LABO: Fine-Tuning LLMs and Multi-Domain Agents for Time Series Forecasting

Keitaro Sakamoto [1] **Daiki Sato** [2] **Aki Matsumoto** [2] **Issei Sato** [1]

Abstract

LABO introduces a cutting-edge framework for time series forecasting by integrating pre-trained and fine-tuned large language models (LLMs) with multi-domain agent-based systems. Through innovative domain-adaptive fine-tuning techniques and collaborative agent learning, LABO achieves unparalleled accuracy and adaptability for forecasting tasks across diverse industries, including finance, healthcare, energy, and supply chain management. This paper details LABO's methodology, architecture, and practical applications, showcasing its ability to address complex temporal dependencies, domain-specific requirements, and real-world dynamic environments.

1. Introduction

Over the past few years, pre-trained large language mod els (LLMs) such as GPT-4 (Achiam & et al., 2023) and LLaMA(Touvron et al., 2023b;c) not only achieved great success across a diverse range of natural language process ing (NLP) tasks, i.e., generate coherent and contextually relevant text, answer questions, and translate sentences be tween multiple languages, but also exhibited tremendous potential in tackling applications of more complex or struc tured domains, such as code generation, healthcare, finance, and autonomous systems, etc (Singhal et al., 2022; Cui et al., 2024; Li et al., 2023). As time series analysis is be coming increasingly important for strategic planning and op erational efficiency in various real-world applications, e.g., energy load management, traffic forecasting, weather fore casting, health risk analysis, etc (Friedman, 1962; Courty & Li, 1999; B¨ ose et al., 2017; Gao et al., 2020; Li et al., 2022; Liu et al., 2023a; Dimri et al., 2020), a natural question to ask is whether we should train a general purpose founda tion model from scratch, or fine-tune pre-trained LLMs to perform time series forecasting?

Recently, significant efforts have been made to build foun dation models for general-purpose time series analysis (Wu et al., 2023; Garza & Mergenthaler-Canseco, 2023; Rasul et al., 2023). TimesNet (Wu et al., 2023) uses TimesBlock as a task-general backbone to capture multi-periodicity and extract complex intraperiod- and interperiod-variations via transformed 2D tensors. TimeGPT-1 describes a general pre-trained model for time series forecasting (Garza & Mergenthaler-Canseco, 2023). These approaches, however, are hindered by two main challenges. First, time series data can beacquired in various formats, such as univariate or mul tivariate, often in large volumes, and from different domains, like healthcare, finance, traffic, environmental sciences, etc. This escalates the complexity of model training and poses challenges in handling different scenarios. Second, time series data, in practice, often exhibit non-stationary char acteristics, resulting in the underlying statistical properties, such as means, variances, and auto-correlations shifting dur ing collection. This could also result in concept drift, where the statistical properties of target variables change over time. These realities present significant challenges for

large models to be adapted and retrained effectively.

On the other hand, LLMs trained on extensive and diverse text corpora can serve as a foundational knowledge base that can be applied to a variety of downstream tasks with minimal task-specific prompt learning or fine-tuning. Inspired by this, there has been a growing interest in leveraging existing LLMs to facilitate time series analysis. For instance, Tian Zhou & Jin (2023) utilizes a frozen pre-trained language model to attain state-of-the-art or equivalent performance. Jin et al. (2024) develop time-LLM to reprogram the input time series via text prototype representations by incorporating the embeddings of the dataset's text descriptions as context information. In real-world applications, however, dataset description information may not always be available or informative. In addition, the patching operation (i.e., tokenization), which splits a long time series sequence into overlapping segments over instance normalized time series input, may have limited expressibility as it could fail to capture the subtle variations of different components in time

[1]The University of Tokyo [2]Google Japan. Correspondence to: Keitaro Sakamoto <skeitaro@utokyo.ac.jp>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

series.

In this paper, we argue that the semantic space in the form of word token embeddings (based on pre-trained LLMs) can already offer a more distinctive and informative representation space (Ethayarajh, 2019) to help align time series embeddings. Based on this, we develop Semantic Space Informed Prompt with LLM (LABO) for time series forecasting. Specifically, as shown in Figure 1, we first design a tokenization module tailored to semantic space alignment, which explicitly concatenates patches of decomposed time series components (i.e., trend, seasonality, and residual) to create an embedding that effectively encodes the temporal dynamics more expressively. Next, we map the pre-trained word embeddings to obtain semantic anchors and align selected anchors with time series embeddings by maximizing the cosine similarity in the joint space. In this way, LABO can retrieve relevant semantic anchors as prefix-prompts to provide strong indicators (context) for time series embeddings that exhibit different temporal dynamics. Our experiments over several standard benchmark datasets demonstrate that LABO can achieve superior forecasting performance over state-of-the-art baselines. Moreover, our ablation studies and visualizations also verify the necessity of prompt learning in the joint space.

