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

Anonymous Authors[1]

Abstract

Time series analysis provides essential insights for real-world system dynamics and informs downstream decision-making, yet most existing methods often overlook the rich contextual signals present in auxiliary modalities (e.g., financial news or domain-specific documents). To bridge this gap, we introduce TimeXL, a multi-modal prediction framework that integrates a prototype-based time series encoder with three collaborating Large Language Models (LLMs) to deliver more accurate predictions and interpretable explanations. First, a multi-modal prototype-based encoder processes both time series and textual inputs to generate preliminary forecasts alongside case-based rationales. These outputs then feed into a prediction LLM, which refines the forecasts by reasoning over the encoder's predictions and explanations. Next, a reflection LLM compares the predicted values against the ground truth, identifying textual inconsistencies or noise. Guided by this feedback, a refinement LLM iteratively enhances text quality and triggers encoder retraining. This closed-loop workflow—prediction, critique (reflect), and refinement—continuously boosts the framework's performance and interpretability. Empirical evaluations on four real-world datasets demonstrate that TimeXL achieves up to 8.9 % improvement in AUC and produces human-centric, multi-modal explanations, highlighting the power of LLM-driven reasoning for time series prediction.

1. Introduction

In the modern big-data era, time series analysis has become indispensable for understanding real-world system behaviors and guiding downstream decision-making tasks across numerous domains, including healthcare, traffic, finance, and weather (Jin et al., 2018; Guo et al., 2019; Zhang et al., 2017; Qin et al., 2017). Although deep learning models have demonstrated success in capturing complex temporal dependencies (Nie et al., 2023; Deng & Hooi, 2021; Zhang et al., 2022; Liu et al., 2024d), real-world time series are frequently influenced by external information beyond purely temporal factors. Such additional context, which may come from textual narratives (e.g., finance news (Dong et al., 2024) or medical reports (King et al., 2023)), can offer critical insights for more accurate forecasting and explainability.

Recent multi-modal approaches for time series have shown promise by integrating rich contextual signals from disparate data sources—such as textual descriptions—to improve performance on tasks ranging from forecasting and classification to imputation and retrieval (Ekambaram et al., 2020; Niu et al., 2023; Lee et al., 2024; Xing & He, 2023; Moroto et al., 2024; Zhao et al., 2022; Bamford et al., 2023). While these approaches utilize supplementary data to enhance predictive accuracy, they often lack explicit mechanisms to systematically reason and explain about *why* or *how* contextual signals affect outcomes. This gap in interpretability poses significant barriers for high-stakes applications such as finance or healthcare, where trust and transparency are paramount.

Meanwhile, Large Language Models (LLMs) (Achiam et al., 2023; Team et al., 2023; Touvron et al.) have risen to prominence for their remarkable ability to process and reason over textual data across domains, enabling tasks like sentiment analysis, question answering, and content generation in zero- and few-shot settings (Zhang et al., 2024; Kamalloo et al., 2023; Wang et al., 2024c). Their encoded domain knowledge makes them natural candidates for supporting multi-modal time series analyses, where textual context (e.g., news or expert notes) plays a vital role (Liu et al.; Nie et al., 2024; Koa et al., 2024; Wang et al., 2023; Shi et al., 2024; Yu et al., 2023; Singhal et al., 2023).

Motivated by these observations, we introduce TimeXL, a novel framework that adopts a closed-loop workflow of prediction, critique (reflect), and refinement, and unifies a prototype-driven time series encoder with LLM-based reasoning to deliver both accurate and interpretable multi-modal forecasting (Figure 1). Our approach first employs

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

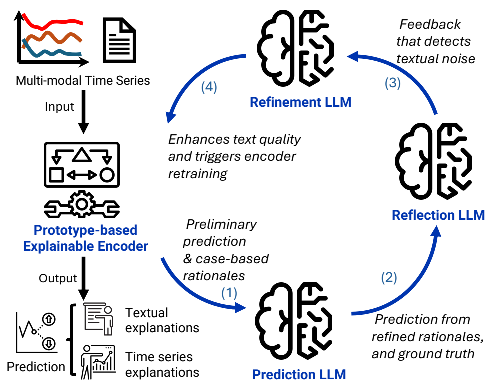

Figure 1. An overview of the TimeXL workflow. A prototype-based explainable encoder first produces predictions and case-based rationales for both time series and text. The prediction LLM refines forecasts based on these rationales (Step 1). A reflection LLM then critiques the output against ground truth (Step 2), providing feedback to detect textual noise (Step 3). Finally, a refinement LLM updates the text accordingly, triggering encoder retraining for improved accuracy and explanations (Step 4).

a *multi-modal prototype-based encoder* to generate preliminary time series predictions alongside human-readable explanations, leveraging case-based reasoning (Kolodner, 1992; Ming et al., 2019; Ni et al., 2021; Jiang et al., 2023) from both the temporal and textual modalities. These explanations not only justify the encoder's predictions but also serve as auxiliary signals to guide an LLM-powered component that further refines the forecasts and contextual rationales.

Unlike conventional methods that merely fuse multi-modal inputs for better accuracy, TimeXL iterates between predictive and refinement phases to mitigate textual noise, fill knowledge gaps, and produce more faithful explanations. Specifically, a *reflection LLM* diagnoses potential weaknesses by comparing predictions with ground-truth signals, while a *refinement LLM* incorporates these insights to update textual inputs and prototypes iteratively. This feedback loop progressively improves both the predictive and explanatory capabilities of the entire system. Our contributions are summarized as follows:

- We present a prototype-based encoder that combines time series data with textual context, producing transparent, case-based rationales.
- We exploit the interpretative prowess of LLMs to reason over the encoder's outputs and iteratively refine both predictions and text, leading to improved prediction accuracy and explanations.
- Experiments on four real-world benchmarks show that TimeXL consistently outperforms baselines, achieving up to a 8.9% improvement in AUC while providing faithful, human-centric multi-modal explanations.

Overall, TimeXL opens new avenues for explainable multi-modal time series analysis by coupling prototype-based inference with LLM-driven reasoning.

2. Related Work

2.1. Multi-modal Time Series Analysis

In recent years, multi-modal time series analysis has gained significant traction in diverse domains such as finance, healthcare, environmental sciences, and industry (Ekambaram et al., 2020; Skenderi et al., 2024; Niu et al., 2023; Yang & Wu, 2021; Zhao et al., 2022; Xing & He, 2023). Multiple approaches have been proposed to model interactions across different modalities for various tasks. For instance, (Lee et al., 2024) introduces a multi-modal augmentation framework for few-shot time series forecasting, which fuses time series and textual representations both at the sample and feature levels using attention. Furthermore, (Bamford et al., 2023) aligns multi-modal time series within a shared latent space of deep encoders and retrieves specific sequences based on textual queries. In addition, (Zheng et al., 2024) performs causal structure learning to uncover root causes in multi-modal time series by separating modality-invariant and modality-specific components via contrastive learning. Most recently, (Liu et al., 2024a) established a multi-modal forecasting benchmark with baselines, and demonstrating performance improvements through the incorporation of a new modality. Although these techniques have advanced predictive performance by leveraging cross-modality interactions, they tend to focus primarily on improving numerical accuracy. The deeper reasoning behind *how* or *why* the textual or other contextual signals influence time series outcomes remains underexplored.

2.2. Time Series Explanation

Recent studies have explored diverse paradigms for time series interpretability. Gradient-based and perturbation-based "saliency" methods, for example, highlight important features at different time steps (Ismail et al., 2020; Tonekaboni et al., 2020), while other works explicitly incorporate temporal structures into models and objectives (Leung et al.; Crabbé & Van Der Schaar, 2021). Surrogate approaches also offer global or local explanations, such as applying Shapley values to time series (Bento et al., 2021), enforcing model consistency via self-supervised objectives (Queen et al., 2024), or using information-theoretic strategies for coherent explanations (Liu et al., 2024f). In contrast to saliency or surrogate-based explanations, we adopt a *case-based reasoning* paradigm (Kolodner, 1992; Ming et al., 2019; Ni et al., 2021; Jiang et al., 2023), which end-to-end generates predictions and built-in explanations from learned prototypes. Our work extends this approach to multi-modal time series by producing human-readable reasoning artifacts for both the temporal and contextual modalities.

2.3. LLMs for Time Series Analysis

The rapid development of Large Language Models (LLMs) (Achiam et al., 2023; Team et al., 2023; Touvron et al.) has begun to inspire new directions in time series research (Jiang et al., 2024; Liang et al., 2024). Many existing techniques fine-tune pre-trained LLMs on time series tasks, achieving state-of-the-art results in forecasting, classification, and beyond (Zhou et al., 2023; Bian et al.; Ansari et al., 2024). Often, textual data—such as domain instructions, metadata, or dataset summaries—are encoded as prefix embeddings to enrich time series representations (Jin et al., 2024; Liu et al., 2024b; Jia et al., 2024; Liu et al., 2025). These techniques also contribute to the emergence of time series foundation models (Ansari et al., 2024; Das et al., 2023; Woo et al., 2024; Liu et al., 2024e; Wang et al., 2024a). An alternative line of research leverages the zero-shot or few-shot reasoning capabilities of LLMs. These methods directly prompt pre-trained language models with text-converted time series (Xue & Salim, 2023) or context-laden prompts representing domain knowledge (Wang et al., 2023; Yu et al., 2023; Singhal et al., 2023), often yielding surprisingly strong performance in real-world scenarios. Furthermore, LLMs can act as knowledge inference modules, synthesizing high-level patterns or explanations that augment standard time series pipelines (Chen et al., 2023b; Shi et al., 2024; Lee et al., 2025; Wang et al., 2024b).

3. Methodology

In this section, we present the framework for explainable multi-modal time series prediction with LLMs. We first introduce the problem statement. Next, we present the design of a time series encoder that provides prediction and multi-modal explanations as the basis. Finally, we introduce three language agents interacting with the encoder towards better prediction and reasoning results.

3.1. Problem Statement

In this paper, we consider a multi-modal time series prediction problem. Each instance is represented by the multi-modal input $(\boldsymbol{x}, \boldsymbol{s})$, where $\boldsymbol{x} = (x_1, x_2, \cdots, x_T) \in \mathbb{R}^{N \times T}$ denotes time series data with N variables and T historical time steps, and $\boldsymbol{s}$ denotes the corresponding text data describing the real-world context. The text data $\boldsymbol{s}$ can be further divided into L meaningful segments. Based on the historical time series and textual context, our objective is to predict the future outcome $\boldsymbol{y}$, either as a discrete value for classification tasks, or as a continuous value for regression tasks. In this paper, we mainly consider a classification task while we provide a demonstration of the regression task in Appendix F. There are three major components in the proposed TimeXL framework, a multi-modal prototype encoder $\mathcal{M}_{\text{enc}}$ that provides initial prediction and case-based explanation, a prediction LLM $\mathcal{M}_{\text{pred}}$ that provides prediction based on the understanding of context with explanation, a reflection LLM $\mathcal{M}_{\text{refl}}$ that generates feedback, and a refinement LLM $\mathcal{M}_{\text{refine}}$ that refines the textual context based on the feedback. Below, we introduce each component and how they synergize toward better prediction and explanation.

3.2. Multi-modal Prototype-based Encoder

We design a multi-modal prototype-based encoder that can generate predictions and explanations across different modalities in an end-to-end manner, as shown in Figure 2. We introduce the model architecture, the learning objectives that yield good explanation properties of prototypes, and the pipeline of case-based explanations using prototypes.

3.2.1. Multi-modal Sequence Modeling with Prototypes

Sequence Encoder. To capture both temporal and semantic dependencies, we adopt separate encoders for time series ($\mathcal{E}_\theta$) and text ($\mathcal{E}_\phi$). For $\boldsymbol{x} \in \mathbb{R}^{N \times T}$, the time series encoder $\mathcal{E}_\theta$ maps the entire sequence into one or multiple representations, which serve as candidates for prototype learning. Simultaneously, the text input $\boldsymbol{s}$ is first transformed by a *frozen* pre-trained language model, PLM (e.g., BERT(Kenton & Toutanova, 2019) or Sentence-BERT(Reimers & Gurevych, 2019)), to produce embeddings $\boldsymbol{e}_s \in \mathbb{R}^{d_s \times L}$. These embeddings are then processed by a separate encoder $\mathcal{E}_\phi$ to extract meaningful text features. It is worth noting that the choice of $\mathcal{E}_\theta$ and $\mathcal{E}_\phi$ also affects the granularity of explanations. As we will introduce shortly, the prototypes are learned based on sequence representations and are associated with the counterparts in the input space, where the correspondences are determined by the encoders. In this paper, we choose convolution-based encoders for both modalities to capture the fine-grained sub-sequence (*i.e.,* segment) patterns:

$$\boldsymbol{Z}_{\text{time}} = \left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{T-w+1}\right) \; = \; \mathcal{E}_\theta(\boldsymbol{x}), \tag{1}$$

$$\boldsymbol{Z}_{\text{text}} = \left(\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_{L-w'+1}\right) \; = \; \mathcal{E}_\phi(\boldsymbol{e}_s), \tag{2}$$

where $\boldsymbol{z}_i \in \mathbb{R}^h$ and $\boldsymbol{z}'_j \in \mathbb{R}^{h'}$ denote segment-level representations learned via convolutional kernels of sizes w and w', respectively.

