TimeXL: Explainable Multi-modal Time Series Prediction with LLM-in-the-Loop

Qingyang Cao

8 LP

Seminar: Cross-Temporal NLP
WiSe 25/26
December 16, 2025

Overview

Overview of This Section

Related Work in Time Series Research

Related work around multi-modality, explanation, and the use of LLMs:

- Multi-modal Time Series Analysis: the interactions across different modalities
- Time Series Explanation: saliency-based, surrogate-based, ...
- LLMs for Time Series Analysis[1]: textual data encoded as prefix embeddings, zero-shot or few-shot capabilities of LLMs[2], knowledge inference modules, ...

[1] framing tasks as natural-language problems

[2] directly prompting pre-trained language models with text-converted time series or context-laden prompts representing domain knowledge

External Contextual Signals in Multi-modal Time Series

- The benefit of external contextual signals
  Worthwhile: to include external textual narratives alongside temporal dependencies (deep learning models, multi-modal approaches) for a more accurate forecasting and explainability
- The lack of explicit mechanisms
  Inefficient: systematically reason and explain about why or how contextual signals affect outcomes

LLMs in Agentic Designs

- The ability to process and reason over textual data across domains
- The facilitating structured reasoning and iterative decision-making for context-rich scenarios
- The encoded domain knowledge(with external textual context) makes them natural candidates for supporting real-world multi-modal time series analyses

Remained Problem: The Mechanism Conflict

- Multi-modal time series models[3]
  - Better accuracy
  - The lack of faithful, human-interpretable reasoning
- LLM-based approaches
  - Good reason well over text
  - Weakly grounded in temporal structure

[3] Combining numeric signals with text

The Ambition of TimeXL

An **accurate** and **interpretable** multi-modal forecasting:

- accurate prediction
- case-based, multi-modal explanations
- with an iterative improvement loop

Overview of This Section

Problem Statement

The multi-modal time series prediction problem:

- Each instance: the multi-modal input (x, s)
  - $x = (x_1, x_2, \cdots, x_T) \in \mathbb{R}^{N \times T}$: time series data
  - N: # variables
  - T: # historical time steps
  - s: text data[4]
- The objective: to predict the future outcome y [5]
  In discrete decision-making scenarios: interpretability is often essential for reliable decision support.

[4] real-world context, divided into L meaningful segments
[5] discrete in classification tasks, or continuous in regression tasks

Four Major Component of the TimeXL Framework

1. The multi-modal prototype encoder: $\mathcal{M}_{\text{enc}}$
   provides an initial prediction and case-based explanations
2. A prediction LLM: $\mathcal{M}_{\text{pred}}$
   provides a prediction based on contextual understanding with explanations
3. A reflection LLM: $\mathcal{M}_{\text{refl}}$
   generates feedback
4. A refinement LLM: $\mathcal{M}_{\text{enc}}$
   refines the textual context based on the feedback