To summarize, our contributions include:

- We design a specialized tokenization module that concatenates patches of decomposed time series components to provide more expressive local contexts and facilitate semantic space informed prompting.
- We leverage semantic anchors derived from pre-trained word token embeddings (semantic space) to align time series embeddings and learn a distinctive and informative joint space. Moreover, aligned semantic anchors are used as prompt indicators (contexts) to enhance the representation of time series.
- Our experiments and analysis on multiple benchmark datasets demonstrate the superiority of LABO over state of the art and the necessity of prompt learning informed by semantic space.

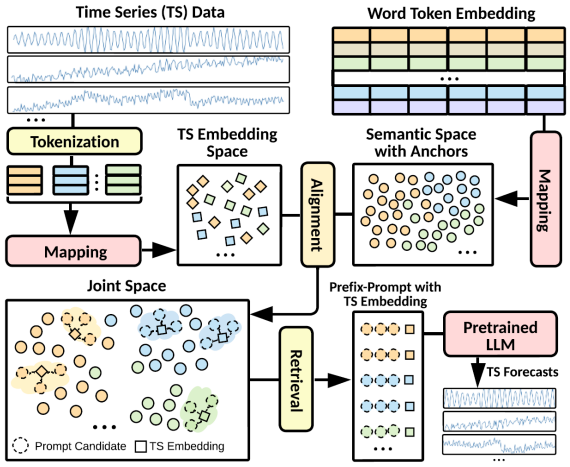

Figure 1. The demonstration of semantic space informed prompting in LABO. The input time series is decomposed and mapped to obtain time series (TS) embedding. Next, the TS embedding is aligned with semantic anchors derived from the pre-trained word token embedding. Finally, top-K similar semantic anchors are retrieved and used as prefix-prompts with TS embedding.

2. Related Work

2.1. Time Series Forecasting

In recent years, a variety of statistical and machine learning methods have been developed for time series analysis, e.g., ARIMA (Anderson & Kendall, 1976), Prophet (Taylor & Letham, 2018), etc. More recently, different types of deep neural networks have been applied for time series analysis. For instance, recurrent neural network (RNN) based models have been developed to capture auto-regressive temporal dynamics (Qin et al., 2017; Li et al., 2017; Lai et al., 2018; Gu et al., 2021). Graph neural networks (GNN) based methods are leveraged to capture variable dependencies among different time series (Cao et al., 2020;Wu et al., 2020; Shang et al., 2021; Pan et al., 2024). Transformer based models leverage the self-attention mechanisms tailored for time series to better capture the temporal dynamics, variable dependencies, or both (Woo et al., 2022; Zhou et al., 2021;Wu et al., 2021; Zhou et al., 2022; Liu et al., 2023b). More recently, MLP-based models (Challu et al., 2023; Zeng et al., 2023) and convolution-based models (Wu et al., 2023) have achieved state-of-the-art performance on par with Transformers, but with much simpler designs. Nevertheless, while these deep forecasters perform well on specific datasets, they lack the flexibility and generalizability to adapt to real-world time series data from different domains.

2.2. Pre-trained Large Model for Time Series Analysis

Recent advancements in natural language processing (NLP) and computer vision (CV) demonstrate that pre-trained models can effectively adapt to a range of downstream tasks through fine-tuning (Bao et al., 2021; He et al., 2022; Brown et al., 2020; Devlin et al., 2018). Inspired by this, several different pre-trained models have been developed for time series based on either supervised (Fawaz et al., 2018) or self-supervised learning (Zhang et al., 2022b; Deldari et al., 2022). During the training stage, models can learn robust representations from a variety of input time series data. Then, these models can be fine-tuned for downstream tasks

of similar domains to further enhance their performance (Tang et al., 2022). With the emergence and success of Large Language Models (LLMs), including T5 (Raffel et al., 2020), GPT-based models (Radford et al., 2018; 2019; Brown et al., 2020; Ouyang et al., 2022), and LLaMA (Touvron et al., 2023a), which have showcase their robust pattern recognition and reasoning abilities over complex sequences of tokens, there is a trend to explore how to effectively transfer knowledge from these powerful pre-trained LLM models to time series domain (Jiang et al., 2024). One line of research focuses on leveraging the pre-trained LLMs as zero-shot learners. For instance, Xue & Salim (2022) and Nate Gruver & Wilson (2023) directly convert time series data to corresponding text sequence inputs and achieve encouraging results for time series forecasting. Another line of research (Tian Zhou & Jin, 2023; Chang et al., 2023) involves tokenizing the input time series data into overlapping patches and strategically leveraging or fine-tuning LLMs for time series analysis. Following this paradigm, TEST (Sun et al., 2023) and Time-LLM (Jin et al., 2024) reprogram time series data with text prototype embedding and incorporate textual prompts for time series analysis. TEMPO (Cao et al., 2023) incorporates the decomposition of time series and retrieval-based prompt design for non-stationary time series data. Different from those methods, we explicitly leverage semantic anchors derived from pre-trained word token embeddings (semantic space) to align time series embeddings and develop a simple yet effective prompt mechanism to inform LLM for forecasting tasks.

