUTE ERROR (MAE): 16.4096
- ROOT MEAN SQUARED ERROR (RMSE): 21.2526

5. PROPHET



결과(2025년 환율 예측)



저번주부터 실제 환율과 비교

평균 오차: 약 39.9975원

| date | actual | predict | predict_lower | predict_upper |
|---------------------|--------|-------------|---------------|---------------|
| 2025-02-07 00:00:00 | 1447.1 | 1404.105453 | 1377.059699 | 1430.77618 |
| 2025-02-08 00:00:00 | - | 1407.06105 | 1380.377972 | 1434.632522 |
| 2025-02-09 00:00:00 | - | 1407.42647 | 1379.440671 | 1433.160057 |
| 2025-02-10 00:00:00 | 1447.4 | 1404.939666 | 1377.835199 | 1430.156647 |
| 2025-02-11 00:00:00 | 1453.4 | 1406.005874 | 1378.354455 | 1431.915818 |
| 2025-02-12 00:00:00 | 1453.5 | 1406.096101 | 1376.617374 | 1433.976382 |
| 2025-02-13 00:00:00 | 1452.6 | 1406.513716 | 1376.85259 | 1431.848449 |
| 2025-02-14 00:00:00 | 1450.5 | 1407.036513 | 1380.443414 | 1431.069887 |
| 2025-02-15 00:00:00 | - | 1410.179668 | 1384.814492 | 1438.862011 |
| 2025-02-16 00:00:00 | - | 1410.737499 | 1384.759795 | 1438.063608 |
| 2025-02-17 00:00:00 | 1443.3 | 1408.445559 | 1380.849578 | 1436.860259 |
| 2025-02-18 00:00:00 | 1441.1 | 1409.706238 | 1383.179902 | 1437.827046 |
| 2025-02-19 00:00:00 | 1443.9 | 1409.987292 | 1382.865333 | 1434.632342 |
| 2025-02-20 00:00:00 | 1440.6 | 1410.588492 | 1383.268191 | 1437.347049 |

TEMPO와 비교했을 때 2025년 환율이 더 변동성있게 잘 예측

→ 최근 발생한 비상계엄 선포와 대통령 탄핵과 같이 정치적으로 불안정한 상황으로 인한 환율 변동 예측에는 어려움

SECTION 06

환차익 전략 수립

▶ 외환(Forex, FX) 마진 거래

외환(Forex, FX) 마진 거래는 전 세계 통화(외환)를 사고파는 투자 방식으로, 레버리지를 활용하여 적은 자본으로도 큰 규모의 거래가 가능한 특징

- ◆ 통화쌍(Currency Pair) 거래
 - 모든 거래는 두 개의 통화로 이루어지며, 예를 들어 $USD/KRW = 1,300$ 이면 1달러당 1,300원의 가치가 있다는 의미로 해석됨.
- ◆ 롱(Long) & 숏(Short) 투자 가능
 - 롱(Long, 매수): 통화 가치가 상승할 것으로 예상될 때 매수
 - 숏(Short, 매도): 통화 가치가 하락할 것으로 예상될 때 매도