Prototype Allocation. To establish interpretability, we learn a set of *time series prototypes* and *text prototypes* for each class $c \in \{1, \ldots, C\}$. Specifically, we introduce:

$$\boldsymbol{P}^{(c)}_{\text{time}} \in \mathbb{R}^{k \times h}, \quad \boldsymbol{P}^{(c)}_{\text{text}} \in \mathbb{R}^{k' \times h'},$$

so that each prototype $\boldsymbol{p}^{(c)}_i \in \mathbb{R}^h$ (time series) or $\boldsymbol{p}'^{(c)}_i \in \mathbb{R}^{h'}$ (text) resides in the same feature space as the relevant encoder outputs. For an input sequence, we measure the

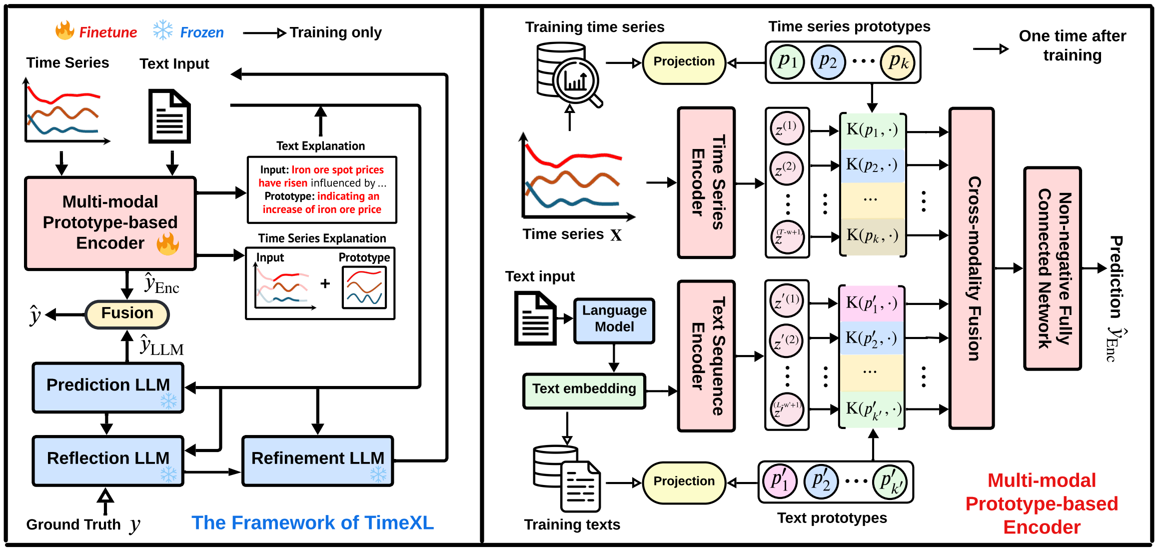

Figure 2. The overview of TimeXL (left), and multi-modal prototype-based encoder (right).

similarity between each prototype and the most relevant segment in the corresponding modality:

$$\begin{aligned} &\mathrm{Sim}_i^{(c)} = \max(\mathrm{Sim}_{i,1}^{(c)}, \cdots, \mathrm{Sim}_{i,T-w+1}^{(c)}) \\ &\text{where } \mathrm{Sim}_{i,j}^{(c)} = \exp\left(-\left\|\boldsymbol{p}_i^{(c)} - \boldsymbol{z}_j\right\|_2^2\right) \in [0,1] \end{aligned} \tag{3}$$

We aggregate similarity scores across all prototypes for each modality, yielding $\mathrm{Sim}_{\text{time}} \in \mathbb{R}^{kC}$ and $\mathrm{Sim}_{\text{text}} \in \mathbb{R}^{k'C}$. Finally, we jointly consider the cross-modal relevance and use a non-negative fusion weight matrix $\boldsymbol{W} \in \mathbb{R}^{C \times (k+k')}$ that translates these scores into class probabilities:

$$\hat{\boldsymbol{y}}_{\text{enc}} = \mathrm{Softmax}\Big(\boldsymbol{W}\left[\mathrm{Sim}_{\text{time}} \,\|\, \mathrm{Sim}_{\text{text}}\right]\Big) \in [0,1]^C. \tag{4}$$

3.2.2. Learning Prototypes toward Better Explanation

Learning Objectives. The learning objectives include three regularization terms that reinforce the interpretability of multi-modal prototypes. In this paper, we focus on a predicting discrete label, where the basic objective is the cross-entropy loss for the prediction drawn from multi-modal explainable artifacts $\mathcal{L}_{\text{CE}} = \sum_{\boldsymbol{x},\boldsymbol{s},\boldsymbol{y}} \boldsymbol{y}\log(\hat{\boldsymbol{y}}_{\text{enc}}) + (1-\boldsymbol{y})\log(1-\hat{\boldsymbol{y}}_{\text{enc}})$. Besides, we encourage a clustering structure of segments in the representation space by enforcing each segment representation to be adjacent to its closest prototype. Reversely, we regularize each prototype to be as close to a segment representation as possible, to help the prototype locate the most evidencing segment. Both regularization terms are denoted as $\mathcal{L}_c$ and $\mathcal{L}_e$, respectively, where we omit the modality and class notations for ease of understanding:

$$\begin{aligned} \mathcal{L}_c &= \sum_{\boldsymbol{z}_j \in \boldsymbol{Z}_{(\cdot)}} \min_{\boldsymbol{p}_i \in \boldsymbol{P}_{(\cdot)}} \|\boldsymbol{z}_j - \boldsymbol{p}_i\|_2^2, \\ \mathcal{L}_e &= \sum_{\boldsymbol{p}_i \in \boldsymbol{P}_{(\cdot)}} \min_{\boldsymbol{z}_j \in \boldsymbol{Z}_{(\cdot)}} \|\boldsymbol{p}_i - \boldsymbol{z}_j\|_2^2 \end{aligned} \tag{5}$$

Moreover, we encourage a diverse structure of prototype representations to avoid redundancy and maintain a compact explanation space, by penalizing their similarities via a hinge loss $\mathcal{L}_d$, with a threshold d_{min} :

$$\mathcal{L}_d = \sum_{i=1}\sum_{j \neq i} \max\left(0, d_{\min} - \left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|_2^2\right) \tag{6}$$

The full objective is written as: $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_1\mathcal{L}_c + \lambda_2\mathcal{L}_e + \lambda_3\mathcal{L}_d$, with hyperparameters λ_1, λ_2, and λ_3 that balance regularization terms towards achieving an optimal and explainable prediction.

Prototype Projection. After learning objectives converge, the multi-modal prototypes are well-regularized and reflect good explanation properties. However, these prototypes are still not readily explainable as they are only close to some exemplar segments in the representation space. Therefore, we perform prototype projection to associate each prototype with a training segment from its own class that preserves $\mathcal{L}_e$ in the representation space, for both time series and text:

$$\boldsymbol{p}_i^{(c)} \leftarrow \arg\min_{\boldsymbol{z}_j \in \boldsymbol{Z}_{(\cdot)}^{(c)}} \left\|\boldsymbol{p}_i^{(c)} - \boldsymbol{z}_j\right\|_2^2, \quad \forall \boldsymbol{p}_i^{(c)} \in \boldsymbol{P}_{(\cdot)}^{(c)} \tag{7}$$

By associating each prototype with a training segment in the representation space, the multi-modal physical meaning is induced. During testing phase, a multi-modal instance will be compared with prototypes across different modalities to infer predictions, where the similarity scores, contribution weights, and prototypes' class information assemble the explanation artifacts for reasoning.

3.3. Explainable Prediction with LLM-in-the-Loop

To further leverage the reasoning and inference capabilities of LLMs in real-world time series contexts, we propose a framework with three interacting LLM agents: a prediction agent $\mathcal{M}_{\text{pred}}$, a reflection agent $\mathcal{M}_{\text{refl}}$, and a refinement agent $\mathcal{M}_{\text{refine}}$. These LLM agents interact with the multi-modal prototype-based encoder $\mathcal{M}_{\text{enc}}$ toward better prediction accuracy and explainability.

3.3.1. Model Synergy for Augmented Prediction

Prediction with Enriched Contexts. The prediction LLM agent $\mathcal{M}_{\text{pred}}$ generates predictions based on the input text $\boldsymbol{s}$. To improve prediction accuracy, the encoder $\mathcal{M}_{\text{enc}}$ supplements $\boldsymbol{s}$ with *case-based explanations*. Specifically, $\mathcal{M}_{\text{enc}}$ selects the ω prototypes that exhibit the highest relevance to any of the textual segments within $\boldsymbol{s}$. Relevance is determined by the similarity scores used in Equation 3. These selected prototypes are then added to the input prompt of $\mathcal{M}_{\text{pred}}$ as explanations, providing richer real-world context and leading to more accurate predictions. The ω prototype-segment pairs, which construct the explanation $\mathbf{expl}_s$ of the input text $\boldsymbol{s}$, are retrieved as follows:

$$\mathbf{expl}_s = \left\{ \left(\boldsymbol{p}_i^{(c)}, \boldsymbol{s}_j \right) : (i, j, c) \in \text{Top-}\omega(\text{Sim}_{\text{text}}) \right\}$$

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

Note that i, j, c denotes the prototype index, segment index, and class index, respectively. As $\mathbf{expl}_{\boldsymbol{s}}$ can contain relevant contextual guidance across multiple classes, it augments the input space and removes semantic ambiguity for prediction agent $\mathcal{M}_{\text{pred}}$. Therefore, the prediction is drawn as $\hat{\boldsymbol{y}}_{\text{LLM}} = \mathcal{M}_{\text{pred}}(\boldsymbol{s}, \mathbf{expl}_{\boldsymbol{s}})$. The prompt for querying the prediction agent $\mathcal{M}_{\text{pred}}$ is provided in Appendix D, Figure 13.

Fused Predictions. We compile the final prediction based on a fusion of both the multi-modal encoder $\mathcal{M}_{\text{enc}}$ and prediction LLM $\mathcal{M}_{\text{pred}}$. Specifically, we linearly combine the continuous prediction probabilities $\hat{\boldsymbol{y}}_{\text{enc}}$ and discrete prediction $\hat{\boldsymbol{y}}_{\text{LLM}}$: $\hat{\boldsymbol{y}} = \alpha \hat{\boldsymbol{y}}_{\text{enc}} + (1 - \alpha)\hat{\boldsymbol{y}}_{\text{LLM}}$, where $\alpha \in [0, 1]$ is the hyperparameter selected from validation data. The encoder $\mathcal{M}_{\text{enc}}$ and prediction agent $\mathcal{M}_{\text{pred}}$ enhance each other based on their unique strengths. The $\mathcal{M}_{\text{enc}}$ is fine-tuned based on explicit supervised signals, ensuring accuracy in capturing temporal and contextual dependencies of multi-modal time series. On the other hand, $\mathcal{M}_{\text{pred}}$ contributes deep semantic understanding drawn from extensive text corpora. By fusing predictions from two distinct perspectives, we achieve a synergistic augmentation toward more accurate and comprehensive predictions for complex multi-modal time series.

