From Values to Tokens: An LLM-Driven Framework for Context-Aware Time Series Forecasting via Symbolic Discretization

Xiaoyu Tao[1], **Shilong Zhang**[1], **Mingyue Cheng**[1*], **Daoyu Wang**[1], **Tingyue Pan**[1], **Bokai Pan**[1] and **Changqing Zhang**[2], **Shijin Wang**[3]
[1]State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
[2]College of Intelligence and Computing, Tianjin University
[3]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research
{txytiny, zhangshilong, wdy030428, pty12345, bk2585934928}@mail.ustc.edu.cn,
mycheng@ustc.edu.cn, zhangchangqing@tju.edu.cn, sjwang3@iflytek.com

Abstract

Time series forecasting plays a vital role in supporting decision-making across a wide range of critical applications, including energy, healthcare, and finance. Despite recent advances, forecasting accuracy remains limited due to the challenge of integrating historical numerical sequences with contextual features, which often comprise unstructured textual data. To address this challenge, we propose TokenCast, a large language model (LLM) driven framework that leverages language-based symbolic representations as a unified intermediary for context-aware time series forecasting. Specifically, TokenCast employs a discrete tokenizer to transform continuous numerical sequences into temporal tokens, enabling structural alignment with language-based inputs. To effectively bridge the semantic gap between modalities, both temporal and contextual tokens are embedded into a shared representation space via a pre-trained LLM, further optimized with generative objectives. Building upon this unified semantic space, the aligned LLM is subsequently fine-tuned in a supervised manner to predict future temporal tokens, which are then decoded back into the original numerical space. Extensive experiments on real-world datasets demonstrate the effectiveness of our framework and highlight its potential as a generative framework for context-aware time series forecasting. The code is available at https://github.com/Xiaoyu-Tao/TokenCast.

1 Introduction

Time series forecasting (TSF) is critical for decision-making in domains such as energy [Cheng and others, 2025a], healthcare [Qiu *et al.*, 2024], and finance [Feng *et al.*, 2019]. In practice, forecasting requires not only modeling temporal dependencies, but also understanding how they interact with external contextual factors—such as static attributes or dynamic events [Liu *et al.*, 2024]. Fundamentally, TSF can be viewed as learning a mapping from past values and contextual features to future outcomes [Jiang *et al.*, 2025].

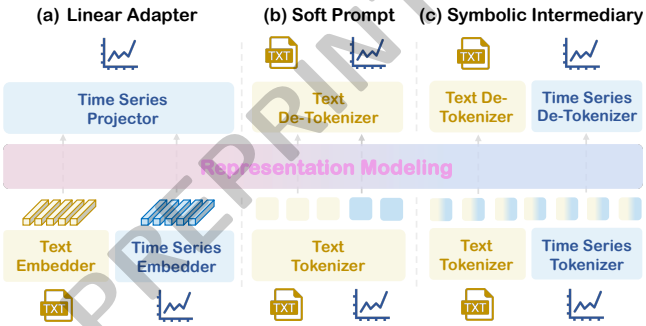

Figure 1: Methods for representation modeling of numerical sequences and contextual features: (a) linear adapter, (b) soft prompt, and (c) symbolic intermediary.

To learn this mapping, researchers have proposed a comprehensive range of methods, ranging from classical statistical models to modern data-driven approaches. Traditional methods, such as ARIMA [Hyndman and Khandakar, 2008] and state-space models [Winters, 1960], rely on strong assumptions about data generation and often incorporate domain-specific priors. In contrast, recent data-driven approaches such as deep learning models aim to learn patterns directly from data without handcrafted assumptions. Architectures based on RNNs [Lai *et al.*, 2018], CNNs [Cheng *et al.*, 2025b], Transformers [Zhou *et al.*, 2022], and MLPs [Zeng *et al.*, 2023] have been widely adopted, each capturing different aspects of temporal dependencies. However, most of these models assume homogeneous numerical inputs and struggle to effectively incorporate complex contextual features, particularly those with heterogeneous modalities.

Beyond capturing temporal dependencies, there is an increasing emphasis in recent research on incorporating contextual features to enhance forecasting performance [Williams *et al.*, 2025]. These features typically fall into two categories: dynamic exogenous variables (e.g., weather conditions, event indicators) and static attributes (e.g., product types, patient demographics, market segments). When contextual features

*Corresponding Author.

share the same numerical modality as the target series, they can be directly modeled as additional input channels. However, many particularly high-value contextual features, such as clinical notes, policy texts, or user logs, are expressed in unstructured textual form. This heterogeneity poses significant challenges for aligning and integrating information across modalities.

