Enhancing Time Series Forecasting via Multi-Level Text Alignment with LLMs

Taibiao Zhao, Xiaobing Chen, and Mingxuan Sun✉

Louisiana State University, Baton Rouge, LA 70803, USA
{tzhao3,xchen87,msun11}@lsu.edu

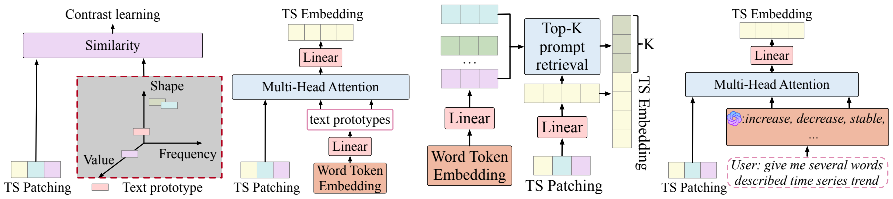
Abstract. The adaptation of large language models (LLMs) to time series forecasting poses unique challenges, as time series data is continuous in nature, while LLMs operate on discrete tokens. Despite the success of LLMs in natural language processing (NLP) and other structured domains, aligning time series data with language-based representations while maintaining both predictive accuracy and interpretability remains a significant hurdle. Existing methods have attempted to reprogram time series data into text-based forms, but these often fall short in delivering meaningful, interpretable results. In this paper, we propose a multi-level text alignment framework for time series forecasting using LLMs that not only improves prediction accuracy but also enhances the interpretability of time series representations. Our method decomposes time series into trend, seasonal, and residual components, which are then reprogrammed into component-specific text representations. We introduce a multi-level alignment mechanism, where component-specific embeddings are aligned with pre-trained word tokens, enabling more interpretable forecasts. Experiments on multiple datasets demonstrate that our method outperforms state-of-the-art models in accuracy while providing good interpretability. Our code is available at `https://github.com/ztb-35/MLTA`.

Keywords: Time Series · alignment · LLMs.

1 Introduction

Large language models (LLMs), trained on vast and diverse text corpora, provide a powerful foundation for various downstream tasks, requiring only minimal task-specific prompt engineering or fine-tuning. This flexibility has sparked a growing interest in leveraging LLMs for time series analysis. For example, methods like Promptcast [14] and LLMTime [5] reformulate numerical inputs and outputs into prompts, treating time series forecasting as a sentence-to-sentence task, which enables the direct application of LLMs. Meanwhile, approaches like TEMPO [2] and GPT4TS [16] take a different route by fine-tuning pre-trained LLMs, modifying components such as the Add&Norm layers and positional embeddings, further demonstrating LLMs' adaptability for time series forecasting.

Despite their potential, the benefits of LLMs in time series forecasting depend on the effective alignment between time series data and natural language



(a) (b) (c) (d)

Fig. 1: Cross-modality time series embeddings of (*a*) contrast learning of text-prototype-aligned time series embeddings, (*b*) text prototypes reprogramming, (*c*)semantic informed prompt and and (*d*) proposed anchors alignment of our multi-level alignment.

modalities. TEST [10] maps time series to word embeddings via contrastive alignment (Figure 1a), while TimeLLM [6] reprograms patches into text prototypes (Figure 1b). S^{2}IP-LLM [8] (Figure 1c) employs semantic prompts to bridge embeddings and text. These approaches, however, primarily achieve a "time series→pattern→text" transformation to activate LLMs for time series tasks. This process often leads to unexpected outcomes. For example, the embedding of a subsequence with an upward trend may be misaligned with a word representing a decline or with a word that doesn't capture the trend at all. As a result, the challenge remains to fully unlock LLMs' capabilities for general time series forecasting in a way that is both accurate and interpretable.

In this paper, we address the challenge of interpretability in LLM-based time series forecasting by developing an interpretable multi-level text alignment framework while preserving the backbone model. Our approach consists of two key principles: (a) modeling time series components—trend, seasonality, and residuals, and (b) deriving interpretable explanations via multi-level text alignment. Using additive decomposition methods, like LOESS [4], we decompose time series into three additive components, reprogramming them into component-specific anchors for better LLM alignment (Figure 1d). Additionally, we employ component-specific prompts to guide the generation of learnable continuous vector representations that encode temporal knowledge of each component.