Algorithm 1 TimeXL: Explainable Multi-modal Time Series Prediction with LLM Agents

Inputs: Multi-modal time series $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{y})$, 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 evaluation $\text{Eval}(\cdot)$ based on metrics

Training:

Initialize $\boldsymbol{s}_0 = \boldsymbol{s}$, $i = 0$, $\hat{\boldsymbol{y}}_{\text{all}} = \{\}$

while $\text{Eval}(\hat{\boldsymbol{y}}_{\text{all}}, \boldsymbol{y})$ *not pass or iteration* $i < \tau$ **do**

- Train $\mathcal{M}_{\text{enc}}$ using multi-modal data $\mathcal{D}_i = \{(\boldsymbol{x}, \boldsymbol{s}_i, \boldsymbol{y}), \cdots\}$
- Infer explainable prediction $\hat{\boldsymbol{y}}_{\text{enc}}, \mathbf{expl}_{\boldsymbol{s}_i} = \mathcal{M}_{\text{enc}}(\boldsymbol{x}, \boldsymbol{s}_i)$
- Infer LLM prediction $\hat{\boldsymbol{y}}_{\text{LLM}} = \mathcal{M}_{\text{pred}}(\boldsymbol{s}_i, \mathbf{expl}_{\boldsymbol{s}_i})$
- Fuse prediction $\hat{\boldsymbol{y}} = \alpha \hat{\boldsymbol{y}}_{\text{enc}} + (1 - \alpha)\hat{\boldsymbol{y}}_{\text{LLM}}$
- Generate reflection $\text{Refl} = \mathcal{M}_{\text{refl}}(\boldsymbol{y}, \hat{\boldsymbol{y}}_{\text{LLM}}, \boldsymbol{s}_i)$
- Refine text based on reflection $\boldsymbol{s}_{\boldsymbol{i+1}} = \mathcal{M}_{\text{refine}}(\text{Refl}, \boldsymbol{s}_i)$
- Append $\hat{\boldsymbol{y}}$ to $\hat{\boldsymbol{y}}_{\text{all}}$
- Increment i

return $\mathcal{M}_{\text{enc}}$, Refl, $\boldsymbol{s}_{i+1}$

Validation and Testing:

Refinement based on reflection $\boldsymbol{s'} = \mathcal{M}_{\text{refine}}(\text{Refl}, \boldsymbol{s})$

Infer explainable prediction $\hat{\boldsymbol{y}}_{\text{enc}}, \mathbf{expl}_{\boldsymbol{s'}} = \mathcal{M}_{\text{enc}}(\boldsymbol{x}, \boldsymbol{s'})$

Infer LLM prediction $\hat{\boldsymbol{y}}_{\text{LLM}} = \mathcal{M}_{\text{pred}}(\boldsymbol{s'}, \mathbf{expl}_{\boldsymbol{s'}})$

Fuse prediction $\hat{\boldsymbol{y}} = \alpha \hat{\boldsymbol{y}}_{\text{enc}} + (1 - \alpha)\hat{\boldsymbol{y}}_{\text{LLM}}$

3.3.2. Iterative Context Refinement via Reflective Feedback

While the prediction agent $\mathcal{M}_{\text{pred}}$ leverages the explainable artifacts to make informed predictions, it is not inherently designed to fit into the context of multi-modal time series data, which could lead to inaccurate predictions when the quality of textual context is inferior. To tackle this issue, we exploit another two language agents $\mathcal{M}_{\text{refl}}$ and $\mathcal{M}_{\text{refine}}$ to generate reflective feedback and refinements on the context, respectively, for better predictive insights.

Given the prediction $\hat{\boldsymbol{y}}_{\text{LLM}}$ generated by the prediction agent $\mathcal{M}_{\text{pred}}$, the reflection agent $\mathcal{M}_{\text{refl}}$ aims to understand the reasoning behind the implicit prediction logic of $\mathcal{M}_{\text{pred}}$. Specifically, it generates a *reflective feedback*, Refl, by analyzing the input text $\boldsymbol{s}$ and its prediction $\hat{\boldsymbol{y}}_{\text{LLM}}$, against the ground truth $\boldsymbol{y}$, to provide actionable insights for refinement, i.e., $\text{Refl} = \mathcal{M}_{\text{refl}}(\boldsymbol{y}, \hat{\boldsymbol{y}}_{\text{LLM}}, \boldsymbol{s})$. Guided by the feedback, the refinement agent $\mathcal{M}_{\text{refine}}$ refines the previous text $\boldsymbol{s}_i$ into $\boldsymbol{s}_{i+1}$ by selecting and emphasizing the most relevant content, ensuring that important patterns are appropriately contextualized, which is similar to how a domain expert would perform, i.e., $\boldsymbol{s}_{\boldsymbol{i+1}} = \mathcal{M}_{\text{refine}}(\text{Refl}, \boldsymbol{s}_i)$. The prompts for querying $\mathcal{M}_{\text{refl}}$ and $\mathcal{M}_{\text{refine}}$ are provided in Figures 14, 15 16, 17, and discussed in Appendix D.

We finally integrate the refinement via reflection into the optimization loop of our proposed TimeXL, which is summarized in Algorithm 1. Once the textual context is improved, it is used to retrain the multi-modal prototype-based encoder $\mathcal{M}_{\text{enc}}$ for the next iteration. As such, the explanation (*e.g.,* quality of the prototypes) and predictive performance of $\mathcal{M}_{\text{enc}}$ can be improved through this iterative process. Consequently, the prediction agent $\mathcal{M}_{\text{pred}}$ could yield better prediction with more informative inputs, further enhancing the accuracy of $\hat{\boldsymbol{y}}$. We evaluate the trajectory of predictive performance and terminate the iteration if at least an improvement is observed ($\text{Eval}(\cdot)$ pass) when max iteration is reached. Note that, in the testing phase, we use the reflection Refl generated in the best training iteration (evaluated on validation set) to guide $\mathcal{M}_{\text{refine}}$ for context refinement, mimicking how an optimized deep model is applied to testing data.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate methods on four multi-modal time series datasets from three different real-world domains, including weather, finance, and healthcare. The detailed data statistics are summarized in Table 3 of Appendix A.1. The **weather** dataset contains meteorological reports and the hourly time series records of temperature, humidity, air pressure, wind speed, and wind direction in New York City. The task is to predict if it will rain in the next 24 hours, given the last 24 hours of weather records and summary. The **finance** dataset contains the daily record of the raw material prices together with 14 related indices from January 2017 to July 2024. Given the last 5 business days of stock price data and news, the task is to predict if the target price will exhibit an increasing, decreasing, or neutral trend on the next business day. The **healthcare** datasets contain Test-Positive (TP) and Mortality (MT). The Test-Positive dataset consists of the weekly records and healthcare reports of the number of positive specimens for Influenza A and B. The task is to predict if the percentage of respiratory specimens testing positive in the upcoming week for influenza will exceed the average value, given the records and summary in the last 20 weeks. Similarly, the Mortality dataset contains the weekly records and reports of influenza and pneumonia deaths. The task is to predict if the mortality ratio from influenza and pneumonia will exceed the average value, given the records and summary in the last 20 weeks.

Baselines, Evaluation Metrics and Setup We compare TimeXL with state-of-the-art baseline methods for time series prediction. These baselines includes Autoformer (Wu et al., 2021), Dlinear (Zeng et al., 2023), Crossformer (Zhang & Yan, 2023), TimesNet (Wu et al., 2023), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024c), FreTS (Yi et al., 2024), TSMixer (Chen et al., 2023a) and LLM-based methods like LLMTime (Gruver et al., 2023), PromptCast (Xue & Salim, 2023), OFA (Zhou et al., 2023), Time-LLM (Jin et al., 2024) and TimeCMA (Liu et al., 2025), where LLMTime and PromptCast don't need fine-tuning. While these methods are primarily used for time series prediction with continuous values, they can be easily adapted for discrete value prediction. We also evaluate the multi-modal time series methods. Besides the Time-LLM and TimeCMA where input text is used for embedding reprogramming and alignment, we also evaluate Multi-modal PatchTST and Multi-modal iTransformer from (Liu et al., 2024a), as well as TimeCAP (Lee et al., 2025). We evaluate the discrete prediction via F1 score and AUROC (AUC) score, due to label imbalance in real-world time series datasets. We split all datasets for training/validation/testing by a ratio of 6/2/2. We alternate different embedding methods for texts based on its average length, where we use Bert (Kenton & Toutanova, 2019) as the embedding model for weather and healthcare datasets, and sentence transformer (Reimers & Gurevych, 2019) for finance dataset.

4.2. Performance Evaluation

The results of predictive performance are shown in Table 1. It is notable that multi-modal methods generally outperform time series methods across all datasets. These methods include LLM methods (*e.g.,* Time-LLM, TimeCMA) that leverage text embeddings to enhance time series predictions. Moreover, the multi-modal variants (MM-iTransformer and MM-PatchTST) improve the performance of state-of-the-art time series methods, suggesting the benefits of integrating real-world contextual information. Besides, TimeCAP integrates the predictions from both modalities, further improving the predictive performance. TimeXL constantly achieves the highest F1 and AUC scores, consistently surpassing both time series and multi-modal baselines by up to 8.9% of AUC (compared to TimeCAP on Weather dataset). This underscores the advantage of TimeXL, which synergizes multi-modal time series encoder with language agents to enhance interpretability and thus predictive performance in multi-modal time series.

4.3. Explainable Multi-modal Prototypes

Next, we present the explainable multi-modal prototypes rendered by TimeXL, which establishes the case-based reasoning process. Figure 3 shows a subset of time series and text prototypes learned on the weather dataset. The time series prototypes demonstrate the typical temporal patterns aligned with different real-world weather conditions (*i.e.,* rain and not rain). For example, a constant or decreasing humidity at a moderate level, combined with high and steady air pressure, typically indicates a non-rainy scenario. The consistent wind direction is also a sign of mild weather conditions. On the contrary, high humidity, low and fluctuating pressure, along with variable winds typically reveal an unstable weather system ahead. In addition

Table 1. The F1 score (F1) and AUROC (AUC) for TimeXL and state-of-the-art baselines on real-world multi-modal time series datasets.

Datasets →	Weather		Finance		Healthcare (TP)		Healthcare (MT)	
Methods ↓	F1	AUC	F1	AUC	F1	AUC	F1	AUC
DLinear (Zeng et al., 2023)	0.540	0.660	0.255	0.485	0.393	0.500	0.419	0.388
Autoformer (Wu et al., 2021)	0.546	0.590	0.565	0.747	0.774	0.918	0.683	0.825
Crossformer (Zhang & Yan, 2023)	0.500	0.594	0.571	0.775	0.924	0.984	0.737	0.913
TimesNet (Wu et al., 2023)	0.494	0.594	0.538	0.756	0.794	0.867	0.765	0.944
iTransformer (Liu et al., 2024c)	0.541	0.650	0.600	0.783	0.861	0.931	0.791	0.963
TSMixer (Chen et al., 2023a)	0.488	0.534	0.465	0.689	0.770	0.797	0.808	0.931
FreTS (Yi et al., 2024)	0.623	0.688	0.546	0.737	0.887	0.950	0.751	0.762
PatchTST (Nie et al., 2023)	0.592	0.675	0.604	0.795	0.841	0.934	0.695	0.928
LLMTime (Gruver et al., 2023)	0.587	0.657	0.315	0.498	0.802	0.817	0.769	0.803
PromptCast (Xue & Salim, 2023)	0.499	0.365	0.418	0.607	0.727	0.768	0.696	0.871
OFA (Zhou et al., 2023)	0.501	0.606	0.512	0.745	0.774	0.879	0.851	0.977
Time-LLM (Jin et al., 2024)	0.613	0.699	0.589	0.792	0.671	0.864	0.733	0.912
TimeCMA (Liu et al., 2025)	0.636	0.731	0.559	0.727	0.729	0.828	0.693	0.843
MM-iTransformer (Liu et al., 2024a)	0.608	0.689	0.605	0.793	0.926	0.986	0.901	0.990
MM-PatchTST (Liu et al., 2024a)	0.621	0.718	0.619	**0.812**	0.863	0.968	0.780	0.929
TimeCAP (Lee et al., 2025)	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**

to time series, the text prototypes also highlight consistent semantic patterns for different weather conditions, such as the channel-specific (*e.g.*, drier air moving into the area, strengthening of high-pressure system) and overall (*e.g.*, a likelihood of dry weather) descriptions of weather activities. In Appendix C.1, we also present more multi-modal prototypes for the weather dataset in Figure 8, for the finance dataset in Figure 9, and for healthcare datasets in Figures 10 and 12. The results validate that TimeXL provides coherent and informative prototypes from the exploitation of time series and its real-world contexts, which facilitates both prediction and explanation.