To address these challenges, some studies have explored shallow fusion strategies to incorporate contextual features. Models such as DeepAR [Salinas *et al.*, 2020] and Temporal Fusion Transformer (TFT) [Lim *et al.*, 2021] typically concatenate external variables with time series or introduce gating mechanisms. While offering basic integration, these methods often rely on weak alignment and struggle to capture deep semantic interactions across modalities. More recently, LLMs have been introduced into time series forecasting [Sun *et al.*, 2024]. Methods like Time-LLM [Jin *et al.*, 2024] inject time series features into LLMs using linear adapters (Figure 1 (a)) or soft prompts (Figure 1 (b)). Although promising, these approaches fall short in resolving the structural discrepancies between numerical sequences and contextual features. Moreover, they fail to fully leverage the generative capabilities of LLMs, which are pretrained on large-scale corpora. This observation raises a fundamental question: *Can time series be effectively modeled in a discrete token space to unlock the potential of LLMs*?

Motivated by this question, and as illustrated in Figure 1(c), we explore a more expressive paradigm that remains insufficiently explored in prior TSF literature, which formulates time series forecasting as a multimodal discrete context understanding and generation problem powered by pre-trained LLMs. The key idea is to transform continuous numerical sequences into discrete tokens and embed them into the same semantic space as contextual language inputs. This formulation enables the full use of LLMs' capabilities in semantic understanding, contextual reasoning, and autoregressive generation. However, this paradigm introduces several non-trivial challenges. First, discretizing dynamic time series is more difficult than compressing static data, as it requires preserving temporal dependencies while reducing granularity. Second, even with symbolic representations, semantic misalignment between temporal tokens and contextual features may hinder effective fusion. Finally, it remains unclear whether time series forecasting can be effectively addressed through autoregressive generation over discrete tokens.

Based on the above analysis, we propose TokenCast, an LLM-driven framework for context-aware time series forecasting via symbolic discretization. TokenCast begins with a time series tokenizer that converts continuous sequences into discrete tokens, mitigating structural discrepancies across data modalities. To bridge the semantic gap, temporal and contextual tokens are jointly embedded into a shared representation space using a pre-trained LLM, optimized via a generative objective while keeping the backbone frozen and tuning only the embedding layer. Building on this unified semantic space, the aligned LLM is further fine-tuned with supervised forecasting signals to enhance predictive performance. We evaluate TokenCast on diverse real-world datasets enriched with contextual features. Experimental results show that TokenCast achieves strong accuracy and generalization across domains. We also conduct comprehensive ablation and qualitative studies, offering insights into the flexibility of symbolic, LLM-based time series forecasting.

2 Related Work

Time series forecasting (TSF) is a fundamental task across various domains. Traditional approaches typically rely on statistical assumptions such as stationarity and linearity, and often depend on handcrafted assumptions that limit their flexibility [Holt, 2004; Kalekar and others, 2004]. Alternatively, data-driven methods [Chen and Guestrin, 2016], particularly those based on deep learning, have advanced TSF by learning temporal patterns directly from data. RNN-based models [Wang *et al.*, 2019] capture dependencies through recurrence, CNN-based models [Wang *et al.*, 2023] enhance local pattern extraction, and Transformer-based architectures [Shi *et al.*, 2025] are well-suited for modeling long-range interactions. Furthermore, MLP-based approaches [Wang *et al.*, 2024] demonstrate that simple architectures can achieve competitive performance with improved computational efficiency. These models mainly focus on numerical data, with less emphasis on unstructured contextual features.

In addition to modeling temporal dependencies, recent research increasingly emphasizes the integration of contextual features for accurate forecasting [Chang *et al.*, 2025; Hu *et al.*, 2025]. Two major lines of research have emerged in this direction. One line of research focuses on deep learning architectures that model feature interactions [Gasthaus *et al.*, 2019]. Another line of research leverages pre-trained LLMs for multimodal modeling [Cheng *et al.*, 2025a; Cheng and others, 2025b]. Some approaches, such as TEMPO [Cao *et al.*, 2024], utilize linear adapters to project temporal features into the LLM's semantic space. Others, like Promptcast [Xue and Salim, 2023], employ soft prompts to guide the frozen LLM's behavior. However, these promising approaches fail to bridge the structural gap between numerical and textual modalities [Jia *et al.*, 2024].

3 The Proposed TokenCast

In this section, we present the formal problem definition, clarify the key concepts and notations used consistently throughout the paper, and provide an overview of the TokenCast.

3.1 Problem Formulation

We consider a dataset $\mathcal{D} = \{(H_i, T_i, P_i)\}_{i=1}^{N}$ of N multimodal instances, where $H \in \mathbb{R}^{L \times C}$ denotes the historical multivariate time series, T denotes the contextual features, and $P \in \mathbb{R}^{L_P \times C}$ denotes the future time series. The contextual features T are tokenized into language tokens Y by a pre-trained LLM tokenizer, while H is mapped to discrete time series tokens Z_q through a learnable function $f_\theta : H \mapsto Z_q$. These tokens are concatenated as $Z = [Z_q; Y] \in \mathcal{V}^{T'}$. With boundary markers delimiting the generated temporal tokens $\hat{Z}$, a decoding function $g_\phi : \hat{Z} \mapsto \hat{P}$ reconstructs the predicted time series $\hat{P} \in \mathbb{R}^{L_P \times C}$.