In summary, the main contributions of this paper are as follows: (1) We propose an interpretable multi-level text alignment framework for time series forecasting using LLMs while keeping the backbone model unchanged. (2) Our method leverages this multi-level alignment to map decomposed time series components—trend, seasonality, and residuals—into distinctive, informative joint representations. The aligned trend-specific anchors enhance the interpretability of LLMs, while the aligned seasonality and residual prototypes improve the overall representation of the input time series. (3) Experimental results on multiple datasets validate the superiority of our model over state-of-the-art approaches, highlighting the effectiveness of interpretable multi-level text alignment.

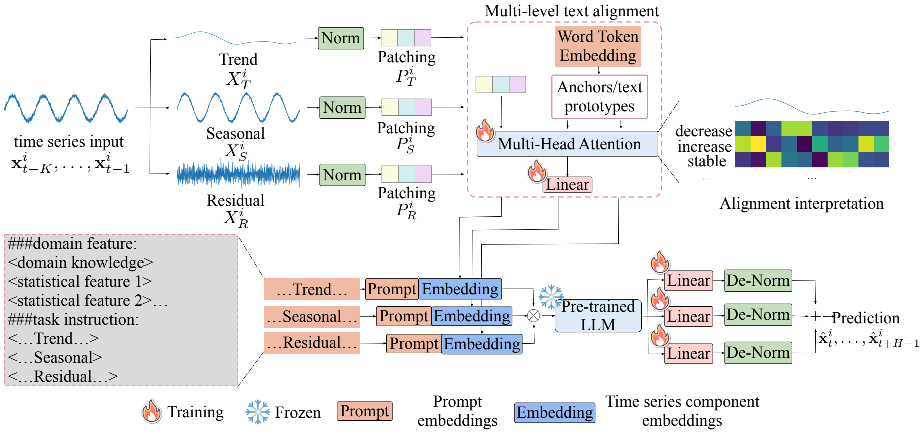


Fig. 2: The architecture of the proposed multi-level aligned embeddings begins with the decomposition of the input time series into three components: trend, seasonal, and residual. These components, tokenized and embedded, are reprogrammed using anchors and condensed text prototypes to align time series data with word tokens. Component-specific prefixed prompts are added to guide the transformation of input patches. Outputs from the LLM are projected, denormalized, and summed to generate the final prediction.

2 Related work

Recent advancements in LLMs like GPT-2 [9] and LLaMA [11] have enabled new possibilities for time series modeling. Efforts to adapt LLMs for this domain include converting time series into text [14], tokenizing data into patches for fine-tuning [5, 16], and leveraging retrieval-based prompts [2, 8]. However, these approaches often lack interpretability, treating time series as mere token sequences and ignoring temporal structures. This misalignment can lead to inaccurate predictions and reduced model transparency, particularly for multivariate time series. To address this, we propose an interpretable multi-level text alignment framework that preserves the integrity of pre-trained LLMs while effectively aligning time series components with anchors.

A key challenge in adapting LLMs for time series forecasting is aligning continuous data with discrete token embeddings. Inspired by prototype-level contrast methods, works like [10] select text embeddings as prototypes (Figure 1a), while [6] reprogram time series using source modality prompts without altering the backbone LLM (Figure 1b). These methods follow a "time series → pattern → text" paradigm, but their arbitrary prototype selection may not capture the true characteristics of the data [10]. More detailed related works see full paper [1]. To overcome this, our approach uses time series-specific anchors (e.g., "increase, decrease, stable") to guide token embedding learning (Figure 1d), enhancing both interpretability and forecasting performance.