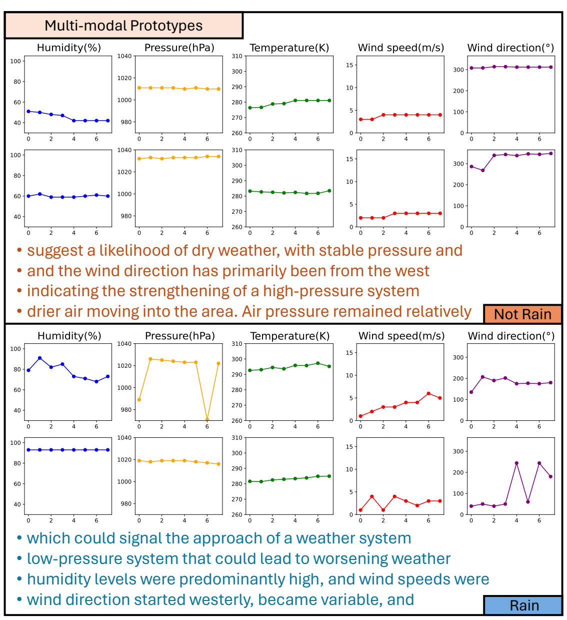

Figure 3. Key time series prototypes and text prototypes learned on weather data. Each row in the figure represents a time series prototype with different channels.

4.4. Multi-modal Case-based Reasoning

Building upon the multi-modal prototypes, we present a case study on the testing set of weather data, comparing the original and TimeXL's reasoning processes to highlight its explanatory capability, as shown in Figure 4. In this case, the original text is incorrectly predicted as not rain. We have three key observations: **(1)** The refinement process filters the original text to emphasize weather conditions more indicative of rain, guided by reflections from training examples. The refined text preserves the statement on stability while placing more emphasis on humidity and wind as key indicators. **(2)** Accordingly, the matched segment-prototype pairs from the original text focus more on temperature stability and typical diurnal variations, while the matched pairs in the refined text highlights wind variability, moisture transport, and approaching weather system, aligning more with rain conditions. **(3)** Furthermore, the reasoning on time series provides a complementary view for assessing weather conditions. The matched time series prototypes identify high humidity and its drop-and-rise trends, wind speed fluctuations and directional shifts, and the declining phase of air pressure fluctuations, all of which are linked to the upcoming rainy conditions. The matched multi-modal prototypes from TimeXL demonstrate its effectiveness in capturing relevant information for both predictive and explanatory analysis. We also provide a case study on finance data in Figure 11, where textual explanations are generated at the granularity of a half-sentence.

4.5. Iterative Analysis

To verify the effectiveness of overall workflow with reflection and refinement LLMs as shown in Figure 1, we conduct an iterative analysis of text quality and TimeXL perfor-

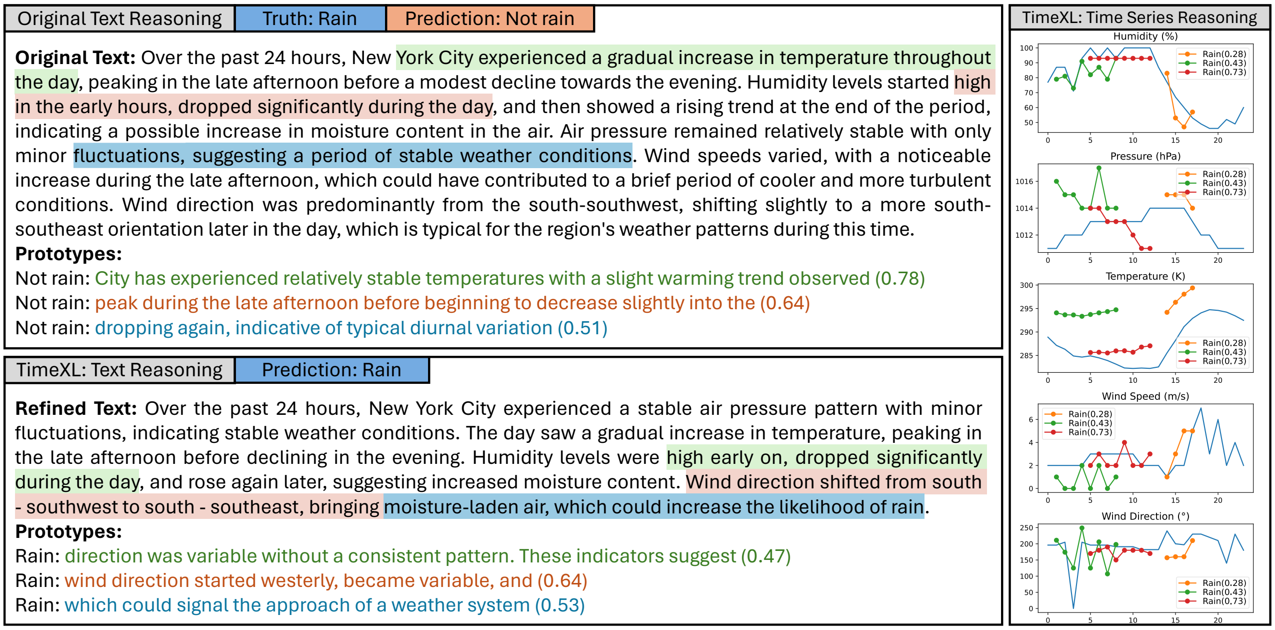

Figure 4. Multi-modal case-based reasoning example on weather data. The left part illustrates the reasoning process for both the original and refined text in TimeXL, with matched prototype-input pairs highlighted in the same color along with their similarity scores. The right part presents the time series reasoning in TimeXL, where matched prototypes are overlaid on the time series.

mance, as shown in Figure 5. Specifically, we evaluate the text quality based on its zero-shot predictive accuracy using an LLM. Notably, the text quality benefits from iteration improvements and mostly saturates after one or two iterations. Correspondingly, TimeXL performance quickly improves and stabilizes with very minor fluctuations. These observations underscore how TimeXL alternates between predictive and reflective refinement phases to mitigate textual noise, thus enhancing its predictive capability.

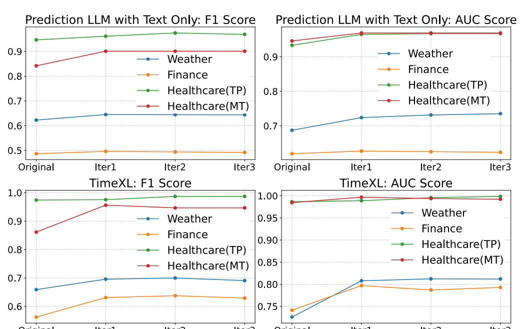

Figure 5. Iterative analysis: the text quality and TimeXL performance over iterations.

Table 2. Ablation studies of TimeXL on real-world multi-modal time series datasets, evaluated by F1 score.

Ablation	Variants	Weather	Finance	TP	MT
Encoder	Multi-modal	0.674	0.619	0.934	0.937
LLM	Time(PromptCast)	0.499	0.418	0.727	0.696
	Text	0.645	0.496	0.974	0.901
	Text + Prototype	0.667	0.544	0.987	0.952
Fusion	Select-Best	0.674	0.619	0.987	0.952
	TimeXL	**0.696**	**0.631**	**0.987**	**0.956**

4.6. Ablation Studies

In this subsection, we present the component ablations of TimeXL, as shown in Table 2, where we have several observations. Firstly, the performance of prediction LLM with text is better than PromptCast (Xue & Salim, 2023), which highlights the importance of contextual information for LLM in a zero-shot prediction scenario. Furthermore, the text prototypes consistently improve the predictive performance of LLM, underscoring the effectiveness of explainable artifacts from the multi-modal encoder, in terms of providing relevant contextual guidance. In addition, the fusion of prediction LLM and multi-modal encoder further boosts the predictive performance that surpasses the best of both multi-modal encoder and prediction LLM. These observations demonstrate the advantage of our framework synergizing the time series model and LLM for mutually augmented prediction. In Appendix B, full results (F1 and AUC) of TimeXL component ablation are provided in Table 4, and other ablations are provided in Figures 6, 7.

5. Conclusions

In this paper, we present TimeXL, an explainable multi-modal time series prediction framework that synergizes a designed prototype-based encoder with three collaborative LLM agents in the loop (prediction, reflection, and refinement) to deliver more accurate predictions and explanations. Experiments on four multi-modal time series datasets show the advantages of TimeXL over state-of-the-art baselines and its excellent explanation capabilities.

6. Impact Statement

This work presents significant advancements in explainable multi-modal time series prediction by integrating time series encoders with large language model-based agents. The broader impact of this work is multifaceted. It has the potential to support high-stakes decision-making in domains such as finance and healthcare by delivering more accurate predictions accompanied by reliable case-based explanations that lead to more robust analyses. No ethical concerns must be considered in our work. The social impact is substantial as it provides a new paradigm for analyzing real-world multi-modal time series data through the integration of emerging AI tools like language agents.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Bamford, T., Coletta, A., Fons, E., Gopalakrishnan, S., Vyetrenko, S., Balch, T., and Veloso, M. Multi-modal financial time-series retrieval through latent space projections. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 498–506, 2023.

Bento, J., Saleiro, P., Cruz, A. F., Figueiredo, M. A., and Bizarro, P. Timeshap: Explaining recurrent models through sequence perturbations. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2565–2573, 2021.

Bian, Y., Ju, X., Li, J., Xu, Z., Cheng, D., and Xu, Q. Multi-patch prediction: Adapting language models for time series representation learning. In *Forty-first International Conference on Machine Learning*.

Chen, S.-A., Li, C.-L., Arik, S. O., Yoder, N. C., and Pfister, T. Tsmixer: An all-mlp architecture for time series forecast-ing. *Transactions on Machine Learning Research*, 2023a.

Chen, Z., Zheng, L. N., Lu, C., Yuan, J., and Zhu, D. Chatgpt informed graph neural network for stock movement prediction. *Available at SSRN 4464002*, 2023b.

Crabbé, J. and Van Der Schaar, M. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*, pp. 2166–2177. PMLR, 2021.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

Deng, A. and Hooi, B. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 4027–4035, 2021.

Dong, Z., Fan, X., and Peng, Z. Fnspid: A comprehensive financial news dataset in time series. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4918–4927, 2024.

Ekambaram, V., Manglik, K., Mukherjee, S., Sajja, S. S. K., Dwivedi, S., and Raykar, V. Attention based multi-modal new product sales time-series forecasting. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3110–3118, 2020.

Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large Language Models Are Zero Shot Time Series Forecasters. In *NeurIPS*, 2023.

Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

Ismail, A. A., Gunady, M., Corrada Bravo, H., and Feizi, S. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

Jia, F., Wang, K., Zheng, Y., Cao, D., and Liu, Y. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 23343–23351, 2024.

Jiang, Y., Yu, W., Song, D., Wang, L., Cheng, W., and Chen, H. Fedskill: Privacy preserved interpretable skill learning via imitation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1010–1019, 2023.

Jiang, Y., Pan, Z., Zhang, X., Garg, S., Schneider, A., Nevmyvaka, Y., and Song, D. Empowering time series analysis with large language models: A survey. In Larson, K. (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 8095–8103. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/895. URL https://doi.org/10.24963/ijcai.2024/895. Survey Track.

Jin, B., Yang, H., Sun, L., Liu, C., Qu, Y., and Tong, J. A treatment engine by predicting next-period prescriptions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1608–1616, 2018.

Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., et al. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Kamalloo, E., Dziri, N., Clarke, C., and Rafiei, D. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5591–5606, 2023.

Kenton, J. D. M.-W. C. and Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, pp. 2. Minneapolis, Minnesota, 2019.

King, R., Yang, T., and Mortazavi, B. J. Multimodal pre-training of medical time series and notes. In *Machine Learning for Health (ML4H)*, pp. 244–255. PMLR, 2023.

Koa, K. J., Ma, Y., Ng, R., and Chua, T.-S. Learning to generate explainable stock predictions using self-reflective large language models. In *Proceedings of the ACM on Web Conference 2024*, pp. 4304–4315, 2024.

Kolodner, J. L. An introduction to case-based reasoning. *Artificial intelligence review*, 6(1):3–34, 1992.

Lee, G., Yu, W., Cheng, W., and Chen, H. Moat: Multi-modal augmented time series forecasting. 2024.

Lee, G., Yu, W., Shin, K., Cheng, W., and Chen, H. Timecap: Learning to contextualize, augment, and predict time series events with large language model agents. In *AAAI*, 2025.

Leung, K. K., Rooke, C., Smith, J., Zuberi, S., and Volkovs, M. Temporal dependencies in feature importance for time series prediction. In *The Eleventh International Conference on Learning Representations*.

Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., Pan, S., and Wen, Q. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6555–6565, 2024.

Liu, C., Xu, Q., Miao, H., Yang, S., Zhang, L., Long, C., Li, Z., and Zhao, R. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *AAAI*, 2025.

Liu, H., Xu, S., Zhao, Z., Kong, L., Kamarthi, H., Sasanur, A. B., Sharma, M., Cui, J., Wen, Q., Zhang, C., et al. Time-mmd: A new multi-domain multimodal dataset for time series analysis. *arXiv preprint arXiv:2406.08627*, 2024a.

Liu, X., McDuff, D., Kovacs, G., Galatzer-Levy, I., Sunshine, J., Zhan, J., Poh, M.-Z., Liao, S., Di Achille, P., and Patel, S. Large language models are few-shot health learners.

Liu, X., Hu, J., Li, Y., Diao, S., Liang, Y., Hooi, B., and Zimmermann, R. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM on Web Conference 2024*, pp. 4095–4106, 2024b.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024c.

Liu, Y., Li, C., Wang, J., and Long, M. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in Neural Information Processing Systems*, 36, 2024d.

Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., and Long, M. Timer: Transformers for time series analysis at scale. *arXiv preprint arXiv:2402.02368*, 2024e.

Liu, Z., Wang, T., Shi, J., Zheng, X., Chen, Z., Song, L., Dong, W., Obeysekera, J., Shirani, F., and Luo, D. Timex++: Learning time-series explanations with information bottleneck. In *Forty-first International Conference on Machine Learning*, 2024f.

Ming, Y., Xu, P., Qu, H., and Ren, L. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 903–913, 2019.

Moroto, Y., Maeda, K., Togo, R., Ogawa, T., and Haseyama, M. Multimodal transformer model using time-series data to classify winter road surface conditions. *Sensors*, 24 (11):3440, 2024.

Ni, J., Chen, Z., Cheng, W., Zong, B., Song, D., Liu, Y., Zhang, X., and Chen, H. Interpreting convolutional sequence model by learning local prototypes with adaptation regularization. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1366–1375, 2021.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Nie, Y., Kong, Y., Dong, X., Mulvey, J. M., Poor, H. V., Wen, Q., and Zohren, S. A survey of large language models for financial applications: Progress, prospects and challenges. 2024.

Niu, K., Zhang, K., Peng, X., Pan, Y., and Xiao, N. Deep multi-modal intermediate fusion of clinical record and time series data in mortality prediction. *Frontiers in Molecular Biosciences*, 10:1136071, 2023.

Qin, Y., Song, D., Cheng, H., Cheng, W., Jiang, G., and Cottrell, G. W. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2627–2633, 2017.

Queen, O., Hartvigsen, T., Koker, T., He, H., Tsiligkaridis, T., and Zitnik, M. Encoding time-series explanations through self-supervised model behavior consistency. *Advances in Neural Information Processing Systems*, 36, 2024.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL http://arxiv.org/abs/1908.10084.

Shi, X., Xue, S., Wang, K., Zhou, F., Zhang, J., Zhou, J., Tan, C., and Mei, H. Language models can improve event prediction by few-shot abductive reasoning. *Advances in Neural Information Processing Systems*, 36, 2024.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. Publisher correction: Large language models encode clinical knowledge. *Nature*, 620(7973):E19, 2023.

Skenderi, G., Joppi, C., Denitto, M., and Cristani, M. Well googled is half done: Multimodal forecasting of new fashion product sales with image-based google trends. *Journal of Forecasting*, 43(6):1982–1997, 2024.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Tonekaboni, S., Joshi, S., Campbell, K., Duvenaud, D. K., and Goldenberg, A. What went wrong and when? instance-wise feature importance for time-series black-box models. *Advances in Neural Information Processing Systems*, 33:799–809, 2020.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models.

Wang, C., Qi, Q., Wang, J., Sun, H., Zhuang, Z., Wu, J., Zhang, L., and Liao, J. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. *arXiv preprint arXiv:2412.11376*, 2024a.

Wang, X., Fang, M., Zeng, Z., and Cheng, T. Where would i go next? large language models as human mobility predictors. *arXiv preprint arXiv:2308.15197*, 2023.

Wang, X., Feng, M., Qiu, J., Gu, J., and Zhao, J. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *arXiv preprint arXiv:2409.17515*, 2024b.

Wang, Z., Duan, Q., Tai, Y.-W., and Tang, C.-K. C3llm: Conditional multimodal content generation using large language models. *arXiv preprint arXiv:2405.16136*, 2024c.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Xing, Z. and He, Y. Multi-modal information analysis for fault diagnosis with time-series data from power transformer. *International Journal of Electrical Power & Energy Systems*, 144:108567, 2023.

Xue, H. and Salim, F. D. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Yang, B. and Wu, L. How to leverage the multimodal ehr data for better medical prediction? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4029–4038, 2021.

Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., and Niu, Z. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.

Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., and Lu, Y. Temporal data meets llm–explainable financial time series forecasting. Technical report, 2023.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zhang, L., Aggarwal, C., and Qi, G.-J. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2141–2149, 2017.

Zhang, W., Deng, Y., Liu, B., Pan, S., and Bing, L. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3881–3906, 2024.

Zhang, X., Zhao, Z., Tsiligkaridis, T., and Zitnik, M. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022.

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Zhao, X., Jia, K., Letcher, B., Fair, J., Xie, Y., and Jia, X. Vimts: Variational-based imputation for multi-modal time series. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 349–358. IEEE, 2022.

Zheng, L., Chen, Z., He, J., and Chen, H. Mulan: Multi-modal causal structure learning and root cause analysis for microservice systems. In *Proceedings of the ACM on Web Conference 2024*, pp. 4107–4116, 2024.

Zhou, T., Niu, P., Sun, L., Jin, R., et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

A. Experimental Settings

A.1. Dataset Statistics

In this subsection, we provide more details of the real-world datasets we used for the experiments. The data statistics are summarized in Table 3, including the meta information (*e.g.,* domain resolution, duration of real-world time series records), the number of channels and timesteps and so on. We used the weather and healthcare datasets in TimeCAP (Lee et al., 2025), and the finance dataset in (Lee et al., 2024).

The **Weather** dataset contains the hourly time series record of temperature, humidity, air pressure, wind speed, and wind direction[1], and related weather summaries in New York City from October 2012 to November 2017. The task is to predict if it will rain in the next 24 hours, given the last 24 hours of weather records and summary.

The **Finance** dataset contains the daily record of the raw material prices together with 14 related indices ranging from January 2017 to July 2024[2], with news articles gathered from S&P Global Commodity Insights. The task is to predict if future prices will increase by more than 1%, decrease by more than 1%, or exhibit a neutral trend on the next business day, given the last 5 business days of stock price data and news.

The healthcare datasets are related to testing cases and deaths of influenza[3]. The **Healthcare (Test-Positive)** dataset consists of the weekly records of the number of positive specimens for Influenza A and B, and related healthcare reports. The task is to predict if the percentage of respiratory specimens testing positive in the upcoming week for influenza will exceed the average value, given the records and summary in the last 20 weeks. Similarly, the **Healthcare (Mortality)** dataset contains the weekly records and healthcare reports of influenza and pneumonia deaths. The task is to predict if the mortality ratio from influenza and pneumonia will exceed the average value, given the records and summary in the last 20 weeks.

Table 3. Summary of dataset statistics.

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%)

A.2. Hyperparameters

First, we provide the hyperparameters of baseline methods. Unless otherwise specified, we used the default hyperparameters from the Time Series Library (Wu et al., 2023). For LLMTime (Gruver et al., 2023), OFA (Zhou et al., 2023), Time-LLM (Jin et al., 2024), TimeCMA (Liu et al., 2025), TimeCAP (Lee et al., 2025), we use their own implementations. For all methods, the dropout rate $\in \{0.0, 0.1, 0.2\}$, learning rate $\in \{0.0001, 0.0003, 0.001\}$. For transformer-based and LLM fine-tuning methods (Wu et al., 2021; Zhang & Yan, 2023; Liu et al., 2024c; Nie et al., 2023; Liu et al., 2025; Jin et al., 2024), the number of attention layers $\in \{1, 2\}$, the number of attention heads $\in \{4, 8, 16\}$. For Dlinear (Zeng et al., 2023), moving average $\in \{3, 5\}$. For TimesNet (Wu et al., 2023) the number of layers $\in \{1, 2\}$. For PatchTST and MM-PatchTST, the patch size $\in \{3, 5\}$ for the finance dataset.

Next, we provide the hyperparameters of TimeXL. The numbers of time series prototypes and text prototypes are $k \in \{5, 10, 15, 20\}$ and $k' \in [5, 10]$, respectively. The hyperparameters controlling regularization strengths are λ_1, λ_2, $\lambda_3 \in [0.1, 0.3]$ with interval 0.05 for individual modality, $d_{\min} \in \{1.0, 1.5, 2.0\}$ for time series, $d_{\min} \in \{3.0, 3.5, 4.0\}$ for text. Learning rate for multi-modal encoder $\in \{0.0001, 0.0003, 0.001\}$ The number of case-based explanations fed to prediction LLM $\omega \in \{3, 5, 8, 10\}$.

[1]https://www.kaggle.com/datasets/selfishgene/historical-hourly-weather-data
[2]https://www.indexmundi.com/commodities
[3]https://www.cdc.gov/fluview/overview/index.html

A.3. Large Language Model

We employed the gpt-4o-2024-08-06 version for GPT-4o in OpenAI API. We use the parameters max_tokens=2048, top_p=1, and temperature=0.7 for content generation (self-reflection and text refinement), and 0.3 for prediction.

A.4. Environment

We conducted all the experiments on a TensorEX server with 2 Intel Xeon Gold 5218R Processor (each with 20 Core), 512GB memory, and 4 RTX A6000 GPUs (each with 48 GB memory).

B. More Ablation Studies

Here we present the full results of TimeXL component ablations in Table 4. In addition to F1 scores, the results of AUC scores consistently demonstrate the importance of contextual information, the effectiveness of prototype from the multi-modal encoder, as well as the advantage of prediction fusion.

We also provide an ablation study on the learning objectives in the TimeXL encoder. The results clearly show that the full objective consistently achieves the best encoder prediction performance, highlighting the necessity of regularization terms that enhance the interpretability of multi-modal prototypes. The clustering (λ_1) and evidencing (λ_2) objectives also play a crucial role in accurate prediction: the clustering term ensures distinguishable prototypes across different classes, while the evidencing term ensures accurate projection onto training data.

Moreover, we assess how the number of matched case-based explanations enhances the prediction LLM, as shown in Figure 7. We conduct experiments on weather and finance datasets, demonstrating that incorporating more relevant case-based explanations consistently improves prediction performance. This further highlights the effectiveness of explainable artifacts in providing meaningful contextual guidance.

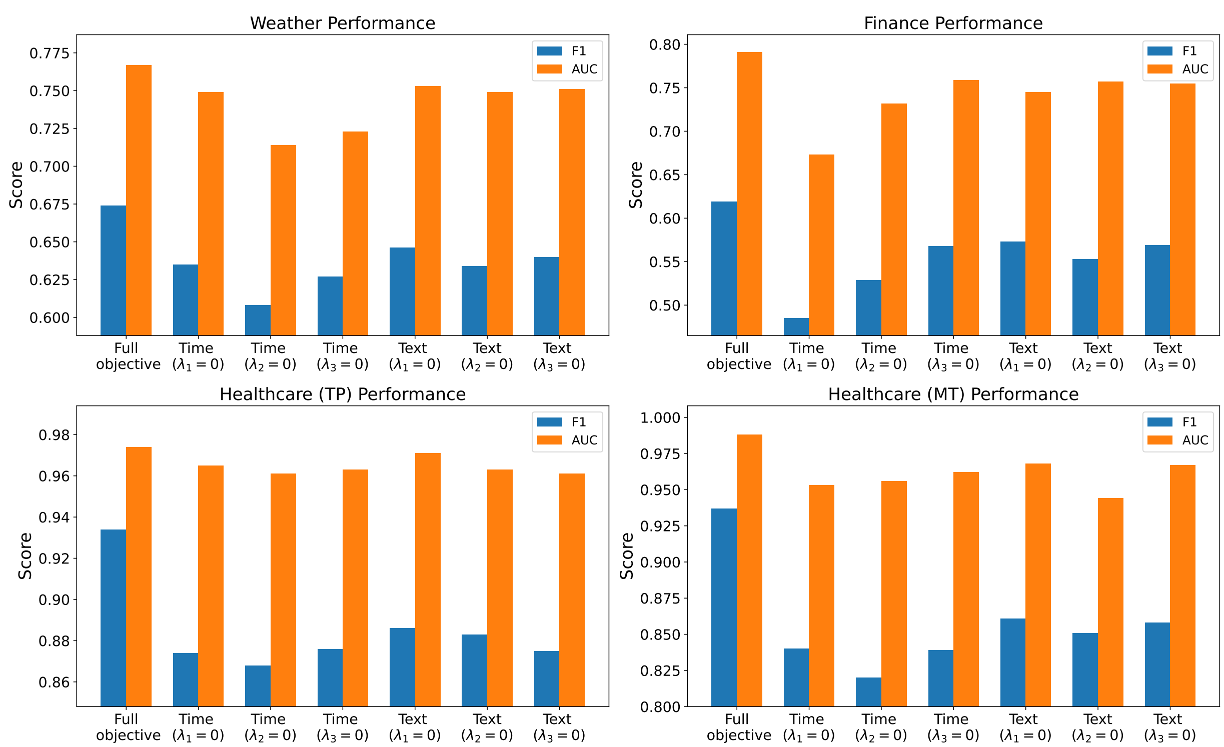

Figure 6. Ablation study of TimeXL encoder learning objectives.