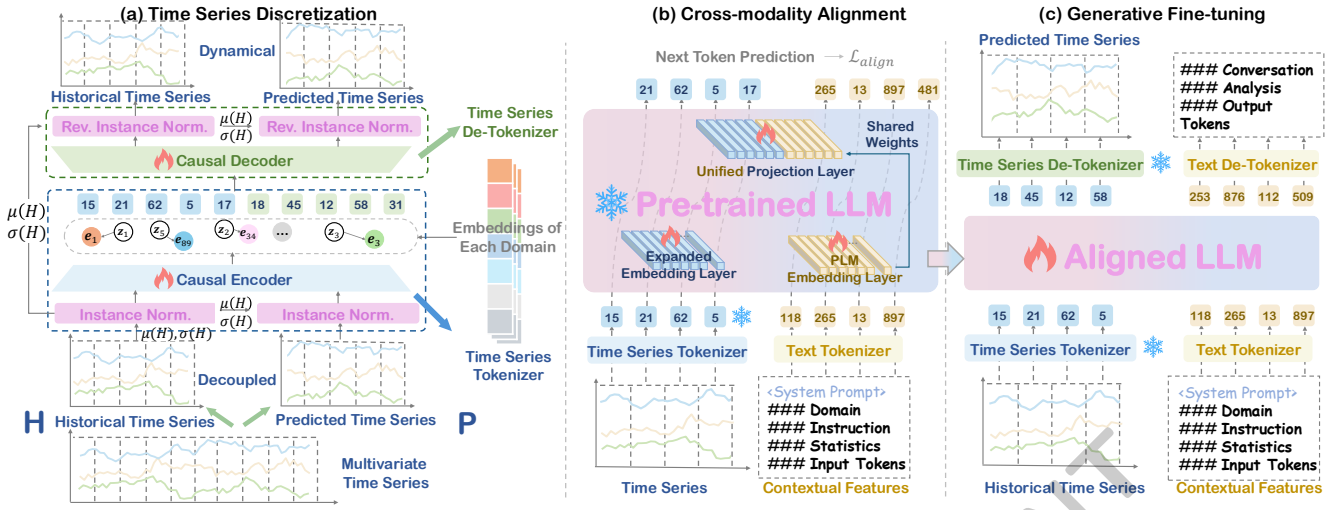

Figure 2: Overview of the framework for context-aware time series forecasting: (a) time series tokenizer to address the structural differences between modalities, (b) cross-modality alignment with a generative objective to bridge the modalities, and (c) generative fine-tuning and context-aware forecasting through time series decoding for horizon prediction.

3.2 Framework Overview

Figure 2 provides an overview of the TokenCast, which consists of three main stages. The process begins with the discretization, which transforms continuous sequences into a sequence of discrete tokens via a decoupled and dynamical vector quantization tokenizer. Subsequently, both the temporal and contextual tokens are then jointly processed by a pre-trained LLM, which performs cross-modality alignment under generative objectives. Following this alignment, the aligned LLM is adapted to the downstream task via generative fine-tuning. The predicted tokens are decoded to raw time series using a frozen de-tokenizer. The following sections elaborate on the principal stages of the TokenCast.

3.3 Time Series Discretization

Time Series Tokenizer. To fully harness the generative and reasoning capabilities of language models, symbolic representation naturally arises as an effective intermediary. Accordingly, we employ time series discretization as a simple yet powerful approach to establish this bridge. It is worth noting that existing approaches, such as Symbolic Aggregate Approximation (SAX) [Lin *et al.*, 2007], have achieved progress in time series discretization but often suffer from significant information loss due to dimensionality reduction. In contrast, reconstruction-based methods map subsequences to discrete codes from a predefined codebook and achieve more precise representations through reconstruction optimization. While preserving the original information is advantageous, previous reconstruction-based methods typically encode the entire sequence, overlooking the statistical properties of time series. Reversible Instance Normalization (RevIN) [Kim *et al.*, 2021] is widely used in forecasting, but its reliance on cached normalization statistics can lead to future information leakage when applied over prediction horizons. To mitigate this issue, we introduce a decoupled and dynamic tokenizer.