TimeXL Workflow

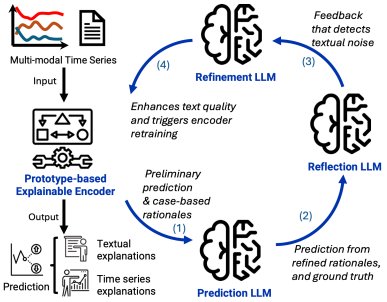


Figure: An overview of TimeXL

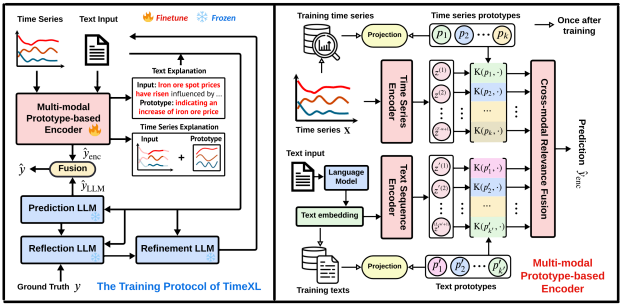
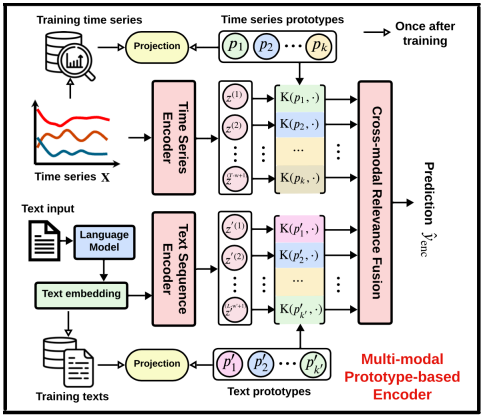


Figure: The training protocol and the encoder

Multi-modal Sequence Encoding for Dependencies

- Candidates for prototype learning: one or multiple representations the time series mapped into
- Embeddings $e_s \in \mathbb{R}^{d_s \times L}$: produced by transform text input s by a *frozen* PLM (e.g., BERT or Sentence-BERT)
- Time series encoder: $\mathcal{E}_\theta$
- Text encoder $\mathcal{E}_\phi$: extract text features from embeddings



Choice of Encoders

- The choice of $\mathcal{E}_\theta$ and $\mathcal{E}_\phi$ also affects the granularity of explanations
- Choice here: convolution-based encoders
  for both modalities to capture the fine-grained sub-sequence (i.e., segment) patterns

$$\mathbf{Z}_{\text{time}} = (z_1, \ldots, z_{T-w+1}) = \mathcal{E}_\theta(x), \qquad \mathbf{Z}_{\text{text}} = (z'_1, \ldots, z'_{L-w'+1}) = \mathcal{E}_\phi(e_s). \quad (1)$$

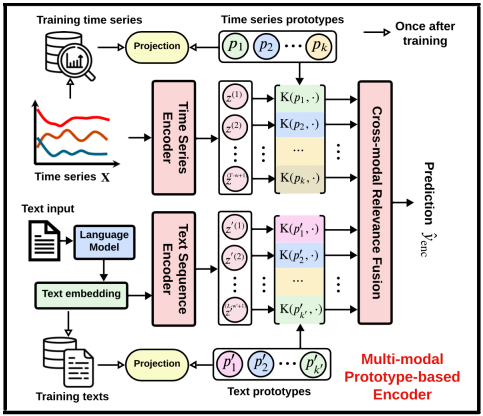
$z_i \in \mathbb{R}^h$ and $z'_j \in \mathbb{R}^{h'}$: segment-level representations learned via convolutional kernels of sizes w and w'

Prototype Allocation to Establish the Interpretability (I)

1. **Learn a set of prototypes** for each class $c \in \{1, \ldots, C\}$:
   - time series prototypes: $\mathbf{P}_{\text{time}}^{(c)} \in \mathbb{R}^{k \times h}$
   - text prototypes: $\mathbf{P}_{\text{text}}^{(c)} \in \mathbb{R}^{k' \times h'}$
2. **Measure the similarity** between:
   - each prototype
   - the most relevant segment

$$\text{Sim}_i^{(c)} = \max\left(\text{Sim}_{i,1}^{(c)}, \ldots, \text{Sim}_{i,T-w+1}^{(c)}\right), \tag{2}$$

$$\text{Sim}_{i,j}^{(c)} = \exp\left(-\left\|p_i^{(c)} - z_j\right\|_2^2\right) \in [0, 1].$$

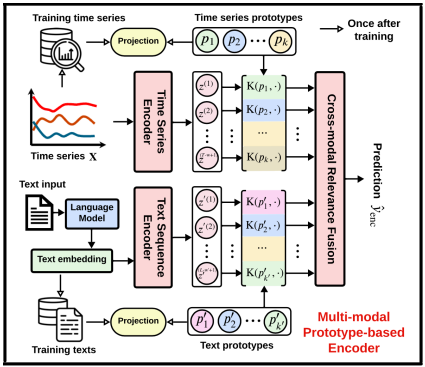


Prototype Allocation to Establish the Interpretability (II)