3 Methodology

Our approach enhances the interpretability of LLMs for time series data using multi-level aligned embeddings. As shown in Figure 2, the framework consists of four core modules: (1) time series decomposition, (2) multi-level text alignment, (3) component-specific prompts, and (4) output projection. A multivariate time series is first partitioned into N univariate series, each processed independently as $\mathbf{X}^{(i)} \in \mathbb{R}^{1\times L}$. The series undergoes decomposition, normalization, patching, and embedding before alignment with anchor points and text prototypes. To enhance reasoning, we introduce component-specific prompts, enabling the model to generate meaningful representations, which are projected through a linear layer to produce the final forecasts, $\hat{\mathbf{x}}_t^{(i)}, \ldots, \hat{\mathbf{x}}_{t+H-1}^{(i)}$. We utilize GPT-2 [9], employing its first six layers as the backbone for forecasting without fine-tuning.

3.1 Problem statement

Given the previous L observed values, the multivariate time series forecasting is predicts the values for the next H timestamps. It can be formally as:

$$\hat{\mathbf{x}}_t^{(i)}, \ldots, \hat{\mathbf{x}}_{t+H-1}^{(i)} = F\left(\mathbf{x}_{t-L}^{(i)}, \ldots, \mathbf{x}_{t-1}^{(i)}; \mathbf{V}^{(i)}\right), \tag{1}$$

where $\hat{\mathbf{x}}_t^{(i)}, \ldots, \hat{\mathbf{x}}_{t+H-1}^{(i)}$ is the vector of H-step prediction from timestamp t of channel i corresponding to the i-th feature. Given the historical values $\mathbf{x}_{t-L}^{(i)}, \ldots,$ $\mathbf{x}_{t-1}^{(i)}$, a large language model F uses prompt $\mathbf{V}^{(i)}$ to make these predictions. Leveraging the strong reasoning capabilities of pre-trained LLMs, we aim to align time series data with text to enable LLMs to interpret the input series and accurately forecast the H future steps, with the overall objective expressed as:

$$\frac{1}{H}\sum_{h=1}^{H} \|\hat{\mathbf{x}}_{t-1+h}^{(i)} - \mathbf{x}_{t-1+h}^{(i)}\|^2. \tag{2}$$

3.2 Time series input decomposition

For time series data, decomposing complex inputs into meaningful components such as trend, seasonal, and residual elements can help optimally extract valuable information. In this paper, given the input $X \in \mathbb{R}^{N\times L}$, where N is the feature size and L is the length of the time series, the additive decomposition can be represented as:

$$X^{(i)} = X_T^{(i)} + X_S^{(i)} + X_R^{(i)}, \tag{3}$$

where i refers to the feature index for multivariate time series input. The trend component $X_T \in \mathbb{R}^{N\times L}$, capturing the underlying long-term patterns in the data, is expressed as $X_T = \frac{1}{m}\sum_{j=-k}^{k} X_{t+j}$, where $m = 2k + 1$ and k is the averaging step size. The seasonal component $X_S \in \mathbb{R}^{N\times L}$ reflects the repeating short-term cycles and can be estimated after removing the trend. The residual component $X_R \in \mathbb{R}^{N\times L}$ represents the remainder. Additive decomposition can be performed using various methods. A common approach first extracts the long-term trend via moving averages, estimates the seasonal component by averaging the detrended series, and obtains the residual by subtraction. Another widely used method is Seasonal-Trend decomposition using Loess (STL) [4]. The decomposition method in this paper is chosen based on validation results.

Following the approach outlined in [7], we patch the decomposed components of the time series. Specifically, for the i-th normalized trend component, we obtain the patched token $\mathbf{P}_T^{(i)} \in \mathbb{R}^{L_P\times K}$, where L_P represents the patch length and $K = \lfloor \frac{(L-L_P)}{s} \rfloor + 2$ denotes the number of patches, with s is the stride. Similarly, we apply this patching process to the seasonal and residual components, obtaining patched tokens $\mathbf{P}_S^{(i)}$ and $\mathbf{P}_R^{(i)}$, respectively. These patches are then fed into the multi-level text alignment to produce aligned time series embeddings.