As illustrated in Figure 2 (a), similar to the forecasting phase, we divide the multivariate time series into a historical time series $H \in \mathbb{R}^{L_H \times C}$ and a predicted time series $P \in \mathbb{R}^{L_P \times C}$, which can be formally represented as $X = [H; P] \in \mathbb{R}^{L \times C}$. The process begins with a reversible instance normalization (RIN) layer. We compute the mean $\mu(H)$ and standard deviation $\sigma(H)$ solely from the historical time series H, and apply them to normalize the time series X, thereby preventing future information leakage. These statistics are retained for inverse transformation during decoding. Instead of employing separate encoders, we adopt a shared encoder, which facilitates the joint modeling of both local and global information. The normalized time series is then passed through a causal encoder f_{enc}, yielding a sequence of continuous latent representations $Z = f_{\text{enc}}(X) \in \mathbb{R}^{T \times d}$, where T is the number of latent vectors and d is the feature dimension. To discretize the latent representations, we apply a vector quantization layer. For domain i, a learnable codebook $C_i = \{e_{i,k}\}_{k=1}^{K} \subset \mathbb{R}^d$ is maintained, containing K embedding vectors. Each latent vector $z_t \in \mathbb{R}^d$ is mapped to its nearest neighbor in the codebook as $z_t^q = e_{i,k^*}$, where $k^* = \arg\min_k \|z_t - e_{i,k}\|_2^2$. The output of this layer is a quantized sequence $Z_q = (z_1^q, \ldots, z_T^q)$, and the corresponding sequence of indices $\{k^*\}$ serves as the discrete tokens for downstream modeling. These tokens are subsequently decoded by a shared causal decoder f_{dec}, rather than by separate decoders, which ensures consistent reconstruction and enables the predicted part to dynamically exploit richer historical features. Then, the final reconstruction $\hat{X}$ is obtained by applying the inverse RIN operation using the stored statistics $\mu(H)$ and $\sigma(H)$, i.e., $\hat{X} = f_{\text{denorm}}(f_{\text{dec}}(Z_q))$.

Training Objective. The tokenizer is optimized by minimizing the objective function defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \beta\left(\mathcal{L}_{\text{commit}} + \mathcal{L}_{\text{codebook}}\right) + \gamma\mathcal{L}_{\text{diversity}}, \quad (1)$$

where $\mathcal{L}_{\text{recon}} = \|\hat{X} - X\|_2^2$ is the reconstruction loss that optimizes both the encoder and decoder. Due to the non-

differentiability of the $\arg\min$ operation in quantization, we employ the straight-through estimator (STE) during backpropagation. To train the vector quantizer, we include: $\mathcal{L}_{\text{codebook}} = \|\text{sg}[Z] - Z_q\|_2^2$, $\mathcal{L}_{\text{commit}} = \|Z - \text{sg}[Z_q]\|_2^2$, where $\text{sg}[\cdot]$ denotes the stop-gradient operator, which prevents gradients from flowing into its argument during backpropagation. To promote diverse usage of codebook entries, we further add a diversity loss $\mathcal{L}_{\text{diversity}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{d_i+\epsilon}$, where $d_i = \min_{j\neq i}\|e_i - e_j\|_2$ denotes the nearest-neighbor distance between codebook embeddings. This penalty discourages vectors from clustering too closely and encourages more uniform utilization of the codebook.

3.4 Pre-trained LLM Backbone Formulation

Following the discretization of time series into discrete tokens, the next challenge is to model the complex dependencies embedded in these sequences. While architectures like TCNs or Transformers can be trained from scratch, we argue that a pre-trained LLM serves as a more effective backbone. This is supported by two observations: (1) a pre-trained LLM possesses strong semantic understanding and contextual reasoning capabilities acquired from large-scale corpora, and (2) the structure of discrete time series tokens closely resembles that of language tokens [Zhao *et al.*, 2023]. By casting forecasting as a generative task, we directly leverage the LLM's autoregressive generation ability. To guide LLM reasoning and incorporate contextual features, we employ a structured prompt template, as shown in Figure 2 (b). This prompt template consists of four essential components: domain knowledge, task instructions, statistical properties, and discrete time series tokens. This design ensures token-level consistency with language tokens and introduces task-specific descriptions alongside statistical attributes, enabling the LLM to perform instruction-driven generation.

3.5 Cross-Modality Alignment of Time Series and Contextual Features

While discretization aligns time series structurally with language tokens, a semantic gap remains between time series and contextual features. Existing methods often introduce projection modules (e.g., MLPs) to map time series into the LLM's latent space for fusion [Jia *et al.*, 2024]. Although effective in downstream tasks, these strategies rely on external transformation modules for alignment, which bypass the language model's native vocabulary modeling mechanism. To this end, we implement a more explicit vocabulary-level alignment strategy. As illustrated in Figure 2 (b), we construct a unified vocabulary by directly appending K temporal tokens and S task-specific special tokens to the original vocabulary V_{orig} of the pre-trained LLM, forming an extended vocabulary V. Correspondingly, a shared embedding matrix $E \in \mathbb{R}^{|V|\times d}$ is used to encode all tokens, regardless of their modality origin. This unified embedding mechanism enables seamless fusion of time series and contextual features while maintaining alignment with the pre-trained model. To ensure distributional alignment with pre-trained embeddings for fine-tuning, the embedding of the newly introduced time series tokens is initialized by sampling from a multivariate gaussian distribution defined by the mean μ and covariance Σ of the original word embeddings. Then, temporal tokens Z_q and contextual tokens Y are concatenated at the token level and jointly transformed into embeddings via the shared embedding layer: $E([Z_q, Y]) = [E(z_1), \ldots, E(z_n), E(y_1), \ldots, E(y_m)]$, where E denotes the unified embedding matrix. This unified embedding process enables the LLM to reason over concatenated sequences without requiring architectural modification.