3. **Aggregate similarity scores**: $\mathrm{Sim}_{\mathsf{time}} \in \mathbb{R}^{kC}$ and $\mathrm{Sim}_{\mathsf{text}} \in \mathbb{R}^{k'C}$
4. **Translates into class probabilities**:

$$\hat{y}_{\mathrm{enc}} = \mathrm{Softmax}\big(\mathbf{W}\,[\mathrm{Sim}_{\mathsf{time}} \,\|\, \mathrm{Sim}_{\mathsf{text}}]\big) \in [0,1]^C. \tag{3}$$

with fusion weight matrix $\mathbf{W} \in \mathbb{R}^{C\times(k+k')}$



Learning Objectives with Regularization Terms (I)

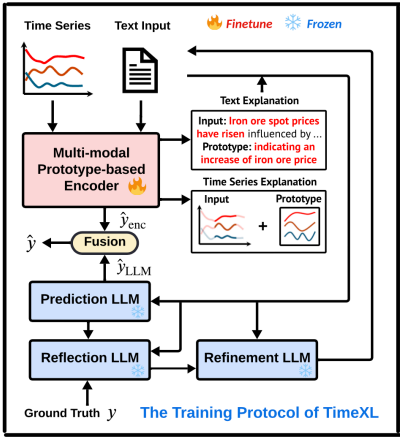
- Basic objective: the cross-entropy loss

$$\mathcal{L}_{\text{CE}} = \sum_{x,s,y} y \log(\hat{y}_{\text{enc}}) + (1-y) \log(1-\hat{y}_{\text{enc}}).$$

- Clustering $\mathcal{L}_c$ and evidencing $\mathcal{L}_e$:

$$\mathcal{L}_c = \sum_{z_j \in \mathbf{Z}_{(\cdot)}} \min_{p_i \in \mathbf{P}_{(\cdot)}} \|z_j - p_i\|_2^2, \qquad (4)$$

$$\mathcal{L}_e = \sum_{p_i \in \mathbf{P}_{(\cdot)}} \min_{z_j \in \mathbf{Z}_{(\cdot)}} \|p_i - z_j\|_2^2.$$

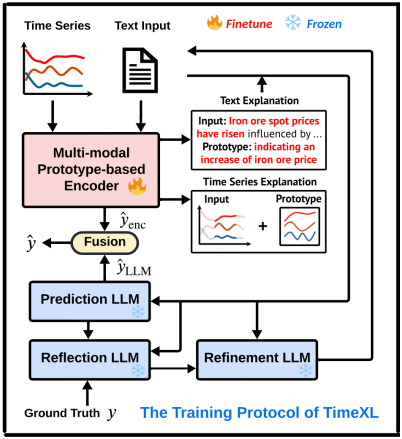


Learning Objectives with Regularization Terms (II)

- Diversion penalty of the hinge loss $\mathcal{L}_d$ with a threshold $d_{\min}$:

$$\mathcal{L}_d = \sum_{i=1} \sum_{j \neq i} \max\left(0,\ d_{\min} - \|p_i - p_j\|_2^2\right). \tag{5}$$

- The final objective:
$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_1 \mathcal{L}_c + \lambda_2 \mathcal{L}_e + \lambda_3 \mathcal{L}_d,$



Prototype Projection for Explaination

- Associating each prototype with a training segment in the representation space

$$p_i^{(c)} \leftarrow \arg \min_{z_j \in \mathbf{Z}_{(\cdot)}^{(c)}} \left\| p_i^{(c)} - z_j \right\|_2^2, \qquad \forall\, p_i^{(c)} \in \mathbf{P}_{(\cdot)}^{(c)}. \tag{6}$$

- Explanation:
  - the similarity scores
  - the contribution weights
  - the prototypes' class information