3.3 Multi-level text alignment

We reprogram patch embeddings into the LLMs' pre-training space to align time series and natural language modalities, enhancing the backbone's reasoning capabilities. A naive approach aligns time series and text embeddings via similarity estimation, placing time series tokens near relevant text descriptors (e.g., up, down, stable). However, the pre-trained word embedding space is vast and dense, and selecting appropriate text prototypes is often arbitrary, sometimes involving unrelated words or clusters [6, 8, 10], leading to poor interpretability.

In this work, we propose multi-level text alignment to enhance the interpretability of LLMs on time series forecasting. We first decompose the time series into trend, seasonal, and residual and align the trend X_T with selected trend-specific anchors $\mathbb{W}_{trend}$. However, accurately defining anchors for the seasonal and residual components is challenging. To address this, we reprogram seasonal X_S and residual X_R using pre-trained word embeddings $\mathbf{E} \in \mathbb{R}^{V\times D}$ in the backbone, where V is the vocabulary size, D is the hidden dimension of the pre-trained LLM. Directly leveraging $\mathbf{E}$ will result in large and potentially dense reprogramming space. We adapt linearly probing $\mathbf{E}$. The text prototypes of seasonal and residual denoted as $\mathbf{E}^{'}_{seasonal} \in \mathbb{R}^{V^{'}_{seasonal}\times D}$ and $\mathbf{E}^{'}_{residual} \in \mathbb{R}^{V^{'}_{residual}\times D}$, where $V^{'}_{seasonal} < V^{'}_{residual} \ll V$ because the residual is more inconsistent and variable compared to the seasonal.

As illustrated in the top-right of Figure 2, our multi-level text alignment aims to give a connection between anchors and trend patches. The selected anchors are sparse. We reprogram seasonal and residual with text prototypes connecting time series patches with a more dense reprogramming space. To realize this, we employ a multi-head cross-attention layer for each component. Specifically, for i-th input feature, we define query matrices $\mathbf{Q}_T^{(i)} = \mathbf{P}_T^{(i)}\mathbf{W}_T^{Q(i)}$, key matrices $\mathbf{K}_T^{(i)} = \mathbb{W}_{trend}\mathbf{W}_T^{K(i)}$, value matrices $\mathbf{V}_T^{(i)} = \mathbb{W}_{trend}\mathbf{W}_T^{V(i)}$ for trend; query matrices $\mathbf{Q}_S^{(i)} = \mathbf{P}_S\mathbf{W}_S^{Q(i)}$, key matrices $\mathbf{K}_S^{(i)} = \mathbf{E}^{'}_{seasonal}\mathbf{W}_S^{K(i)}$, value matrices

$\mathbf{V}_S^{(i)} = \mathbf{E}^{'}_{seasonal}\mathbf{W}_S^{V(i)}$ for seasonal; query matrices $\mathbf{Q}_R^{(i)} = \mathbf{P}_R\mathbf{W}_R^{Q(i)}$, key matrices $\mathbf{K}_R^{(i)} = \mathbf{E}^{'}_{residual}\mathbf{W}_R^{K(i)}$, value matrices $\mathbf{V}_R^{(i)} = \mathbf{E}^{'}_{residual}\mathbf{W}_R^{V(i)}$ for residual. Through multi-head attention, we reprogram each time series component.

$$\mathbf{Z}_T^{(i)} = \text{ATTENTION}(\mathbf{Q}_T^{(i)}, \mathbf{K}_T^{(i)}, \mathbf{V}_T^{(i)}) = \text{SOFTMAX}\left(\frac{\mathbf{Q}_T^{(i)}\mathbf{K}_T^{(i)\top}}{\sqrt{d_k}}\right)\mathbf{V}_T^{(i)}, \quad (4)$$

where d_k is the dimension of each head in the multi-head attention module.

3.4 Component-specific prompts & output projection

Prompting techniques effectively encode task-specific knowledge, improving model accuracy and coherence. However, translating time series into natural language is challenging, complicating instruction-following datasets and real-time prompting [13]. Recent advances show that prompts can enrich input context and guide reprogrammed time series patches [6]. To leverage semantic information, we propose a component-specific prefix prompting strategy, incorporating dataset context, statistical features, and task instructions for trend, seasonal, and residual components. For example, the instruction 'forecast the next 96 steps given the previous 512 steps [trend, seasonal, residual]' serves as a template.