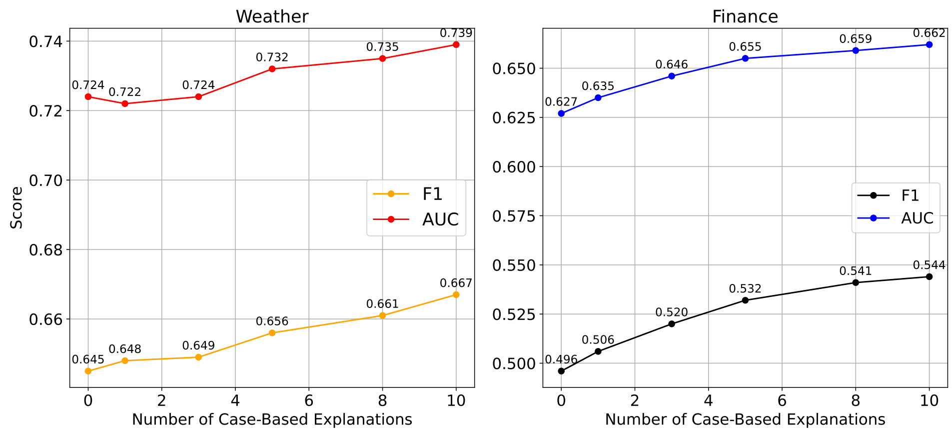

Figure 7. Effect of case-based explanations on LLM prediction performance.

Table 4. Ablation studies of TimeXL on real-world multi-modal time series datasets.

Ablation ↓	**Variants**	**Weather**		**Finance**		**Healthcare** (Test-Positive)		**Healthcare** (Mortality)	
		F1	**AUC**	**F1**	**AUC**	**F1**	**AUC**	**F1**	**AUC**
Encoder	Multi-modal	0.674	0.767	0.619	0.791	0.934	0.974	0.937	0.988
LLM	Time(PromptCast)	0.499	0.365	0.418	0.607	0.727	0.768	0.696	0.871
	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**

C. Explainable Multi-modal Prototypes and Case Study

C.1. Multi-modal Prototypes for All Datasets

We present the learned multi-modal prototypes across all datasets, including Weather (Figure 8), Finance (Figure 9), Healthcare (Test-Positive) (Figure 10), and Healthcare (Mortality) (Figure 12). It is noticeable that the prototypes from both modalities align well with real-world ground truth scenarios, ensuring faithful explanations and enhancing LLM predictions.

C.2. Case-based Reasoning Example on Finance

We provide another case-based reasoning example to demonstrate the effectiveness of TimeXL in explanatory analysis, as shown in Figure 11. In this example, the original text is incorrectly predicted as neutral instead of a decreasing trend of iron ore stock price. We have a few key observations based on the results. First, the refinement LLM filters the original text to emphasize economic and market conditions more indicative of a declining trend, based on the reflections from training examples. The refined text preserves discussions on port inventories and steel margins while placing more emphasis on subdued demand, thin profit margins, and bearish market sentiment as key indicator of prediction. Accordingly, the case-based explanations from the original text focus more on inventory management and short-term stable patterns, while those in the refined text highlight demand contraction, production constraints, and macroeconomic uncertainty, which is more consistent with a decreasing trend. Furthermore, the reasoning on time series provides a complementary view for predicting iron ore price trends. The time series explanations identify declining price movements across multiple indices. In general, the multi-modal explanations based on matched prototypes from TimeXL demonstrate its effectiveness in capturing relevant iron ore market condition for both predictive and explanatory analysis.

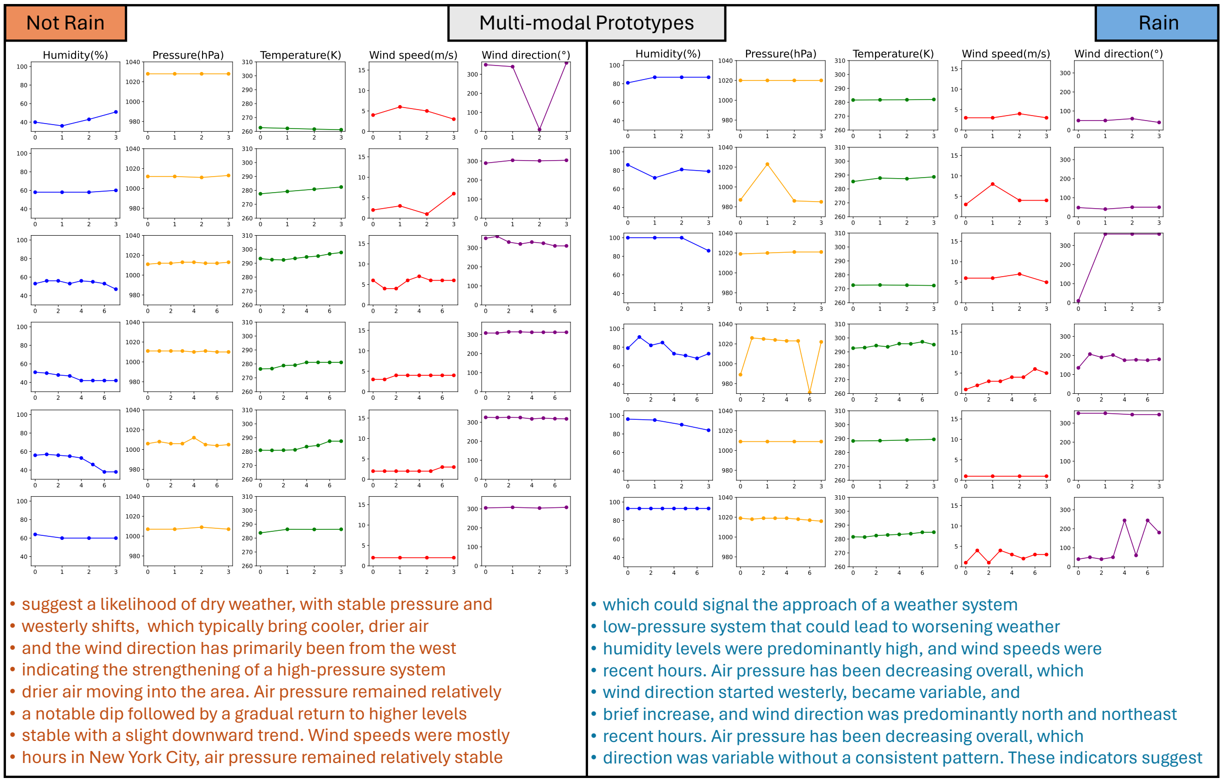

Figure 8. More multi-modal prototypes learned from Weather dataset. Each row in the figure represents a time series prototype.

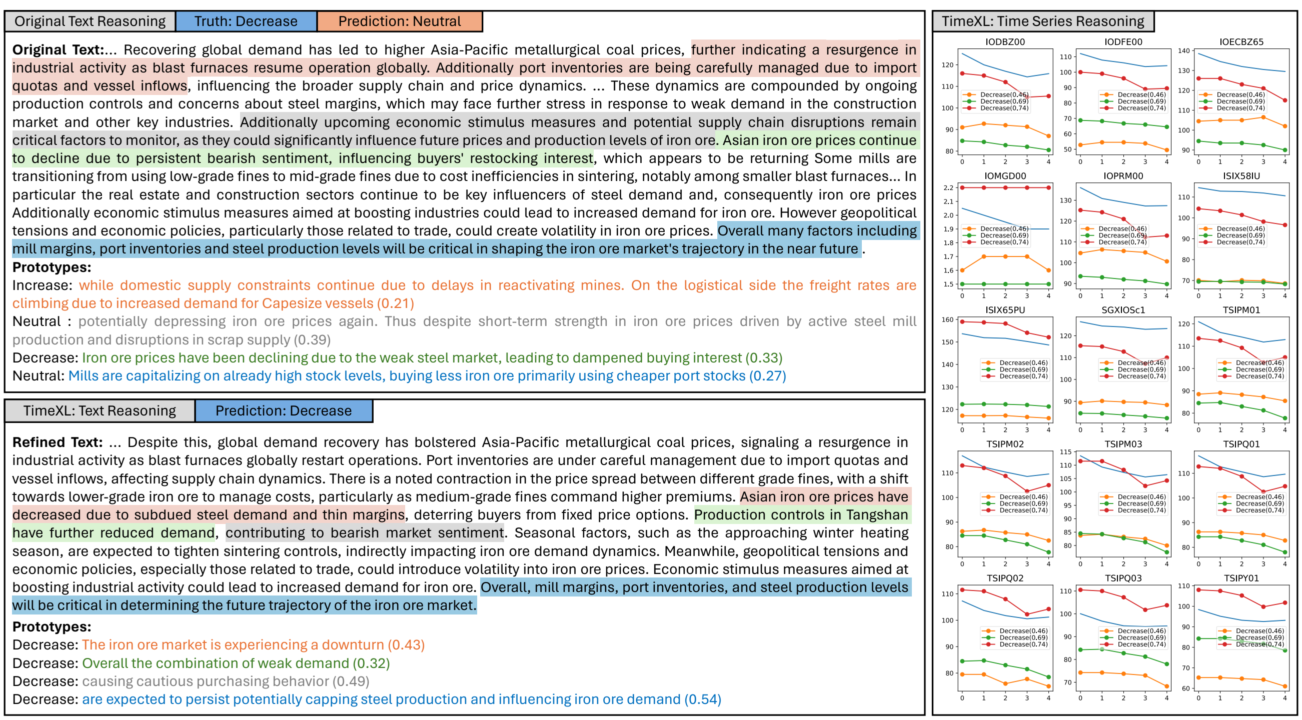

Figure 11. Multi-modal case-based reasoning example on Finance dataset.