3. Methodology

Overview: LABO consists of three key components as shown in Figure 2. Given the input time series, we first tokenize it and obtain the time series (TS) embedding based on time series decomposition and patching. Next, we will align the TS embedding with semantic anchors derived from the pre-trained word token embedding. Finally, top-K similar semantic anchors will be retrieved to serve as prefixprompts for the TS embedding and the concatenated vector will be leveraged as the query for pre-trained LLMs.

In this paper, GPT-2 is used as the backbone. During the training stage, we not only learn the mapping functions of input and output but also fine-tune the positional embedding and layer norm block of GPT-2.

3.1. Problem Statement

We first formalize the time series forecasting problem. Let $X \in R^{N \times T}$ denote the time series data containing N variables and T time steps, where $X_{:,t} \in R^{N \times 1}$ denotes t-th time step across all variables and $X_{i,:} \in R^{1 \times T}$ denotes i-th variable. Given a historical t-step window of time series, we aim to learn a forecasting module $F()$ that will predict the next t' time steps based on the input window. Mathematically, at a starting time step t, the corresponding forecast is given by $\hat{Y} = \hat{X} :, t : t + t' - 1 = F(X :, t - t : t - 1)$.

3.2. Time Series Tokenization

In real-world applications, non-stationary data is prevalent. To tackle this problem, we first apply the reversible instance normalization (Kim et al., 2021) on time series input such that the data has zero mean and unit standard deviation to mitigate the distribution shift in time series. Specifically, given the i-th time series input at time step t, i.e., Xi,t, the transformed value X' i,t can be given by:

$$X_{i,t}^{'} = r_T(X_{i,t} - \frac{E_t(X_{i,t})}{\sqrt{Var(X_{i,t}) + \epsilon_T}} + \beta_T) \quad (1)$$

where $E_t[Xi, t]$ and Var $[X_{i,t}]$ are the instance-specific mean and variance, respectively. r_T and β_T are trainable parameters. Next, we adopt an additive seasonal-trend decomposition method to decompose normalized time series into long-term trend, seasonal, and residual components. The additive seasonal-trend decomposition is given $X_{i,t}^{'} = X_{i,t}^{tre} + X_{i,t}^{sea} + X_{i,t}^{res}$, where tre, sea, res denotes the long-term trend, seasonal, and residual component, respectively. There are several options for additive seasonal-trend decomposition. One option is the classical additive seasonal-trend decomposition that first obtains long-term trend components using moving averages. Then, the seasonal component is estimated by averaging the detrended time series with pre-defined season parameters. Finally, the residual component is obtained by subtracting the estimated trend and seasonal components from the normalized time series. Another option is the Seasonal-Trend decomposition using Loess (STL) (Cleveland et al., 1990). The choice of decomposition method will based on validation results.

Next, we follow (Nie et al., 2023) to encode temporal information and local contexts of input time series by aggregating consecutive time steps into overlapped patched tokens. Take the trend component as an example, the normalized component series, $X_{i,t-t:t-1}^{tre} \in R^{1 \times t}$ is converted to patched token representation $P_{i,t-t:t-1}^{tre} \in R^{NP \times LP}$, in which LP is the patch length, NP is the number of patches and S is the horizontal sliding stride. We apply patching to each variable component over the temporal dimension and then concatenate the tokens of t components into a single meta-token. We then feed the meta-token into a projection layer to get the time series embedding Pi, where D is the embedding size for the pre-trained LLMs.

3.3. Semantic Space Informed Prompting

Prompting has emerged as an effective technique in various applications, enabling LLMs to utilize task-specific information to achieve enhanced reasoning capabilities (Yin et