After processing these prompts through the backbone LLM, we retain the component embeddings and apply a linear projection. By denormalizing and summing these representations, we derive the final forecasts $\hat{\mathbf{x}}_t^{(i)}, \ldots, \hat{\mathbf{x}}_{t+H-1}^{(i)}$.

4 Experiments

In our experiments, the proposed method outperforms state-of-the-art forecasting approaches across various benchmarks, including long-term, short-term, and few-shot forecasting. For a fair comparison, we follow the configurations outlined in [12] across all baselines, utilizing a unified evaluation pipeline[1].

Baselines. We compare with the SOTA time series models and cite their performance from [3, 16] if applicable. The baselines include Transformer-based models like PatchTST [7], non-transformer-based techniques, i.e., DLinear [15], and four LLM-based approaches: TimeLLM [6], LLM4TS [3], GPT4TS [16], and LLMTime [5]. Aligned with the GPT4TS configuration [16], we utilize only the first 6 layers of the 12-layer GPT-2 base [9] as the backbone for our model and TimeLLM.

Long-term Forecasting. For long-term forecasting, we evaluate on ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity(ECL), and Traffic, which have been widely adopted as benchmarking datasets for long-term forecasting works [12]. The input time series length L is set as 512, and we evaluate across four prediction horizons: $H \in \{96, 192, 336, 720\}$. The evaluation metrics include mean square error (MSE) and mean absolute error (MAE).

[1] https://github.com/thuml/Time-Series-Library

Table 1: Long-term forecasting results. All results are averaged from four different forecasting horizons: $H \in \{96, 192, 336, 720\}$ horizons. Lower values indicate better performance. **bold**: the best, underline: second best.

Methods	Ours		Time-LLM		LLM4TS		GPT4TS		DLinear		PatchTST	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.424	0.448	0.420	0.438	**0.404**	**0.418**	0.428	0.426	0.422	0.437	0.413	0.435
ETTh2	0.378	0.408	0.364	0.403	0.333	0.383	0.355	0.394	0.431	0.446	**0.330**	**0.379**
ETTm1	**0.251**	**0.325**	0.351	0.383	0.343	0.378	0.352	0.383	0.357	0.378	0.351	0.380
ETTm2	**0.214**	**0.289**	0.270	0.328	0.251	0.313	0.267	0.326	0.267	0.333	0.255	0.315
Weather	**0.171**	**0.226**	0.229	0.267	0.223	0.260	0.237	0.271	0.248	0.300	0.225	0.264
ECL	**0.159**	0.262	0.164	0.261	0.159	0.253	0.167	0.263	0.166	0.263	0.161	**0.252**
Traffic	**0.346**	0.265	0.408	0.290	0.401	0.273	0.414	0.294	0.433	0.295	0.390	**0.263**

Table 1 presents the performance of various time series forecasting models on MSE and MAE metrics across different prediction horizons on multiple benchmarks. Our proposed model consistently outperforms existing baselines, demonstrating superior performance on average across most datasets and prediction lengths. This highlights the applicability of multi-level text alignment. Notably, our comparison with TimeLLM—a recent work leveraging text prototype reprogramming to align time series with text tokens—is significant. Specifically, our model achieves substantial improvements on the Weather and ETTm1 datasets, exceeding the best-performing model, LLM4TS, by **23.3%** and **26.8%**, respectively, in terms of MSE. Additionally, it records the lowest error rates across numerous dataset-prediction length configurations. These results suggest that integrating LLMs with multi-level text alignment can significantly enhance the accuracy of long-term time series forecasting. We also conduct a few-shot forecasting. More detailed experimental results for long-term forecasting and few-shot forecasting are in the appendix and full paper [1].