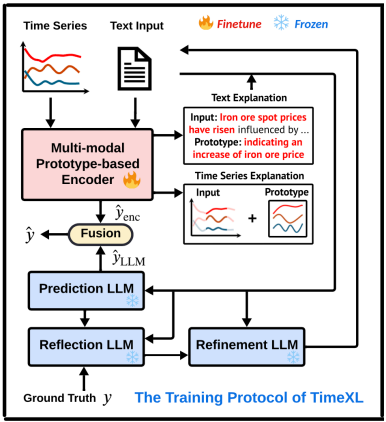
Prediction with Enriched Contexts

- $\mathcal{M}_{\text{enc}}$: supplements s with *case-based explanations* (Eq. (2)):
- $\mathcal{M}_{\text{pred}}$: generates predictions based on:
  - input text s
  - selected
- The explanation expl_s:

$$\text{expl}_s = \left\{ (p_i^{(c)}, s_j) : (i, j, c) \in \text{Top-}\omega(\text{Sim}_{\text{text}}) \right\}$$

$$\text{Top-}\omega(\text{Sim}_{\text{text}}) = \arg \text{Top-}\omega_{(i,j,c)} \left(\text{Sim}_{i,j}^{(c)} \right).$$

constructed with ω prototype – segment pairs



The Prompt of Prediction Querying

The prediction for prediction agent $\mathcal{M}_{\text{pred}}$: $\hat{y}_{\text{LLM}} = \mathcal{M}_{\text{pred}}(s, \text{expl}_s)$.

System Prompt

Your job is to act as [specific role]. You will be given a summary of [data description] and related prototypes that you can refer to. Based on this information, your task is to predict [task description].

User Prompt

Your task is to [task description]. First, review the following [number of prototypes] prototype text segments and outcomes, so that you can refer to when making predictions.

Prototype #1: [text prototype]
Corresponding Segment#1: [input text segment]
Relevance Score: [similarity score]
Outcome #1: [options]

...

Next, review the [situation] :
Summary: [text input]

Based on your understanding, predict the outcome of [situation]. Respond your prediction with [options]. Response should not include other terms.

Figure: Prompt for prediction LLM $\mathcal{M}_{\text{pred}}$

Fused Predictions

$$\hat{y} = \alpha \hat{y}_{\text{enc}} + (1 - \alpha)\hat{y}_{\text{LLM}}, \qquad \alpha \in [0, 1]$$

a

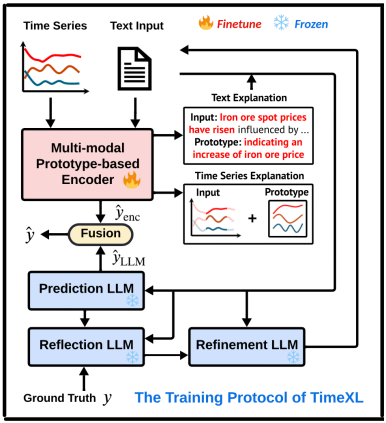
The final prediction is based on a fusion:

- The multi-modal encoder $\mathcal{M}_{\text{enc}}$
- The prediction LLM $\mathcal{M}_{\text{pred}}$

Linearly combine:

- The continuous prediction prob. $\hat{y}_{\text{enc}}$
- The discrete prediction $\hat{y}_{\text{LLM}}$

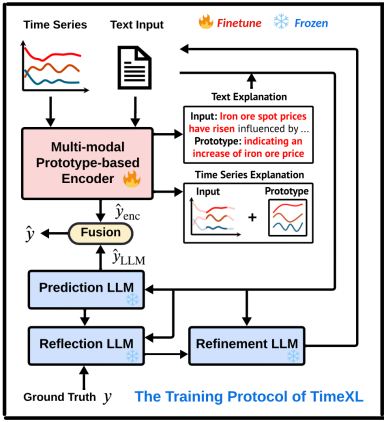
[a] α: the hyperparameter selected from validation data.



Fitting The Context of Multi-modal Time Series Data

Generate reflective feedback and refinements on the context.