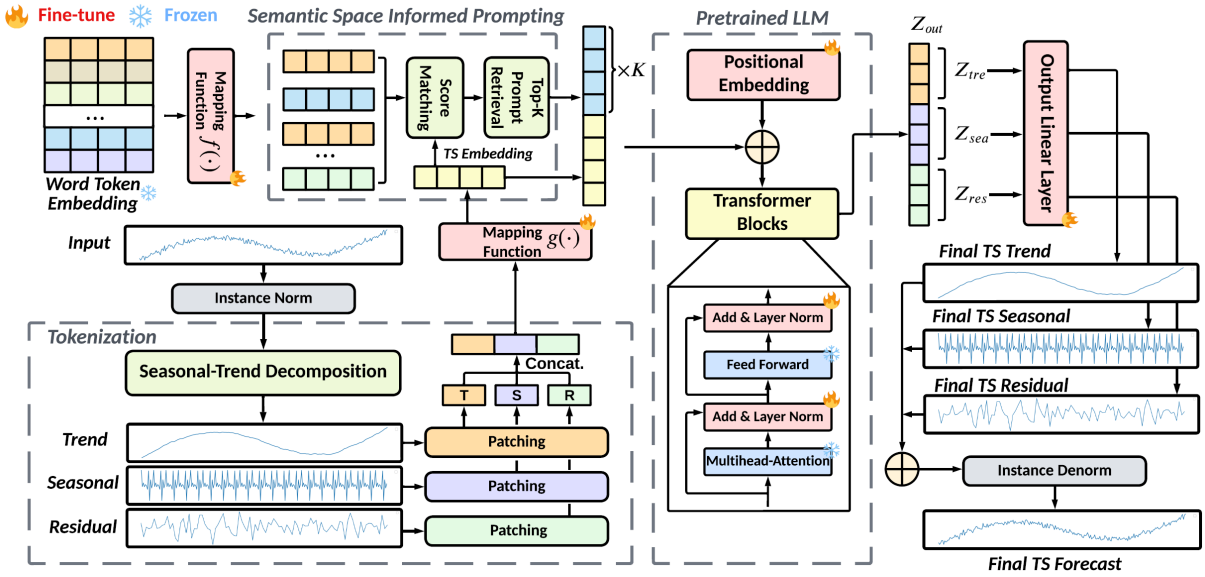

Figure 2. The model architecture of LABO. The input time series is normalized, decomposed, patched individually, and concatenated to represent the context of time series (TS). Semantic space informed prompting performs alignment between the contextual TS embeddings and the semantic anchors extracted from pre-trained word embeddings, and retrieves the most similar K ones as prefix-prompts. The decomposed TS representations from pre-trained LLM are linearly projected and combined as the TS forecast.

al., 2023). Existing works primarily focus on employing template-based and fixed prompts for pre-trained LLMs in time series analysis (Xue & Salim, 2022; Jin et al., 2024). While these methods are intuitive, straightforward, and yield satisfactory results, their rigid prompt contexts are in line with linguistic semantics. However, time series representation inherently lacks human semantics and is more closely tied to sequence patterns in the form of temporal dynamics. Conversely, Lester et al. (2021) demonstrate the effectiveness of soft prompts in enabling LLMs to comprehend inputs more effectively. In the realm of time series analysis with LLMs, recent works (Sun et al., 2023; Cao et al., 2023) start to consider soft prompts as task-specific, randomly initialized, trainable vectors that learn from the supervised loss between LLM's output and the ground truth. However, the semantic space of LLMs, established through the pre-training, is still underexplored and may help yield more distinctive and informative representations for time series data. Based on this intuition, we introduce a prompting mechanism informed by the pre-trained semantic space. Specifically, the pre-trained semantic word token embeddings, represented as E where V is the vocabulary size, are inevitably large and dense. For example, the vocabulary size of GPT-2 (Radford et al., 2019) reaches 50,257 and may raise computational deficiency. Instead of directly using the semantic word token embedding, we derive a small set of semantic anchors E in the hidden space using a generic mapping function $f(\dot{})$ on E, which is denoted as E, where V is the reduced number of semantic anchors and V. To properly retrieve relevant semantic anchors to enhance the time series embedding P, we align the semantic anchors and time series embedding based on a score-matching function r. In this paper, we implement the score-matching function based on cosine similarity

$$\gamma(P_{i,t-t:t-1}, e_m^{'}) = \frac{P_{i,t-t:t-1}\dot{e}_m^{'}}{\|P_{i,t-t:t-1}\|\|\dot{e}_m^{'}\|} \tag{2}$$

We select top-K relevant semantic anchors based on the similarity scores and utilize them as prefix-prompt to enhance the input time series embedding, which will serve as the input for the pre-trained LLMs.

3.4. Optimization Objective

We can obtain the output embedding Zout after the forward path of the prompt enhanced time series embedding through LLMs. We will flatten it and use a linear mapping to project the representation to the forecasting horizon Yout . The overall forecasting should also be the additive combination of the individual component predictions due to the decomposition step. We further split and express Yout into a concatenation form. At every training iteration, the overall training objective is forecasting loss in the form of mean squared error (MSE), and the second term is a scorematching function to align selected semantic anchors with the time series embedding obtained via decomposition and patching. In this way, we could obtain a more informative space to facilitate the underlying forecasting task.