Zero-shot forecasting. Beyond few-shot learning, LLMs also show promise as effective zero-shot learners. In this section, we evaluate the zero-shot learning capabilities of the multi-level text-aligned LLM. Specifically, we assess how well the model performs on one dataset after being optimized on another. Similar to the few-shot learning setup, we use the long-term forecasting protocol and evaluate various cross-domain scenarios utilizing the ETT datasets.

The brief results are presented in Table 2. Our model demonstrates performance that is comparable to or surpasses other baselines. In data-scarce scenarios, our model significantly outperforms other LLM-based models, consistently providing better forecasts. Both our model and TimeLLM [6] outperform traditional baselines, likely due to cross-modality alignment, which more effectively activates LLMs' knowledge transfer and reasoning capabilities for time series tasks. Additionally, our multi-level aligned embeddings better align language cues with

Table 2: Zero-shot learning results on ETT datasets. All results are averaged from four different forecasting horizons: $H \in \{96, 192, 336, 720\}$.

Methods	Ours		Time-LLM		GPT4TS		LLMTime		PatchTST		DLinear	
Datasets	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1 → ETTh2	**0.346**	**0.396**	0.354	0.400	0.406	0.422	0.992	0.708	0.380	0.405	0.493	0.488
ETTh1 → ETTm2	**0.294**	**0.357**	0.310	0.363	0.325	0.363	1.867	0.869	0.314	0.360	0.415	0.452
ETTh2 → ETTm2	**0.276**	**0.345**	0.303	0.356	0.335	0.370	1.867	0.869	0.325	0.365	0.328	0.386
ETTm1 → ETTm2	**0.217**	**0.284**	0.275	0.325	0.313	0.348	1.867	0.869	0.296	0.334	0.335	0.389
ETTm2 → ETTm1	0.562	0.478	**0.501**	**0.453**	0.769	0.567	1.933	0.984	0.568	0.492	0.649	0.537

temporal components of time series, enabling superior zero-shot forecasting performance compared to TimeLLM. More detailed experimental results for zero-shot forecasting are in the appendix and full paper [1].

Multi-level text alignment variants. Table 3 shows that removing component alignment or prefixed prompts hinders knowledge transfer in LLM reprogramming for time series forecasting. Without alignment (**A.1**), performance drops by 75.4% on average. Retaining only seasonal alignment (**B.2**) performs best but still increases MSE by 4.5%, while trend-only alignment causes a 32.2% loss. Using noise anchors (**C.1**) raises MSE by 14.5%, whereas synonymous anchors (**C.2**) yield similar results to the default with under 2% variation. Removing component-specific instruction (**D.1**) and domain features (**D.2**) increases MSE by 7.7% and 1.7%, respectively.

Variant	Long-term		Few-shot	
	96	192	96	192
Default GPT-2 (6)	0.117	0.198	0.360	0.429
A.1 w/o alignment	0.262	0.347	0.571	0.583
B.1 only trend align	0.184	0.283	0.476	0.578
B.2 only seasonal align	0.127	0.212	0.367	0.432
B.3 only residual align	0.171	0.229	0.433	0.506
C.1 noise anchors	0.134	0.214	0.424	0.464
C.2 synonymous anchors	0.119	0.202	0.366	0.434
D.1 w/o instruction	0.125	0.205	0.408	0.461
D.2 w/o domain features	0.118	0.199	0.371	0.440

Table 3: Comparison of different variants for long-term and few-shot forecasting on ETTm1 dataset with horizons 96 and 192.

Multi-level text alignment interpretation. We conduct a case study on ETTm1 using non-overlapping patching, where the patch stride equals the patch length. Figure 3 shows the selected anchors, including “increase, decrease, upward, downward, linear, exponential, drift, stable, volatile, stationary, persistent, and rapid”. The attention map highlights optimized attention scores between trend patches and aligned anchors, serving as textual shapelets for time series tokens.