- $\hat{y}_{\text{LLM}}$: the prediction generated by $\mathcal{M}_{\text{pred}}$
- $\text{Refl} = \mathcal{M}_{\text{refl}}(y, \hat{y}_{\text{LLM}}, s)$
- Refinement update: $s_{i+1} = \mathcal{M}_{\text{refine}}(\text{Refl}, s_i)$



The Prompt of Reflection & Refinement Querying (I)

System Prompt

You are an advanced reasoning agent that can improve the quality of [domain] summary based on self reflection. You will be given the summaries and [correct flag] predictions of [situation]. Your task is to learn some reflections that guides the refinement of [domain] summaries.

User Prompt

Your task is to analyze the provided [domain] summaries with [correct flag] predictions, in order to generate a reflection report improving its quality for [situation] prediction.

Review the following [number of summaries] [domain] summaries with [ground truth] actual outcomes and [prediction] predictions.

Summary #1: [text input]
Actual Outcome #1: [ground truth]
Prediction #1: [prediction]

...

Based on your analysis, write a high-quality reflection report that summarizes key phrases or sentences that led to correct predictions of [situation] / commonly misinterpreted and overlooked phrases or sentences that led to incorrect predictions of [situation].

Use precise terms to convey a clear and professional analysis, and avoid overly general statements. The report should be a comprehensive and informative paragraph, which can be generalized to refine similar [domain] summaries. Your response should not include other terms.

(a) Reflection generation

System Prompt

You are an advanced reasoning agent that can improve the quality of [domain] summary based on self reflection. You will receive a reflection report up to this point. You will also be given the summaries and [correct flag] predictions of [situation]. Your task is to learn some reflections and update the current report that guides the refinement of [domain] summaries.

User Prompt

Your task is to analyze the provided [domain] summaries with [correct flag] predictions, in order to update a reflection report improving its quality for [situation] prediction.

First, review the following reflection report up to this point: [current reflection report]

Next, review the following [number of summaries] [domain] summaries with [ground truth] actual outcomes and [prediction] predictions.

Summary #1: [text input]
Actual Outcome #1: [ground truth]
Prediction #1: [prediction]

...

Based on your analysis, write a high-quality reflection report that summarizes key phrases or sentences that led to correct predictions of [situation] / commonly misinterpreted and overlooked phrases or sentences that led to incorrect predictions of [situation].

Use precise terms to convey a clear and professional analysis, and avoid overly general statements. The report should contain incremental and context-aware updates, and can be generalized to refine similar [domain] summaries. Your response should not include other terms.

(b) Reflection update

The Prompt of Reflection & Refinement Querying (II)

System Prompt

You are an advanced summarization agent that can generate high-quality summarization. You will be given previously generated reflections for text refinement, from the correct and incorrect predictions of [domain] texts. Your current task is to summarize these long reflections to better guide financial text refinement.

User Prompt

Your task is to summarize the long reflections derived from previous predictions of [domain] contents. The goal is to generate a high-quality report aimed at improving the [domain] text quality for better predictive accuracy.

First, review the reflections from all combinations of possible predictions and actual outcomes: [reflection reports]

Based on your analysis, summarize the reflections of different scenarios and write a comprehensive report that provides guidelines to select the most important content in new [domain] texts where the actual outcome is unknown. Your response should keep the enough details, yet effective, to improve the text quality for downstream prediction. Your response should not include other terms.

(c) Reflection summarization

System Prompt

You are an advanced refinement agent designed to enhance the quality of [domain] summary. You will be provided with reflective thoughts analyzed from other summaries, and a summary that requires refinement. Your task is to generate a refined [domain] summary, by examining how reflective thoughts applied to the current summary.

User Prompt

Your task is to generate a refined weather summary from the current summary to improve its predictions of [situation]. First, review the following reflections that provide guidelines for refinement:

[final reflection report]

Next, review the current [domain] summary that describes [situation]:

Summary #1: [text input]

Based on your understanding, generate a new weather summary by selecting relevant content in the current summary, which provides insights crucial for understanding [situation]. Response should not include other terms.