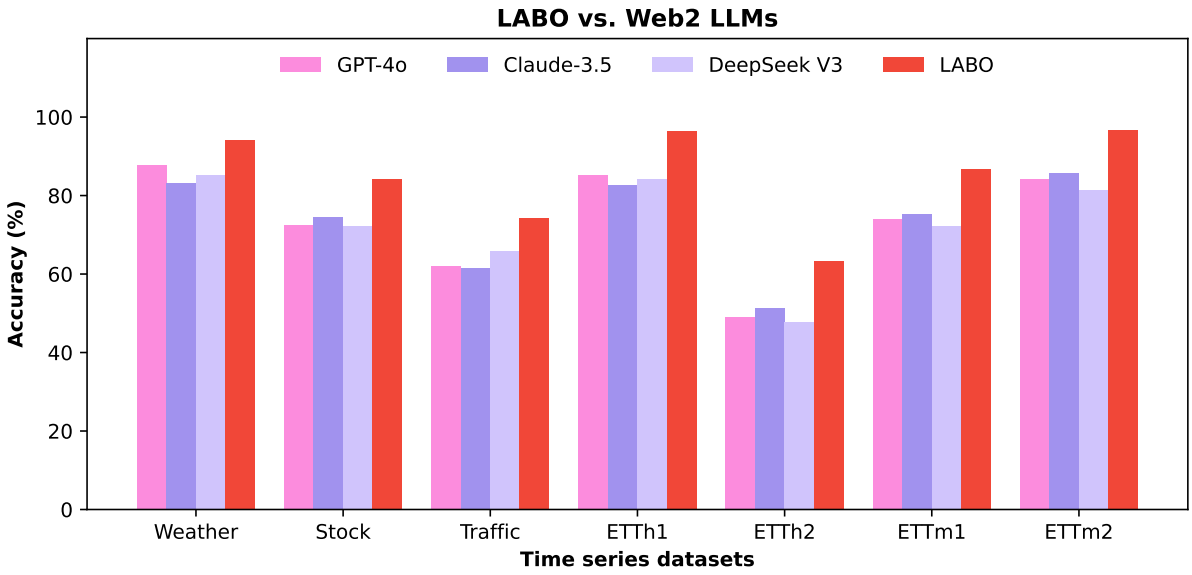

Figure 3. Comparison of the performance of LABO and Web2 LLMs on time series datasets.

3.5. Backbone and Fine-tuning Strategy

In this paper, we employ GPT-2 (Radford et al., 2019) as our pre-trained large language model (LLM) backbone. We choose to keep a significant portion of the parameters frozen, especially those parameters related to the multiheaded attention and the feed-forward networks within the Transformer blocks. This strategy can not only reduce the computational burden but also align with existing literature (Lu et al., 2022; Houlsby et al., 2019; Tian Zhou & Jin, 2023). They suggest that maintaining most of the parameters in their non-trainable state can achieve better outcomes compared to completely retraining LLMs. For GPT-2, we only fine-tune the positional embedding layer and the layernormalization layers.

4. Experiments

In our experiments, we compare the proposed LABO against a variety of baselines on 11 public datasets. We validate the effectiveness of LABO over different time series tasks, including long-term forecasting (Section 4.1), short-term forecasting (Section 4.2), and few-shot forecasting (Section 4.3). We also provide the ablation studies and parameter sensitivity analysis in Section 4.4. Finally, we visualize the prompt enhanced time series embeddings to qualitatively assess the effectiveness of LABO. We follow the experimental configurations (Wu et al., 2023) for all baselines using the unified pipeline.

Baselines. The baselines include a set of Transformerbased methods, i.e., iTransformer (Liu et al., 2023b), PatchTST (Nie et al., 2023), FEDformer (Zhou et al., 2022), Autoformer (Wu et al., 2021), Non-Stationary Transformer (Liu et al., 2022), ETSformer (Woo et al., 2022) and Informer (Zhou et al., 2021). We also select a set of non-transformer based techniques, i.e., DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2023), and LightTS (Zhang et al., 2022a) for comparison. Finally, two approaches based on LLMs, i.e., OFA (Tian Zhou & Jin, 2023) and Time-LLM (Jin et al., 2024).

4.1. Long-term Forecasting

Setup. For long-term forecasting, we evaluate the effectiveness of LABO on Weather, Electricity, Traffic, and four ETT datasets (i.e., ETTh1, ETTh2, ETTm1, and ETTm2), which have been widely adopted as benchmarking datasets for long-term forecasting tasks. Details of these datasets are shown in Appendix A.3, Table 5. The input time series length is 512, and we evaluate the performance on four different horizons 96, 192, 336, 720. The evaluation metrics include the mean square error (MSE) and the mean absolute error (MAE).

Results. We compare the forecasting results of LABO to 6 selected baselines in Table 1. Due to the space limitation, the comparisons with the other 6 baselines are provided in Appendix B and Table 6. We can observe that LLMs based forecasting methods, i.e., Time-LLM and OFA, generally achieve better performance than other baseline methods. This should be attributed to the prevalent expressibility of LLMs and their associated prompt-tuning and fine-tuning strategies, respectively. Moreover, most of the time, LABO outperforms Time-LLM and OFA over 7 different datasets. This is because (1) the unique way LABO tokenized the