▶ TEMPO vs PROPHET

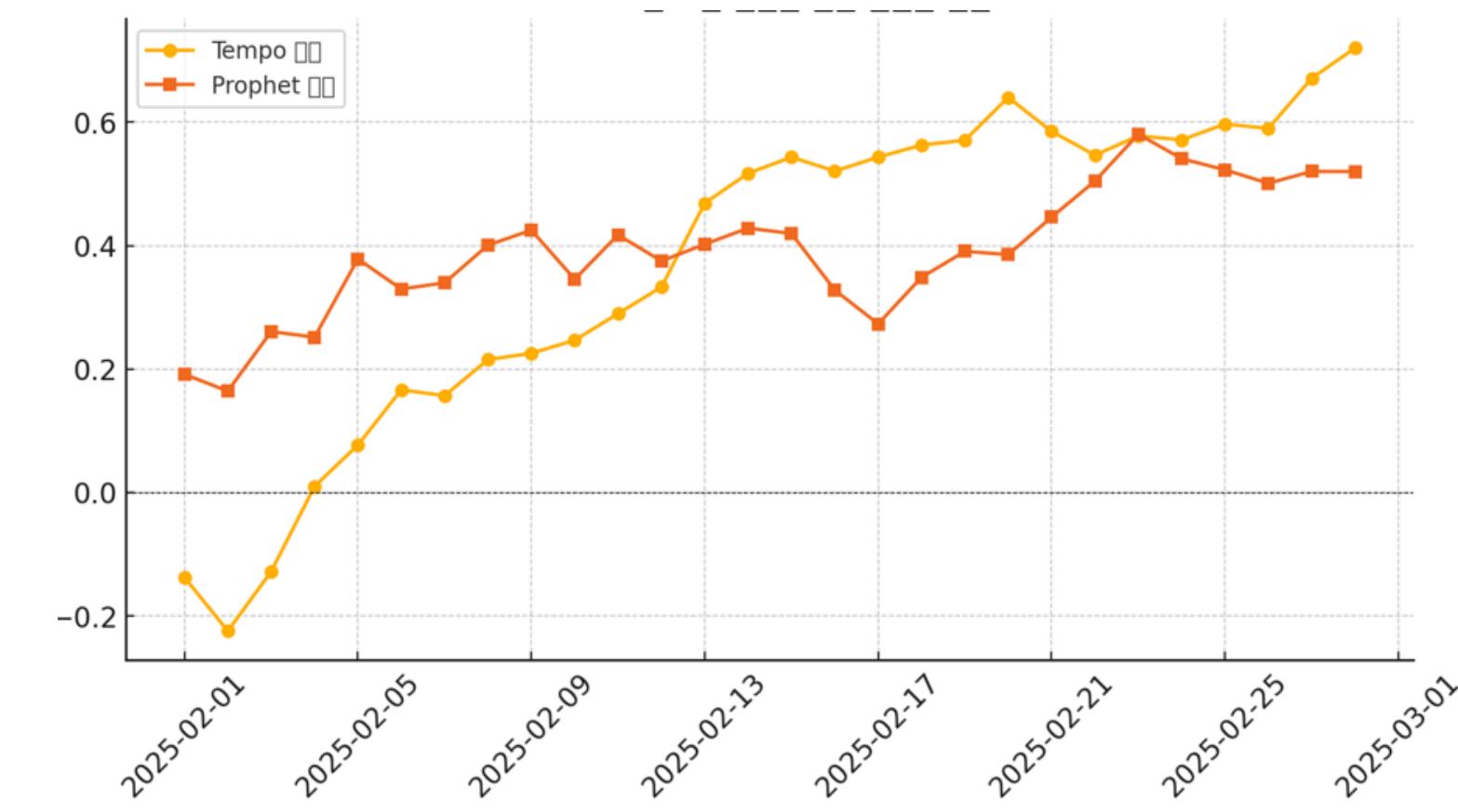
두 모델에 대해서 나온 예측값을 바탕으로 외환(Forex, FX) 마진 거래 진행

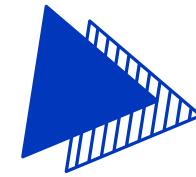
TEMPO 수익률

0.72%

PROPHET 수익률

0.52%



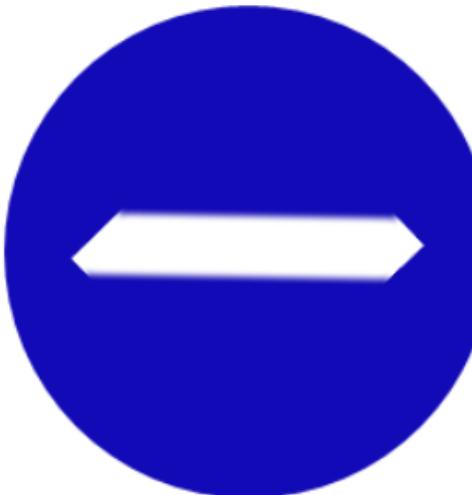


Tempo 모델 기반 예측 트레이딩 (신뢰도 필터 적용)