To optimize cross-modality token representations within the shared embedding space, we adopt an autoregressive training objective. Specifically, we freeze all parameters of the pre-trained LLM and update only the shared embedding matrix E, which is responsible for encoding both temporal and contextual tokens. Given a concatenated token sequence $[Z_q, Y]$, the training objective is formulated as a next-token prediction task over the combined sequence:

$$\mathcal{L}_{\text{align}} = -\sum_{t=1}^{T}\log p(z_t \mid z_1, \ldots, z_{t-1}; E), \tag{2}$$

where $z_t \in V$ denotes the t-th token in the sequence, and $p(\cdot)$ is the conditional probability predicted by the frozen language model given the embedding vectors from E.

3.6 Generative Fine-tuning and Context-aware Time Series Forecasting

We now detail the procedure for adapting the aligned LLM for forecasting tasks. As illustrated in Figure 2 (c), we employ a generative fine-tuning strategy to specialize the model for context-aware time series forecasting. This process consists of two primary stages: (1) structured prompt-based generative fine-tuning; and (2) context-aware time series forecasting with token-based decoding. In the first stage, prompt-based generative fine-tuning is introduced to explicitly transfer the pretrained language modeling capability into the forecasting domain. Instead of relying on external mapping modules, generative fine-tuning directly formulates forecasting as a generation task, where the model is supervised to output both natural language reasoning and sequences of future time series tokens. This paradigm fosters a fast-thinking behavior: by optimizing an autoregressive objective against ground-truth structured responses, the model learns to rapidly recognize patterns, associate contextual features with temporal dynamics, and produce coherent outputs without engaging in deep deliberation. As a result, the aligned LLM acquires the ability to generate fluent and context-aware predictions. In the second stage, the fine-tuned model is utilized for context-aware forecasting and decoding. During inference, the model receives a prompt with historical data and contextual features, and autoregressively generates a complete response. The key component of this generated output is the sequence of discrete tokens, which represents the model's prediction of future time series values. To translate this symbolic representation back into a continuous predicted time series, these tokens are processed by a frozen time series de-tokenizer. We use boundary markers to delimit the temporal tokens.

Model	**TokenCast**		**Time-LLM**		**GPT4TS**		**TimeDART**		**SimMTM**		**Crossformer**		**Autoformer**		**DLinear**	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Economic	**68.911**	**1.701**	81.542	1.760	85.947	1.716	86.029	1.771	90.351	1.672	406.418	4.074	116.745	2.088	122.216	2.070
Health	**2.525**	**0.081**	2.823	0.104	2.565	0.083	2.623	0.088	2.720	0.088	1644.745	2.504	2.617	0.265	28.587	0.455
Web	**497.410**	**1.246**	557.833	1.751	540.492	1.458	773.635	1.369	847.649	1.327	698.316	1.963	722.506	3.303	632.301	1.398
Stock-NY	**0.482**	**0.455**	0.662	0.510	0.638	0.502	0.776	0.606	0.613	0.585	1.111	0.912	0.676	0.573	0.999	0.754
Stock-NA	**1.134**	**0.780**	1.200	0.925	1.272	0.880	1.409	0.883	1.343	0.834	1.913	1.053	1.558	0.914	1.710	0.958
Nature	0.269	0.297	0.258	0.283	0.274	0.299	**0.243**	**0.273**	0.259	0.286	0.735	0.511	0.508	0.481	0.369	0.436
1st Count	**5**	**5**	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Table 1: All reported results are **average** over four horizons and three trials on various context-rich benchmark datasets. Lower values indicate better performance. The best results are highlighted in **bold**, and the second-best are underlined.

Dataset	Domain	Frequency	Length	Variables
Economic	Economic	1 month	728	107
Health	Health	1 day	1,392	948
Web	Web	1 day	792	2,000
Stock-NY	Stock	1 day	1,243	5
Stock-NA	Stock	1 day	1,244	5
Nature	Nature	30 mins	19,934	11

Table 2: Diverse real-world datasets from various domains and with distinct characteristics.

4 Experiments

In this section, we conduct comprehensive experiments to evaluate our TokenCast's performance on diverse real-world datasets enriched with contextual features for time series forecasting. Additionally, we perform extensive ablation studies and exploration analysis to demonstrate the effectiveness of its individual components.