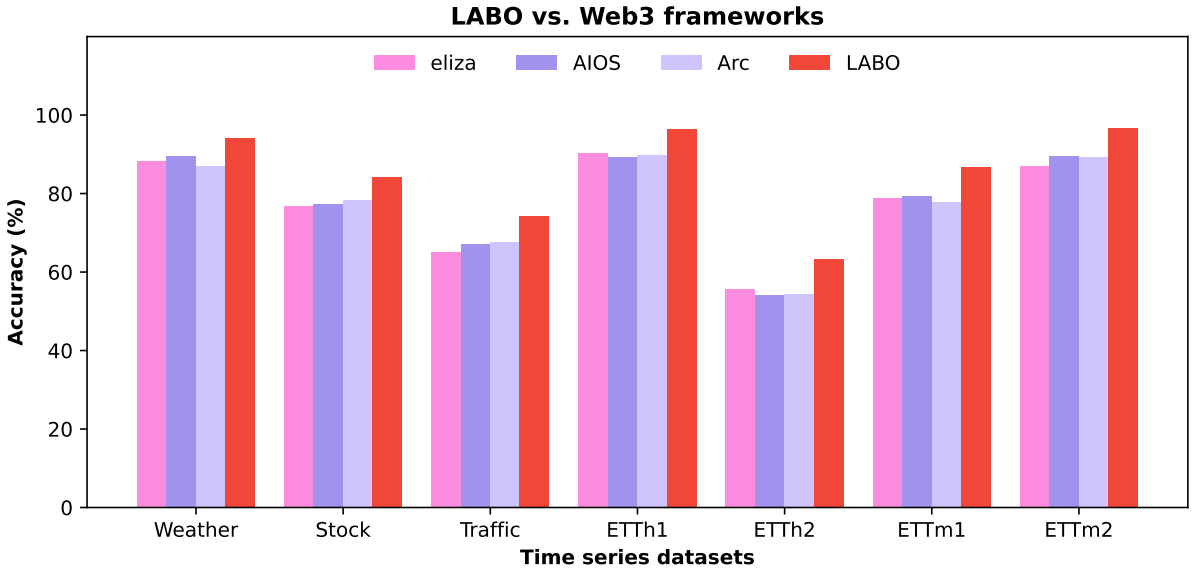

Figure 4. Comparison of the performance of LABO and Web3 agent frameworks on time series datasets.

input time series data can yield better time series representations, and (2) the semantic space informed prompting can help further enhance the time series representation which will be further demonstrated in Section 4.5.

4.2. Short-term Forecasting

Setup. We also evaluate the effectiveness of LABO with the short-term forecasting setting based on the M4 datasets (Makridakis et al., 2018). It contains a collection of marketing data that are sampled at different frequencies. Details of the datasets can be found in Appendix A.3. The prediction horizons are significantly shorter than the longterm forecasting setting and are set between 6 and 48. The input lengths are twice the prediction horizons, similar to the experiment setting in (Jin et al., 2024; Tian Zhou & Jin, 2023). The evaluation metrics for short-term forecasting are symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MASE), and overall weighted average (OWA). The details of these evaluation metrics are provided in Appendix A.4.

Results. Table 2 summarizes the short-term forecasting results and the full experiment results are shown in Appendix Appendix C, Table 7. We observe that LABO outperforms all other baselines by a large margin and is slightly better than PatchTST. This could attribute to the tokenization design as well as the semantic space informed prompting within LABO.

4.3. Few-shot Forecasting

Setup. We follow the experimental settings in Tian Zhou & Jin (2023) to evaluate the performance in the few-shot forecasting setting, which allows us to examine whether the model can generate accurate forecasting with limited training data. We use the first 5% and 10% of the training data in these experiments.

Results. To ensure a fair comparison in the long-term forecasting setting, we summarize the few-shot learning experiment results under 10% and 5% training data in Table 3 and Table 4, respectively. We also report the full experiment results in Table 8 and Table 9 of Appendix D, respectively. When trained with only 10% of the data, LABO typically ranks as either the best or the second-best compared to other baseline models across different datasets. Meanwhile, we also observe that LLMs based methods, LABO, Time-LLM, and OFA significantly outperform other baseline methods. This is because other baseline methods are trained from scratch and they only have limited training data in this case. On the other hand, LLMs based methods can adapt/align the pre-trained knowledge with the time series embedding to enhance its representation. Even with only 5% of training data, LABO still exhibits, if not superior, comparable performance to time-LLM and OFA.

4.4. Ablation Studies and Parameter Sensitivity

We conduct ablation studies on the ETTh2 and ETTm2 datasets to evaluate the parameter sensitivity for LABO. Figure 3 (1) and (4) presents the experiment results when the length of the prompt varies on ETTh2 and ETTm2, respec-