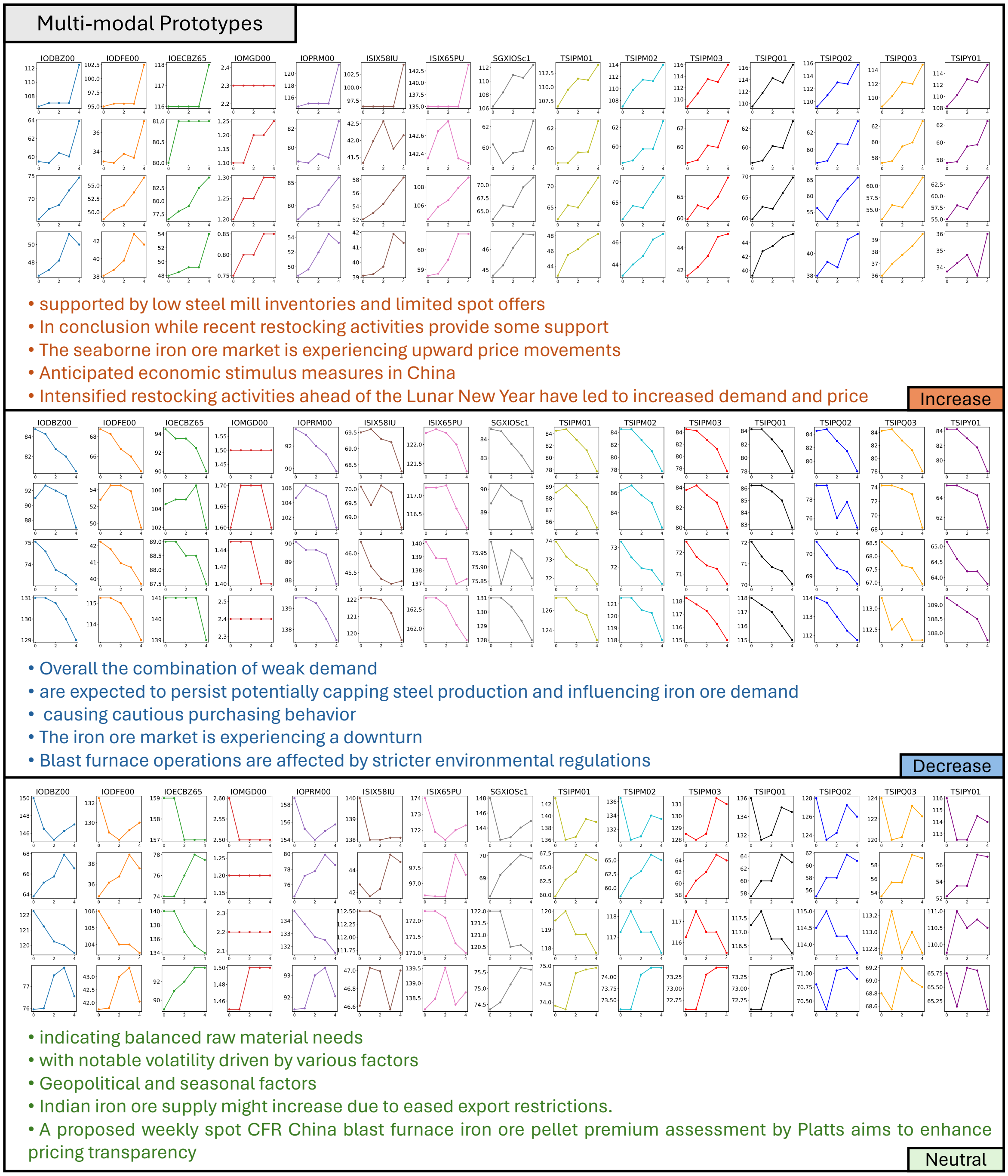

Figure 9. Key multi-modal prototypes learned from Finance dataset. Each row in the figure represents a time series prototype.

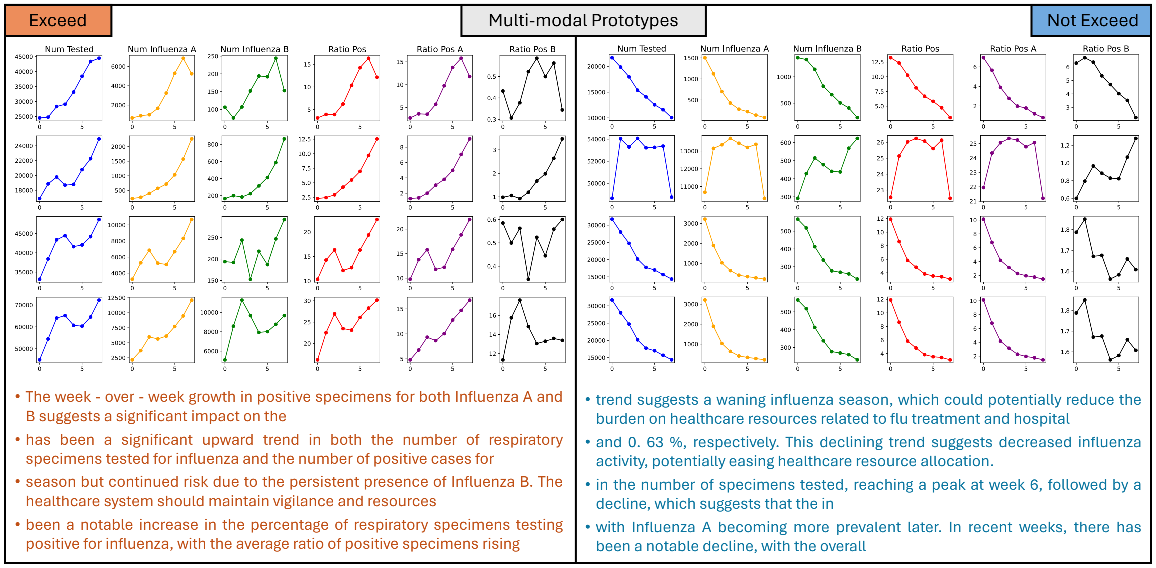

Figure 10. Key multi-modal prototypes learned from Healthcare (Test-positive) dataset. Each row in the figure represents a time series prototype.

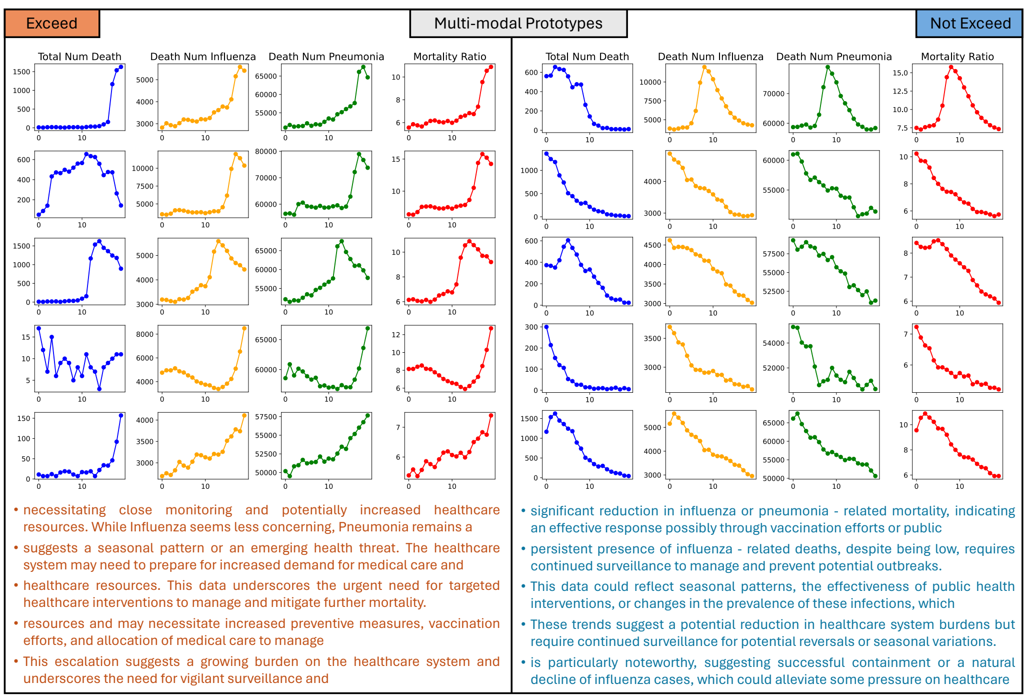

Figure 12. Key multi-modal prototypes learned from Healthcare (Mortality) dataset. Each row in the figure represents a time series prototype.

D. Designed Prompts for Experiments

In this section, we provide our prompts for prediction LLM in Figure 13 (and a text-only variant for comparison, in Figure 18), reflection LLM in Figures 14, 15 16, as well as refinement LLM in Figure 17. Note that we adopt a generate-update-summarize strategy to effectively capture the reflective thoughts from training samples with class imbalances, which is more structured and scalable. We make the whole training texts into batches. First, the reflection LLM generates the initial reflection (Figure 14) by extracting key insights from class-specific summaries, highlighting text patterns that contribute to correct and incorrect predictions. Next, it updates the reflection (Figure 15) by incorporating new training data, ensuring incremental and context-aware refinements. Finally, it summarizes multiple reflections from each class (in Figure 16) into a comprehensive guideline for downstream text refinement. This strategy consolidates knowledge from correct predictions while learning from incorrect ones, akin to the training process of deep models.

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 13. Prompt for prediction LLM

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.

Figure 14. Prompt for reflection LLM: 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.

Figure 15. Prompt for reflection LLM: reflection update

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.

Figure 16. Prompt for reflection LLM: 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.

Figure 17. Prompt for refinement LLM

System Prompt

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

User Prompt

Your task is to [task description]. First, 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 18. Prompt for prediction with text only

E. Reflection Reports for Text Refinement

In this section, we provide the reflection reports after the first iteration, to demonstrate the reflective thoughts by accessing real-world contexts.

Reflection Summary – Weather (New York)

Comprehensive Report on Refining Weather Predictions

The review of weather summaries across various predictive scenarios reveals critical insights that can enhance the accuracy and reliability of weather forecasts. By integrating these refined elements, future weather texts can better anticipate actual outcomes. Below are the summarized guidelines for improving weather prediction texts:

1.Key Indicators for Dry Conditions ("Not Rained" Predictions and Outcomes):

1. **Stable and Slightly Increasing Air Pressure:** Consistent readings with minor upward trends suggest high-pressure systems, indicative of dry weather.
2. **Gentle to Moderate Wind Speeds:** Observations of stable wind speeds without gusts support non-precipitative forecasts.
3. **Variable Wind Directions (Northwesterly/Westerly):** Shifts from southwesterly to northwesterly/westerly directions bring cooler, drier air, reducing rain likelihood.
4. **Decreasing Daytime Humidity:** High early humidity followed by daytime decreases correlates with dry conditions.
5. **Typical Diurnal Temperature Patterns:** Normal temperature variations further support dry forecasts.
6. **Absence of Significant Weather Systems:** Lack of major air pressure or wind pattern changes reinforces stable, dry conditions.

2.Indicators for Rainy Conditions ("Rained" Predictions and Outcomes):

1. **Increasing Humidity Levels:** Significant humidity increases are strong rain indicators.
2. **Decreasing Air Pressure:** A downward pressure trend signals potential rain due to incoming low-pressure systems.
3. **Wind Conditions:**
    1. **Slight Wind Speed Increase:** Often precedes rain, particularly if observed later.
    2. **Easterly/Southeasterly Wind Directions:** Bring moisture-laden air, favoring rain.

3.Common Misinterpretations ("Rained" Predictions with "Not Rained" Outcomes):

1. **Pressure Stability Misinterpretation:** Minor pressure fluctuations without other indicators are weak rain predictors.
2. **Humidity and Fog Confusion:** Differentiate between humidity peaks indicating fog versus those suggesting rain.
3. **Overemphasis on Wind Changes:** Focus on sustained wind patterns rather than minor variations.
4. **Misjudged Temperature Fluctuations:** Evaluate temperature changes alongside corroborative indicators like pressure drops.
5. **Dynamic Weather System Misinterpretation:** Recognize stable atmospheric conditions to avoid false rain predictions.

4.Overlooked Conditions ("Not Rained" Predictions with "Rained" Outcomes):

1. **Humidity Peaks and Variability:** High peaks or fluctuations often precede rain; they shouldn't be underestimated.
2. **Wind Direction Shifts:** Changes to southerly or easterly directions can signal moisture influx.
3. **Temperature and Humidity Interactions:** Combined changes may offset stable pressure, indicating rain potential.
4. **Stable Pressure with Other Variables:** Consider pressure stability with high humidity and wind changes collectively.

By focusing on these detailed indicators and refining the interpretation of weather elements, future weather texts can improve prediction quality, providing more accurate and reliable forecasts.

Figure 19. Reflection summary for text refinement of Weather dataset.

Reflection Summary – Finance (Raw Material)

Comprehensive Guidelines for Enhancing Financial Text Quality in Predictive Analysis

To improve the accuracy of financial predictions, it's crucial to refine the selection and emphasis of content in financial summaries. Based on reflections from past predictions, the following guidelines highlight the essential elements to consider for optimizing text quality:

1.Key Indicators for Price Trends:

1. For predicting **increases**, focus on indicators such as robust demand from major consumers (e.g., Chinese steel mills), strong steel margins, and preferences for high-grade iron ore. Also, consider tight supply chains, low inventories, and economic stimulus measures.
2. For **decreases**, emphasize rising inventory levels, weak demand, and operational adjustments at mills. Bearish market sentiment and geopolitical tensions should be noted as well.
3. For **neutral** outcomes, identify market equilibrium indicators like balanced supply-demand dynamics, stable mill operations, and moderate buyer behavior.