4.1 Experimental Settings

Datasets. As shown in Table 2, we evaluate our framework on six real-world datasets from diverse domains enriched with contextual features: **Economic** [McCracken and Ng, 2016], **Health** [Panagopoulos *et al.*, 2021], **Web** [Gasthaus *et al.*, 2019], two subsets of **Stock** data [Feng *et al.*, 2019] and **Nature** [Poyatos *et al.*, 2020]. These datasets, spanning various temporal patterns and contextual dependencies, serve as a comprehensive benchmark for context-aware forecasting. Data preparation involves imputing missing values and applying z-score normalization to all datasets, ensuring stable convergence and comparability.

Baselines. We compare our proposed framework against eight strong baselines, grouped into four representative categories for comprehensive evaluation. For LLM-based models, we include Time-LLM [Jin *et al.*, 2024] and GPT4TS [Zhou *et al.*, 2023], which adapt pre-trained LLMs for time series forecasting using modality-aware prompting and reprogramming. In the self-supervised frameworks category, we evaluate TimeDART [Wang *et al.*, 2025] and SimMTM [Dong *et al.*, 2023]. Additionally, we include Transformer-based methods like Autoformer [Wu *et al.*, 2021] and Crossformer [Zhang and Yan, 2023]. Finally, we consider the MLP-based method DLinear [Zeng *et al.*, 2023]. Further details are provided in the Appendix.

Implementation Details. For each baseline, we search over multiple input lengths and report the best performance to avoid underestimating its capability. The historical length is set to $L = 96$ for the Nature dataset and $L = 36$ for the other five datasets, based on the data volume and temporal resolution. The forecasting horizons are set to $\{24, 48, 96, 192\}$ for Nature and $\{24, 36, 48, 60\}$ for the other dataset. We adopt two widely used evaluation metrics in time series forecasting: mean absolute error (MAE) and mean squared error (MSE). We report average results for the main and ablation studies. For exploratory analysis, we use 96-to-24 on Nature and 36-to-24 on the other datasets. Complete results for the main experiments, ablation studies, and exploratory analysis are included in the Appendix. All experiments are implemented in PyTorch and conducted on a distributed setup with 8 NVIDIA A100 GPUs.

4.2 Forecasting Performance Analysis

Table 1 comprehensively compares forecasting performance across six benchmark datasets. TokenCast demonstrates superior performance in most scenarios, further confirming previous empirical findings [Zhou *et al.*, 2023] that no single model performs best across all settings. Notably, LLM-based baselines like Time-LLM also show competitive results, particularly on context-rich datasets such as Economic and Stock-NY. This further validates the potential of leveraging large language models in time series forecasting. However, these models often lack the structural alignment mechanisms introduced by our framework, limiting their consistent performance. Conventional baselines such as TimeDART perform well on datasets with strong periodicity and weak contextual dependence (e.g., Nature), but their performance drops significantly on complex datasets rich in contextual features (e.g., Economic and Web). This contrast underscores the importance of contextual feature modeling and cross-modal interaction. In summary, our framework delivers state-of-the-art results with high consistency. This is attributed to its core design: discretizing time series into discrete tokens and aligning them with contextual features. This unified token-based paradigm effectively addresses real-world context-aware time series forecasting challenges.

Model Variants	Economic		Health		web		Stock-NY		Stock-NA		Nature	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/ Cross-Modality Alignment, w/o Generative Fine-Tuning	80.418	1.774	2.875	0.084	555.375	1.447	0.556	0.479	1.317	0.813	0.378	0.357
w/o Cross-Modality Alignment, w/ Generative Fine-Tuning	72.292	1.690	2.783	0.079	504.740	1.264	0.515	0.478	1.181	0.804	0.305	0.318
w/ Cross-Modality Alignment, w/ Generative Fine-Tuning	**68.911**	**1.701**	**2.524**	**0.081**	**497.410**	**1.246**	**0.482**	**0.455**	**1.134**	**0.780**	**0.269**	**0.297**

Table 3: Ablation study on the effects of cross-modality alignment and generative fine-tuning across multiple datasets.

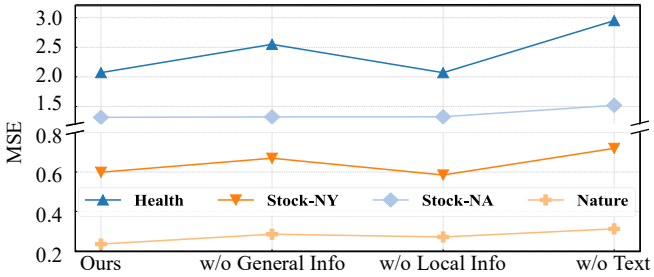

Figure 3: Ablation study on multiple datasets on the contribution of multimodal context in time series forecasting.