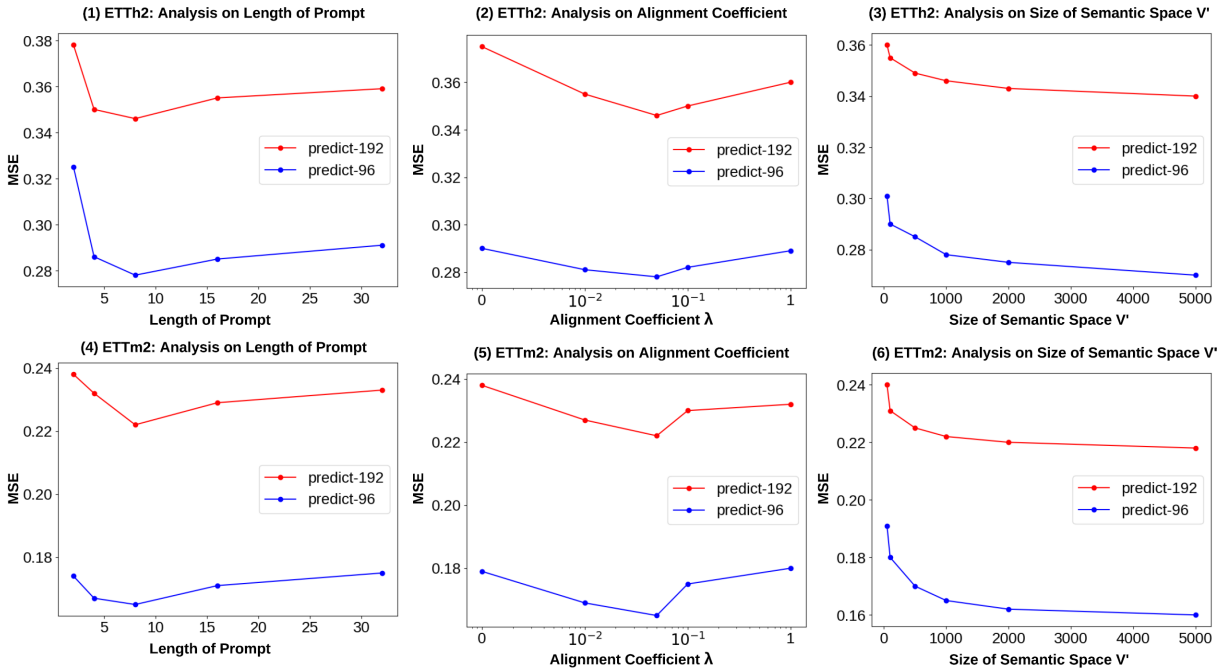

Figure 5. Parameter sensitivity analysis in predicting 96 and 192 steps: (1) and (4) show the effect of prompt length on ETTh2 and ETTm2 datasets; (2) and (5) show the effect of alignment coefficient λ on ETTh2 and ETTm2 datasets; (3) and (6) show the effect of semantic space size V' on ETTh2 and ETTm2 datasets.

tively. Within a limited range, i.e. 2 to 8, an increase in the prompt length tends to improve the forecasting performance. However, excessive prompt length, such as lengths of 16 or 32, results in a significant decline in the forecasting accuracy. A similar pattern can be observed in the hyperparameter analysis of the &, which controls the strength of alignment. As shown in Figure 3 (2) and (5), when & varies from 0 to 0.05, slightly larger & is beneficial for representation learning within the joint space, showing better forecasting results. On the other hand, larger & tends to lead to indistinguishable time series representation and the forecasting performance will thus decrease. Finally, Figure 3 (3) and (6) show the effects of choosing different numbers of semantic anchors. Generally, an increased number of semantic anchors improves the forecasting results. We conjecture that the small number hinders the learning of highly representative semantic anchors in the joint space and thus will generate less informed prompts for time series embedding. We visualize the prompted time series embeddings with the different number of semantic anchors in Appendix E, Figure 6. We notice that a smaller quantity of semantic anchors leads to a less dispersed distribution in the joint space, indicating that the generated prompts could be less informative for time series embedding. We also perform ablation studies by incrementally adding the "alignment & prompting" and "decomposition" modules. In Appendix E Table 10, we observe the forecasting performance increases when we sequentially activate the prompting & alignment component and the decomposition component, which implies the importance of these modules in LABO.

4.5. Qualitative Analysis

In this section, we perform a qualitative analysis of how semantic space informed prompting can facilitate time series representation. Figure 4 shows the visualization of learned semantic anchor embeddings, time series embeddings, and the prompted time series embeddings. The semantic anchor embeddings from the pre-trained language model show distinct clusters, suggesting a robust and differentiated embedding space. In contrast, the raw time series embeddings reveal a more spread-out and less clustered pattern, suggesting that before the alignment, the time series representation is comparatively less informative. After the alignment, the prompted time series embeddings show a clear clustered pattern, suggesting that by aligning with the semantic anchors, time series representation becomes more distinguishable in the joint space.

We also provide the visualizations of prompted time series embeddings under different hyperparameters (when & varies). Within a smaller range, the increase of & appears to enhance the separation of time series embeddings, indicating a more distinct and informative representation. However, as & becomes excessively large, we observe a significant

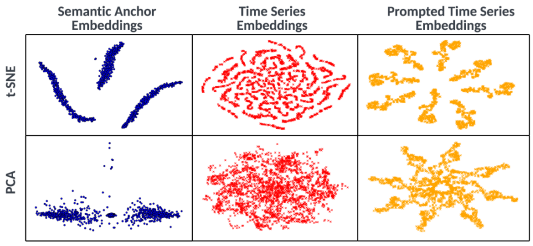

Figure 6. The t-SNE and PCA plots of embeddings space: blue: semantic anchor embeddings; red: time series embeddings; orange: prefix-prompted time series embeddings

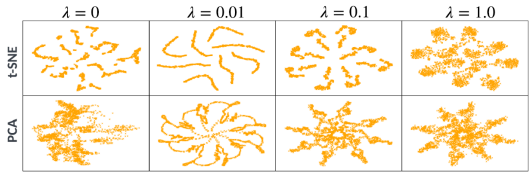

Figure 7. The t-SNE and PCA plots prefix-prompted time series embeddings with different value

decline in the clustering quality of the prompted time series embeddings, which suggests that beyond a threshold, a higher & value leads to less informative embeddings.