2.Contextual Influences:

1. Consider the impact of seasonal trends, restocking activities, and economic stimuli on demand fluctuations. Recognize how these factors may cause temporary market shifts rather than long-term trends.
2. Analyze global trade and supply chain disruptions to assess their potential to cause short-term volatility or stability rather than sustained changes.

3.Regulatory and Policy Factors:

1. Understand the implications of environmental regulations and geopolitical policies on supply and demand. These elements can significantly alter market dynamics and should be carefully integrated into analyses.

4.Market Sentiment and Related Markets:

1. Assess market sentiment and futures movements, recognizing that positive futures alignments often indicate underlying demand. Consider interconnected markets, such as steel and coal, for their influence on iron ore demand.

5.Strategic and Operational Adjustments:

1. Pay attention to how mills and buyers adjust operations and procurement strategies in response to market conditions. These adjustments can affect demand patterns and price stability.

6.Avoiding Common Misinterpretations:

1. Avoid overemphasizing short-term trends as indicators of sustained market shifts. Be wary of speculative influences and temporary policy announcements that may not have immediate or lasting effects.
2. Recognize high inventory levels as potential strategic buffers and consider their role in stabilizing prices during off-peak periods.

By integrating these guidelines, financial summaries can provide a more comprehensive and balanced analysis, enhancing the predictive accuracy of unknown market outcomes. The focus should remain on understanding and articulating the complexities of market dynamics to better align predictions with actual trends.

Figure 20. Reflection summary for text refinement of Finance dataset.

Reflection Summary – Healthcare (Test-Positive)

Comprehensive Report on Improving Healthcare Text Quality for Predictive Accuracy
Introduction
This report synthesizes reflections from past analyses of healthcare summaries concerning influenza positivity rates. The goal is to enhance the quality of these texts to improve predictive accuracy for future assessments where actual outcomes are unknown.
Key Indicators for Accurate Predictions
1.Declining Trends for "Not Exceed" Predictions
1. **Emphasis on Decline**: Correct predictions for outcomes that did not exceed the average often highlighted a clear decline in positivity rates. Terms such as "notable decline," "steady decrease," and "significant reduction" are crucial.
2. **Comparative Analysis**: A pronounced reduction in Influenza A and B, particularly Influenza A, supports accurate assessments.
3. **Testing Volume and Ratios**: A decrease in testing volume, coupled with positivity rates remaining below the average, strengthens predictions.
4. **Low Positivity Percentages**: Explicit references to recent positivity rates consistently falling below the long-term average are vital.

2.Indicators of "Exceed" Predictions
1. **Marked Increases**: Substantial increases in positivity rates, highlighted by specific numerical comparisons, indicate an "exceed" outcome.
2. **Influenza Strain Dominance**: Details on the predominance of Influenza A or B, with their impact on overall positivity rates, are significant.
3. **Rising Testing Volumes**: Increased specimen testing, peaking with positivity rates, suggests heightened incidence and surveillance.
4. **Healthcare System Impact**: References to increased hospitalizations and medical service demand provide context and validate exceeding predictions.
5. **Temporal Patterns**: Tracking weekly peaks aids in understanding and predicting heightened influenza activity.

Common Misinterpretations
•**Relative vs. Absolute Increases**: Misinterpretations often arise from conflating relative increases with exceeding the average positivity rate. It is essential to compare these increases explicitly with historical averages.
•**Contextual Misalignment**: Phrases highlighting rises in positivity rates without contextual alignment to the average threshold can mislead predictions.
Guidelines for Text Refinement
•**Integrate Clear Comparisons**: Ensure summaries incorporate direct comparisons to historical averages to avoid misjudgments.
•**Maintain Trend Clarity**: Differentiate clearly between increasing trends and those that actually surpass the average positivity rate.
•**Include Contextual Analysis**: Provide context for increases in positivity rates by detailing the impact on healthcare resources and system strain.
•**Highlight Temporal Data**: Incorporate temporal patterns and weekly peaks to aid in nuanced predictive analysis.
By adhering to these guidelines, healthcare texts can be refined to enhance predictive accuracy, ultimately aiding in proactive healthcare planning and resource allocation.

Figure 21. Reflection summary for text refinement of Healthcare (Test-Positive) dataset.

Reflection Summary – Healthcare (Mortality)

Reflection Summary Report for Healthcare Text Refinement:
1.Correct 'Not Exceed' Predictions:
1. Key indicators of accurate predictions include phrases such as "general downward trend," "consistent decline," and "significant decrease" in mortality ratios, particularly from Influenza or Pneumonia.
2. Terms like "below the average ratio" and "remained under the historical benchmark" effectively signal outcomes that do not surpass average mortality ratios.
3. References to "improvement in management" and "effective public health interventions" highlight successful control measures that contribute to lower mortality figures.
4. Acknowledging a "persistent burden" of Pneumonia amidst declines provides a nuanced understanding of stability in disease impact.

2.Correct 'Exceed' Predictions:
1. Accurate forecasts are marked by phrases such as "concerning upward trend," "significant increase," and "notable spike," indicating rising mortality ratios.
2. Quantitative expressions like "dramatic rise" in deaths and comparisons exceeding specific averages effectively contextualize high mortality trends.
3. Descriptions of peaks, potential outbreaks, and healthcare strain underscore the urgency and resource implications, enhancing predictive accuracy.

3.Incorrect 'Exceed' Predictions with 'Not Exceed' Outcomes:
1. Misinterpretations arise from emphasizing upward trends without aligning them with final ratios below the 7.84% threshold.
2. Overemphasis on isolated peaks without considering their unsustained nature leads to overestimations.
3. For improved accuracy, texts should differentiate between temporary increases and sustained trends, and focus on cumulative ratios relative to average benchmarks.

4.Incorrect 'Not Exceed' Predictions with 'Exceed' Outcomes:
1. Underestimations often occur by not fully considering the impact of recent historical peaks or seasonal outbreaks, even when declines are present.
2. Incorrect predictions result from neglecting cumulative effects of influenza and pneumonia, especially when one disease overshadows the other.
3. Future accuracy can be enhanced by emphasizing recent peaks, seasonal patterns, and comprehensive disease impacts.

Guidelines for Future Healthcare Texts:
•Prioritize clear identification of trends and ratios, ensuring they are contextualized against historical averages and thresholds.
•Distinguish between short-term fluctuations and sustained trends, emphasizing cumulative impacts over isolated data points.
•Integrate insights from public health interventions and management effectiveness to provide a holistic view of disease control.
•Consider seasonal patterns and recent historical data to anticipate potential outbreaks or shifts in disease prevalence.
•Ensure quantitative comparisons are explicit and supported by comprehensive data analysis to avoid misinterpretations.
This structured approach will aid in crafting accurate healthcare summaries, enhancing predictive accuracy and improving the quality of healthcare texts for future analyses.

Figure 22. Reflection summary for text refinement of Healthcare (Mortality) dataset.

F. TimeXL for Regression-based Prediction: A Finance Demonstration

In this section, we provide a demonstration of using TimeXL for regression tasks. Two minor modifications adapt TimeXL for continuous value prediction. First, we add a regression branch in the encoder design, as shown in Figure 23. On the top of time series prototype layers, we reversely ensemble each time series segment representation as a weighted sum of time series prototypes, and add a regression head for prediction. Accordingly, we add another regression loss term (*i,e.,* MSE and MAE) to the learning objectives. Second, we adjust the prompt for prediction LLM by adding time series inputs and requesting continuous forecast values, as shown in Figure 24. As such, TimeXL is equipped with regression capability.

We perform the experiments on the same finance dataset, where the target value is the raw material stock price. We provide the performance of the LLM-based prediction as well as TimeXL, as preliminary results, as shown in Table 5. It can be seen that the text modality introduces complementary information, which improves both trend and price value prediction. Besides, the identified prototypes provide contextual guidance, which further improves the predictive performance. TimeXL yields the best results, underscoring the efficacy of mutually augmented prediction. Additionally, we provide a visualization comparing the predicted trend and stock price values, with ground truths for reference.

Table 5. Performance of TimeXL for regression tasks.

Variants	F1	AUC	RMSE	MAE	MAPE(%)
Time(PromptCast)	0.418	0.607	4.728	3.227	2.306
Text + Time	0.520	0.651	4.848	3.167	2.238
Text + Time + Prototype	0.571	0.687	4.688	3.095	2.193
TimeXL	**0.626**	**0.785**	**4.608**	**3.077**	**2.186**

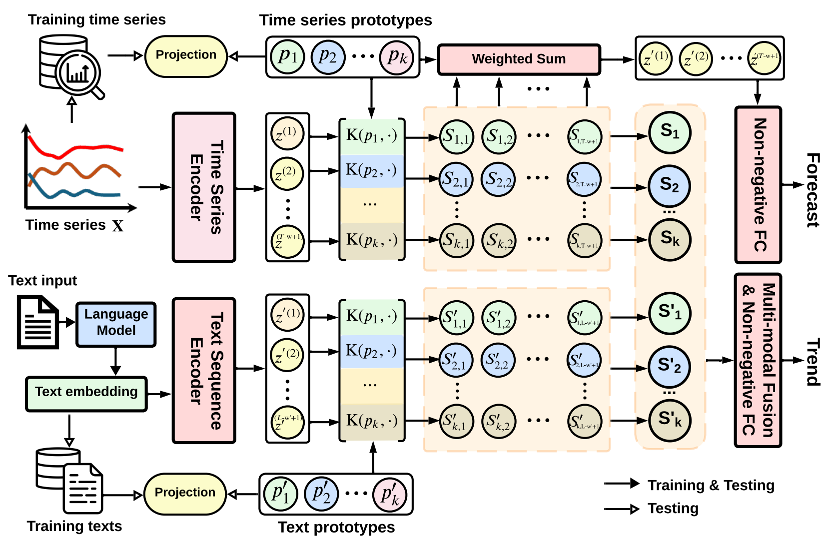

Figure 23. Multi-modal prototype-based encoder design in TimeXL for regression tasks.

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]

Finally, review the [domain] record of [situation] : [time series values]

Based on your understanding, predict the outcome of [situation], followed by the value of [domain] record. Respond your prediction with [options] and [numerical value]. Response should not include other terms.

Figure 24. Prompt for regression tasks.

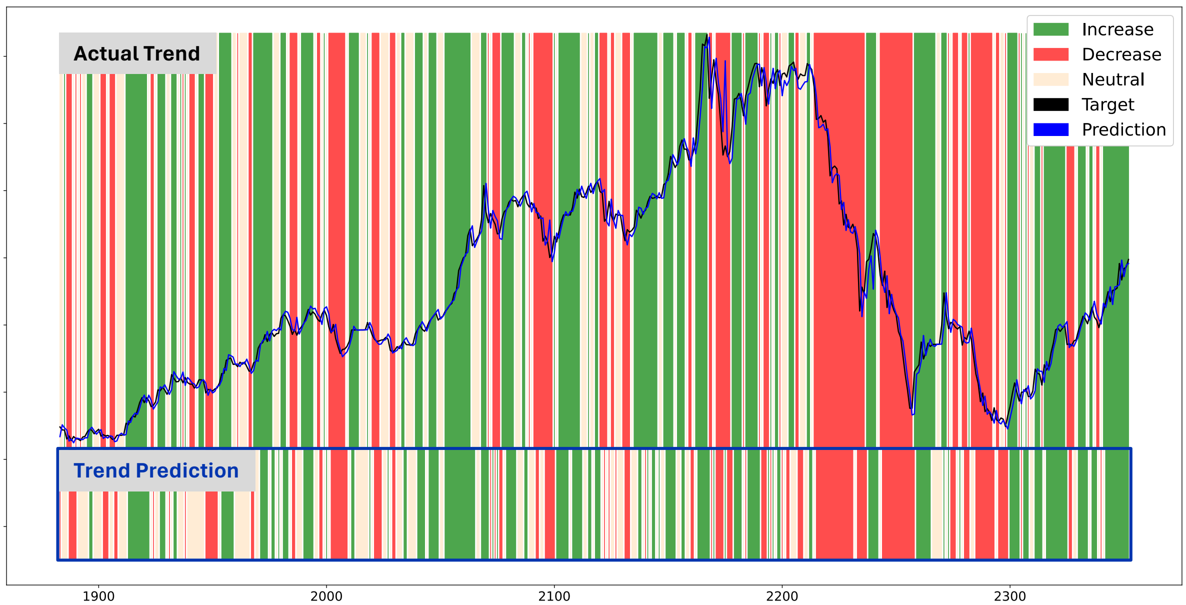

Figure 25. Visualization of TimeXL regression and trend prediction on Finance dataset.