Dataset	Economic			Stock-NA			Nature		
Metrics	Recon.	MSE	MAE	Recon.	MSE	MAE	Recon.	MSE	MAE
32	190.371	50.234	1.372	0.244	0.794	0.636	0.134	0.233	0.281
64	**141.852**	**37.699**	**1.293**	0.213	0.690	0.616	0.158	0.241	0.296
128	170.630	39.379	1.251	**0.205**	**0.571**	**0.600**	**0.104**	**0.203**	**0.265**
256	191.937	39.309	1.339	0.209	0.646	0.593	0.114	0.248	0.288

Table 4: Study on the number of tokens in the codebook across multiple datasets. We report predicted reconstructed MSE (Recon.), downstream MSE, and downstream MAE.

4.3 Ablation Studies

Ablation on Alignment and Fine-tuning. We conduct the ablation study on two crucial training steps: the cross-modality alignment and generative fine-tuning. The comprehensive results in Table 3 clearly demonstrate their indispensable contribution to the overall framework performance. The model equipped with the cross-modality alignment stage consistently achieves lower MSE scores across all six datasets. Without this alignment, contextual features risk being misinterpreted by the time series backbone, leading to suboptimal forecasts. This highlights its critical role in effectively integrating contextual information by bridging structural and semantic discrepancies between time series and contextual features, thus facilitating meaningful feature interaction. This alignment thus acts as a foundational step, ensuring the subsequent fine-tuning stage operates on a semantically rich and coherent feature space.

Concurrently, Table 3 vividly illustrates the pivotal contribution of the generative fine-tuning stage. Across all six benchmark datasets, the model employing generative fine-tuning consistently and substantially outperforms its counterpart that omits this crucial step. The performance degradation when omitting this stage is notable across various datasets, underscoring the general applicability and importance of the fine-tuning process. This drop is particularly stark on datasets like Stock-NA, where the complex, non-stationary patterns demand task-specific adaptation. Ultimately, these findings emphasize that generative fine-tuning is essential for adapting the pre-trained LLM's general capabilities to generative time series forecasting.

Ablation on Multimodal Contributions. Fig. 3 analyzes how different types of contextual features affect forecasting performance. Incorporating any contextual features yields substantial improvements, as the variant without contextual input consistently performs worse. We further divide the input into general info, which provides high-level context such as domain knowledge and task instructions, and local info, which offers event-specific details like static statistical attributes. While both types contribute to performance, general info typically brings larger improvements. The strong results of the full model indicate its ability to effectively combine the broad context from general info and the specific cues from local info to enhance prediction accuracy.

4.4 Exploration Analysis

Codebook Size. We conduct a study to assess the impact of codebook size on model performance, as summarized in Table 4. The results highlight the importance of selecting an appropriate codebook size for time series forecasting. Specifically, a size of 128 achieves state-of-the-art results on the Nature and Stock-NA datasets, while a smaller size of 64 excels on the Economic dataset. Interestingly, both smaller (32) and larger (256) codebook sizes fail to produce better results and often lead to significant performance degradation. This suggests that for our framework, simply increasing token granularity is not always beneficial. Instead, a moderate codebook size strikes a balance between reconstruction fidelity and the complexity of the downstream task.

Generative Uncertainty. To validate the uncertainty modeling capabilities of our TokenCast, we conduct experiments on both the Economic and Stock-NY datasets. As shown in Fig. 4, our method produces predictive distributions that

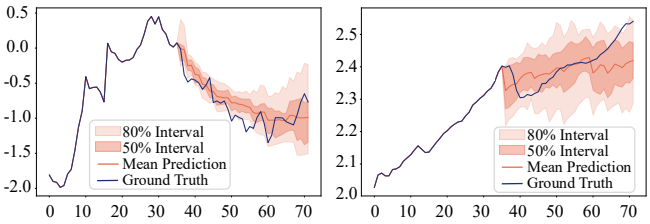

Figure 4: Forecasting with uncertainty on Stock-NY (left) and Economic (right) datasets. The plots compare the ground truth trajectories with the model's mean predictions, along with the 50% and 80% predictive intervals.

Dataset	Economic		Stock-NA		Nature	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
Qwen2.5-0.5B-base	37.164	1.301	**0.668**	**0.605**	**0.180**	**0.246**
Qwen2.5-0.5B-inst.	**36.744**	**1.299**	0.695	0.614	0.187	0.253
Qwen2.5-1.5B-inst.	38.549	1.283	0.722	0.611	0.229	0.270
Qwen3-0.6B-inst.	39.629	1.315	0.936	0.715	0.236	0.281

Table 5: Performance comparison of different backbones and their variants (base/instruct) across varying model scales.