5. Conclusion

In this paper, we present LABO, a novel framework for time series forecasting utilizing pre-trained language models. LABO introduces a time series tokenization module that provides expressive local contexts by the concatenation of decomposed time series patches. It creates informative joint space by aligning time series contexts with semantics anchors derived from the pre-trained word token embeddings. The selected aligned semantic anchors are retrieved as prompt indicators to enhance the time series representation and facilitate underlying forecasting tasks. Our thorough empirical studies justified the effectiveness of LABO.

References

Achiam, O. J. and et al., S. A. Gpt-4 technical report. 2023. URL https://api.semanticscholar. org/CorpusID:257532815.

Anderson, O. D. and Kendall, M. G. Time-series. 2nd edn. The Statistician, 25:308, 1976. URL https://api.semanticscholar.org/CorpusID: 134001785.

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254, 2021.

B¨ose, J.-H., Flunkert, V., Gasthaus, J., Januschowski, T., Lange, D., Salinas, D., Schelter, S., Seeger, M., andWang, Y. Probabilistic demand forecasting at scale. Proceedings of the VLDB Endowment, 10(12):1694-1705, 2017.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. Advances in neural information processing systems, 33: 1877-1901, 2020.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., et al. Spectral temporal graph neural network for multivariate time-series forecasting. Advances in neural information processing systems, 33:17766-17778, 2020.

Cao, D., Jia, F., Arik, S. O., Pfister, T., Zheng, Y., Ye, W., and Liu, Y. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. arXiv preprint arXiv:2310.04948, 2023.

Challu, C., Olivares, K. G., Oreshkin, B. N., Ramirez, F. G., Canseco, M. M., and Dubrawski, A. Nhits: Neural hierarchical interpolation for time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pp. 6989-6997, 2023.

Chang, C., Peng, W.-C., and Chen, T.-F. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. arXiv preprint arXiv:2308.08469, 2023.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. Stl: A seasonal-trend decomposition. J. Off. Stat, 6(1):3-73, 1990.

Courty, P. and Li, H. Timing of seasonal sales. The Journal of Business, 72(4):545-572, 1999.

Cui, C., Ma, Y., Cao, X., Ye, W., Zhou, Y., Liang, K., Chen, J., Lu, J., Yang, Z., Liao, K.-D., et al. A survey on multimodal large language models for autonomous driving. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 958-979, 2024.

Deldari, S., Xue, H., Saeed, A., He, J., Smith, D. V., and Salim, F. D. Beyond just vision: A review on selfsupervised representation learning on multimodal and temporal data. arXiv preprint arXiv:2206.02353, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Dimri, T., Ahmad, S., and Sharif, M. Time series analysis of climate variables using seasonal arima approach. Journal of Earth System Science, 129:1-16, 2020.

Ethayarajh, K. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. arXiv preprint arXiv:1909.00512, 2019.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Transfer learning for time series classification. In 2018 IEEE international conference on big data (Big Data), pp. 1367-1376. IEEE, 2018.

Friedman, M. The interpolation of time series by related series. Journal of the American Statistical Association, 57(300):729-757, 1962.

Gao, J., Song, X., Wen, Q., Wang, P., Sun, L., and Xu, H. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. arXiv preprint arXiv:2002.09545, 2020.

Garza, A. and Mergenthaler-Canseco, M. Timegpt-1. arXiv preprint arXiv:2310.03589, 2023.

Gu, A., Goel, K., and R´e, C. Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396, 2021.

He, K., Chen, X., Xie, S., Li, Y., Doll´ar, P., and Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 16000-16009, 2022.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In International Conference on Machine Learning, pp. 2790-2799. PMLR, 2019.

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. arXiv preprint arXiv:2402.03182, 2024.

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 International Conference on Learning Representations, 2024.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In International Conference on Learning Representations, 2021.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long- and short-term temporal patterns with deep neural networks. In The 41st international ACM SIGIR conference on research & development in information retrieval, pp. 95-104, 2018.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691, 2021.

Li, N., Arnold, D. M., Down, D. G., Barty, R., Blake, J., Chiang, F., Courtney, T., Waito, M., Trifunov, R., and Heddle, N. M. From demand forecasting to inventory ordering decisions for red blood cells through integrating machine learning, statistical modeling, and inventory optimization. Transfusion, 62(1):87-99, 2022.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926, 2017.

Li, Y., Wang, S., Ding, H., and Chen, H. Large language models in finance: A survey. In Proceedings of the Fourth ACM International Conference on AI in Finance, pp. 374-382, 2023.

Liu, H., Ma, Z., Yang, L., Zhou, T., Xia, R., Wang, Y., Wen, Q., and Sun, L. Sadi: A self-adaptive decomposed interpretable framework for electric load forecasting under extreme events. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-5. IEEE, 2023a.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. Advances in Neural Information Processing Systems, 35:9881-9893, 2022.