The attention map illustrates the optimized attention scores between input trend patches and aligned anchors. These matched anchors serve as textual shapelets for the time series tokens. Specifically, subplot **(a)** displays the optimized attention scores for synonymous anchors, which are consistent. The highlighted anchors, “rise”, “increase”, “climb”, “grow”, and “expand”, are associated with upward trend patches. In contrast, subplot **(b)** shows no highlighted anchor when all trend patches are aligned with noise words unrelated to time series trends. This case study demonstrates that aligned anchors effectively summarize the textual shapelets of the input trend patches.

Although multi-level alignment in both seasonal and residual components can provide visual interpretations, visualizing these alignments is challenging. Since the input patches are aligned with text prototypes learned from a large and dense pre-trained word embedding space, more tools are needed to present a better visualization across two optimized layers. Moreover, the trend component is the most interpretable and semantically clear of the three components, in contrast to the noisy residual and the seasonal component, which lacks textual semantics. Our model efficiencies in terms of parameters, memory, and speed are comparable to TimeLLM [6] with only two additional lightweight aligned embedding layers for integrating trend and seasonal components.

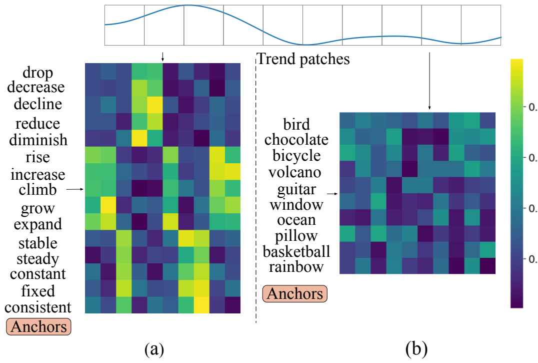


Fig. 3: A showcase of visualization of multi-level alignment interpretation.

5 Conclusion and future work

We propose a multi-level text alignment framework with pre-trained language models for time series forecasting. Our multi-level aligned embeddings enhance the LLM’s interpretability and forecasting performance by aligning time series components with anchors and text prototypes. Our results demonstrate that time series tokens aligned with anchors provide a clearer and more intuitive interpretation of similar time series trends. Future research should focus on optimizing the alignment module for selected anchors and time series tokens, and work toward developing multimodal models capable of joint reasoning across time series, natural language, and other modalities.

6 Acknowledgments

This work was supported in part by the NSF under Grant No. 1943486, No. 2246757, No. 2315612, and 1946231.

References

1. Full paper, `https://github.com/ztb-35/MLTA/blob/main/full.pdf`
2. Cao, D., Jia, F., Arik, S.O., Pfister, T., Zheng, Y., Ye, W., Liu, Y.: Tempo: Prompt-based generative pre-trained transformer for time series forecasting. In: The Twelfth International Conference on Learning Representations (2024)
3. Chang, C., Peng, W.C., Chen, T.F.: Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. arXiv preprint arXiv:2308.08469 (2023)
4. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I., et al.: Stl: A seasonal-trend decomposition. J. off. Stat **6**(1), 3–73 (1990)
5. Gruver, N., Finzi, M., Qiu, S., Wilson, A.G.: Large language models are zero-shot time series forecasters. Advances in Neural Information Processing Systems **36** (2024)
6. 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)
7. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: The Eleventh International Conference on Learning Representations (2023)
8. Pan, Z., Jiang, Y., Garg, S., Schneider, A., Nevmyvaka, Y., Song, D.: Sˆ2 ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In: Forty-first International Conference on Machine Learning (2024)
9. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
10. Sun, C., Li, H., Li, Y., Hong, S.: Test: Text prototype aligned embedding to activate llm's ability for time series. arXiv preprint arXiv:2308.08241 (2023)
11. 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. arXiv preprint arXiv:2302.13971 (2023)
12. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186 (2022)
13. Xue, H., Salim, F.D.: Prompt-based time series forecasting: A new task and dataset. arXiv preprint arXiv:2210.08964 (2022)
14. Xue, H., Salim, F.D.: Promptcast: A new prompt-based learning paradigm for time series forecasting. IEEE Transactions on Knowledge and Data Engineering (2023)
15. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 11121–11128 (2023)
16. 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)