(d) Refinement

The Overall Algorithm

Algorithm 1 Iterative Optimization Loop of TimeXL

Inputs: Multi-modal time series $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{y})$ with $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, $\mathcal{D}_{\text{test}}$, prototype-based encoder $\mathcal{M}_{\text{enc}}$, prediction agent $\mathcal{M}_{\text{pred}}$, reflection agent $\mathcal{M}_{\text{refl}}$, refinement agent $\mathcal{M}_{\text{refine}}$, fusion parameter α, max iteration τ, improvement check $\text{Eval}(\cdot)$

Training & Validation:

Initialize $\boldsymbol{s}_0 = \boldsymbol{s}$, $i = 0$, $\mathcal{M}^*_{\text{enc}} \leftarrow \varnothing$, $\text{Refl}^* \leftarrow \varnothing$
while $i < \tau$ **do**
  Train $\mathcal{M}_{\text{enc}}$ using $\mathcal{D}_{\text{train,i}} = \{(\boldsymbol{x}, \boldsymbol{s}_i, \boldsymbol{y}), \cdots\}$
  $\hat{\boldsymbol{y}}_{\text{enc}}, \mathbf{expl}_{\boldsymbol{s}_i} = \mathcal{M}_{\text{enc}}(\boldsymbol{x}, \boldsymbol{s}_i)$
  $\hat{\boldsymbol{y}}_{\text{LLM}} = \mathcal{M}_{\text{pred}}(\boldsymbol{s}_i, \mathbf{expl}_{\boldsymbol{s}_i})$
  $\hat{\boldsymbol{y}} = \alpha \hat{\boldsymbol{y}}_{\text{enc}} + (1 - \alpha)\hat{\boldsymbol{y}}_{\text{LLM}}$
  $\text{Refl} = \mathcal{M}_{\text{refl}}(\boldsymbol{y}, \hat{\boldsymbol{y}}_{\text{LLM}}, \boldsymbol{s}_i)$
  $\boldsymbol{s}_{i+1} = \mathcal{M}_{\text{refine}}(\text{Refl}, \boldsymbol{s}_i)$
  if $\text{Eval}(\text{relf}, \mathcal{D}_{\text{val}})$ *pass* **then**
    $\mathcal{M}^*_{\text{enc}} \leftarrow \mathcal{M}_{\text{enc}}$, $\text{Refl}^* \leftarrow \text{Refl}$
  increment i
return $\mathcal{M}^*_{\text{enc}}$, Refl^*

Testing:

$\boldsymbol{s}' = \mathcal{M}_{\text{refine}}(\text{Refl}^*, \boldsymbol{s})$ for $\mathcal{D}_{\text{test}}$
$\hat{\boldsymbol{y}}_{\text{enc}}, \mathbf{expl}_{\boldsymbol{s}'} = \mathcal{M}^*_{\text{enc}}(\boldsymbol{x}, \boldsymbol{s}')$
$\hat{\boldsymbol{y}}_{\text{LLM}} = \mathcal{M}_{\text{pred}}(\boldsymbol{s}', \mathbf{expl}_{\boldsymbol{s}'})$
$\hat{\boldsymbol{y}} = \alpha \hat{\boldsymbol{y}}_{\text{enc}} + (1 - \alpha)\hat{\boldsymbol{y}}_{\text{LLM}}$

The Datasets

- **Four real-world multi-modal datasets**
  - Weather forecasting
  - Financial risk prediction
  - Healthcare diagnosis (2 datasets)
- Each instance contains multivariate time series and aligned textual context

Domain	Dataset	Resolution	# Channels	# Timesteps	Duration	Ground Truth Distribution
Weather	New York	Hourly	5	45,216	2012.10 - 2017.11	Rain (24.26%) / Not rain (75.74%)
Finance	Raw Material	Daily	15	1,876	2012.09 - 2022.02	Inc. (36.7%) / Dec. (34.1%) / Neutral (29.2%)
Healthcare	Test-Positive	Weekly	6	447	2015.10 - 2024.04	Not exceed (65.77%) / Exceed (34.23%)
Healthcare	Mortality	Weekly	4	395	2016.07 - 2024.06	Not exceed (69.33%) / Exceed (30.67%)