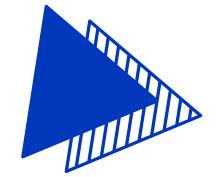
long / short

신뢰도 > 70%



stay

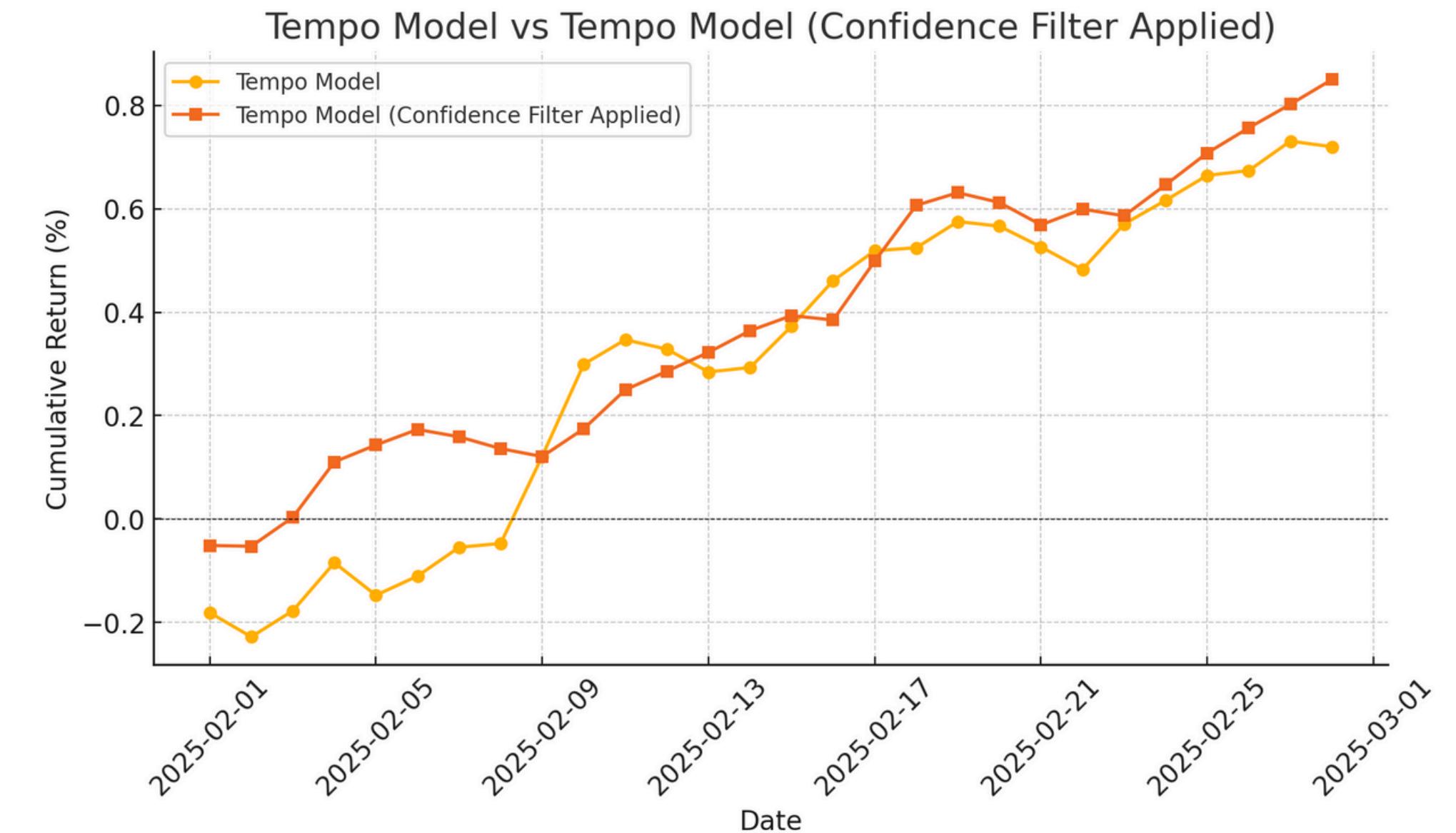
신뢰도 < 70%



Tempo 모델 기반 예측 트레이딩 (신뢰도 필터 적용)

TEMPO
0.72%

TEMPO (신뢰도 필터 적용)
0.85%





BITAMIN 2025-겨울 프로젝트

감사합니다.

Presented by:

시계열 1조: 문예림 박보희 양태원 이수민 이수아 이채연