Figure: Summary of dataset statistics

Baselines, Evaluation Metrics, and Setup

- Baselines
  - **Time-series only models**: CNN / RNN-based predictors
  - **Multi-modal deep models**: joint temporal-text encoders
  - **LLM-based approaches**: prompt-based forecasting
- Metrics
  - AUROC (primary)
  - Accuracy / F1-score
- Identical forecasting horizons
- Hyperparameters tuned on validation set

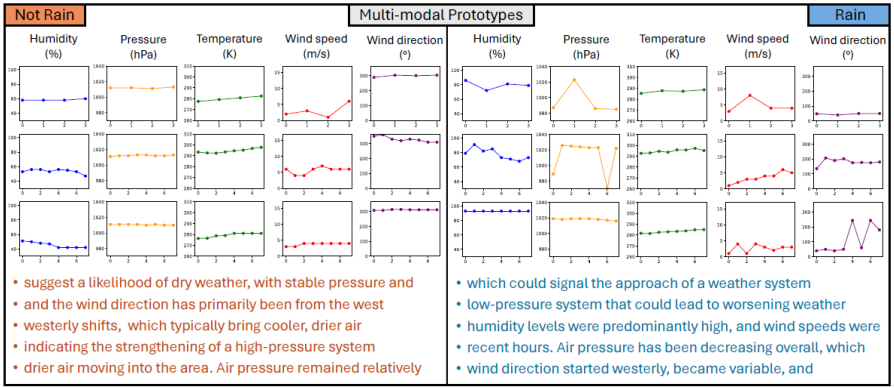
Overall Performance

TimeXL outperforms all baselines acrossing all datasets, up to **+8.9% AUROC**:

Datasets → / Methods ↓	Weather		Finance		Healthcare (TP)		Healthcare (MT)	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
DLinear [75]	0.540	0.660	0.255	0.485	0.393	0.500	0.419	0.388
Autoformer [76]	0.546	0.590	0.565	0.747	0.774	0.918	0.683	0.825
Crossformer [77]	0.500	0.594	0.571	0.775	0.924	0.984	0.737	0.913
TimesNet [78]	0.494	0.594	0.538	0.756	0.794	0.867	0.765	0.944
iTransformer [79]	0.541	0.650	0.600	0.783	0.861	0.931	0.791	0.963
TSMixer [80]	0.488	0.534	0.465	0.689	0.770	0.797	0.808	0.931
TimeMixer [81]	0.577	0.658	0.571	0.776	0.822	0.887	0.824	0.935
FreTS [82]	0.623	0.688	0.546	0.737	0.887	0.950	0.751	0.762
PatchTST [5]	0.592	0.675	0.604	0.795	0.841	0.934	0.695	0.928
LLMTime [83]	0.587	0.657	0.519	0.643	0.802	0.817	0.769	0.803
PromptCast [63]	0.499	0.365	0.418	0.607	0.727	0.768	0.696	0.871
OFA [53]	0.501	0.606	0.512	0.745	0.774	0.879	0.851	0.977
FSCA [58]	0.563	0.647	0.592	0.790	0.820	0.891	0.872	0.977
Time-LLM [56]	0.613	0.699	0.589	0.792	0.671	0.864	0.733	0.912
TimeCMA [57]	0.636	0.731	0.559	0.727	0.729	0.828	0.693	0.843
MM-iTransformer [35]	0.608	0.689	0.605	0.793	0.926	0.986	0.901	0.990
MM-PatchTST [35]	0.621	0.718	0.619	**0.812**	0.863	0.968	0.780	0.929
TimeCAP [65]	0.668	0.742	0.611	0.801	0.954	0.983	0.942	0.988
TimeXL	**0.696**	**0.808**	**0.631**	0.797	**0.987**	**0.996**	**0.956**	**0.997**

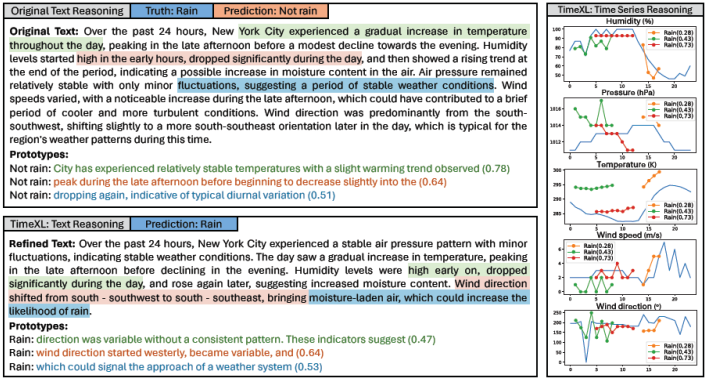
Explainable Multi-modal Prototypes

Prototypes correspond to concrete segments, learned separately for each class, and jointly explain time and text



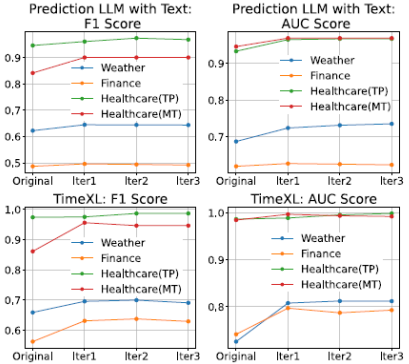
Multi-modal Case-based Reasoning

Predictions are justified by similar past cases, which is similarity learned in representation space with modalities contribution.



Effect of Iterative Refinement

- Performance & explanation improves across iterations
- Early stops after 1–2 iterations



Ablation: Model Components

- Remove text modality → large drop
- Remove prototypes → explanations collapse
- Remove LLM loop → weaker performance

Ablation ↓	Variants	Weather		Finance		Healthcare (Test-Positive)		Healthcare (Mortality)	
		F1	AUC	F1	AUC	F1	AUC	F1	AUC
Encoder	Time Series	0.602	0.691	0.585	0.751	0.889	0.957	0.861	0.965
	Text	0.567	0.658	0.472	0.636	0.871	0.941	0.840	0.887
	Multi-modal	0.674	0.767	0.619	0.791	0.934	0.974	0.937	0.988
LLM	Text	0.645	0.724	0.496	0.627	0.974	0.967	0.901	0.969
	Text + Prototype	0.667	0.739	0.544	0.662	0.987	0.983	0.952	0.976
Fusion	Select-Best	0.674	0.767	0.619	0.791	0.987	0.983	0.952	0.988
	TimeXL	**0.696**	**0.808**	**0.631**	**0.797**	**0.987**	**0.996**	**0.956**	**0.997**

Figure: Component ablation results

The Mechanism of TimeXL

- TimeXL is an **explainable multi-modal time series framework** combining:
  - prototype-based temporal modeling
  - LLM-based semantic reasoning

  where explanations are **intrinsic**, not post-hoc.
- Two Key components:
  1. Multi-modal prototype encoder:
     - learns case-based representations
     - grounds predictions in historical evidence
  2. LLM-in-the-loop:
     - reasons over retrieved cases
     - refines contextual text iteratively

Strengths

- Faithful, human-readable explanations
- Clear separation of roles:
  - encoder = temporal structure
  - LLM = semantic reasoning
- Strong empirical performance across domains

Limitations

- Additional computational cost from LLM calls
- Dependence on LLM quality and calibration
- Iterative refinement may introduce bias

What is The Next Step?

- The choice of encoders (reproducibility)?
- The similarity measurement (possible improvement)?
- The efficiency of the iteration (possible oscillation)?
- The continuous tasks (generalization)?
- Is the linear combination valid (continuous vs. discrete)?
- Optional: neural ODEs, operator learning, ...

Thanks!