Dataset	Economic		Stock-NA		Nature	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
Mean Initialization	**36.744**	**1.299**	0.695	0.614	**0.187**	**0.253**
Word Initialization	39.680	1.261	**0.667**	**0.602**	0.224	0.264
Random Initialization	36.744	1.299	1.101	0.725	0.189	0.256

Table 6: Study on different initialization methods on the embedding layer. We compare mean initialization, word initialization, and random initialization.

closely track the ground truth, with 50% and 80% prediction intervals capturing the inherent variability in the data. By adjusting the temperature during sampling, we observe that the model can flexibly modulate the spread of the predictive intervals, indicating its potential for controllable uncertainty-aware forecasting. This demonstrates that our model not only provides accurate mean predictions but also yields well-calibrated uncertainty estimates.

LLM Backbone. We evaluate four LLM backbones to identify the optimal architecture for our forecasting framework. As summarized in Table 5, the Qwen2.5-0.5B-base models consistently demonstrate superior performance. Specifically, the base version achieves state-of-the-art results on the Nature and Stock-NA datasets, while the instruct-tuned version excels on the more complex Economic dataset. Interestingly, larger models like Qwen2.5-1.5B-inst. fail to yield further gains and often underperform. This suggests that for our tasks, simply scaling up model size is not beneficial. Instead, the 0.5B models strike a balance between representational capacity and generalization.

Embedding Layer Initialization. We investigate three initialization strategies for our model's embedding layer to identify the most effective approach. As shown in Table 6, mean initialization consistently provides the most consistent performance. Specifically, it achieves the best results on the Nature and Economic datasets. While word initialization is superior on the Stock-NA dataset, its performance is less consistent across other domains. Notably, standard random initialization suffers a significant performance degradation on Stock-NA, highlighting its instability. These findings suggest that initializing embeddings with meaningful prior information provides a better starting point for optimization. Therefore, we adopt mean initialization as the default.

Qualitative Analysis of Tokenization. As shown in Figures 5 and 6, we conduct a comprehensive evaluation of the proposed discretization module on the Nature dataset from three complementary perspectives. The token usage heatmap (Figure 5) indicates that all 64 tokens are actively and consistently utilized across samples, demonstrating effective mitigation of codebook collapse and a strong capacity to capture diverse and heterogeneous temporal structures. The codebook clustering visualization (Figure 6, left) further reveals that tokens organize into coherent and well-separated groups in the latent space, suggesting that the learned discrete vocabulary preserves meaningful structural relationships among different temporal patterns. Moreover, the dynamic reconstruction results (Figure 6, right) highlight the tokenizer's context-adaptive decoding behavior: the same token id (e.g., ID = 18) can generate distinct decoded segments under different contextual conditions, thereby enabling faithful and flexible alignment with the original time series. Overall, these empirical findings confirm that the proposed discretization process learns a diverse, semantically organized, and structurally meaningful vocabulary, while effectively supporting context-aware and adaptive decoding for time series forecasting.

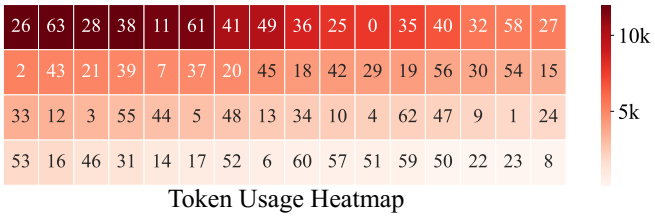

Figure 5: Token usage statistics over the Nature codebook. The heatmap shows the usage frequency of all 64 tokens, with color intensity reflecting how often each token appears.

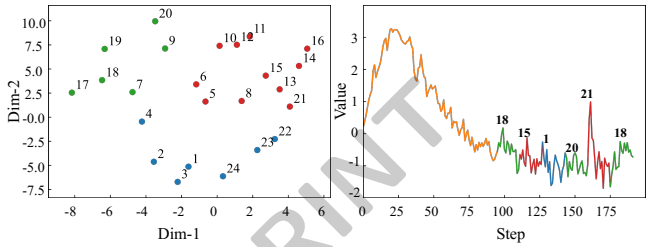

Figure 6: Illustration of codebook clustering in the latent space and dynamic reconstruction.

5 Conclusion

We proposed TokenCast, a context-aware TSF framework based on a pretrained LLM. This approach first converts a continuous time series into discrete tokens. Leveraging a pretrained LLM, it aligns the temporal and contextual tokens through an autoregressive objective, achieving unified modeling of both modalities. The model is then further fine-tuned to generate future token sequences. We evaluate TokenCast on multiple real-world datasets rich in contextual information. Experimental results demonstrate that TokenCast achieves superior accuracy. We also conduct comprehensive ablation experiments and qualitative analysis to validate the framework's adaptability and flexibility for symbolic, LLM-driven TSF. Looking ahead, we believe that leveraging language as a symbolic intermediary will have the potential to advance TSF towards a multimodal and multi-task level.

Acknowledgments

This research was supported by grants from the National Natural Science Foundation of China (No. 62502486, 62376193), the grants of the Provincial Natural Science Foundation of Anhui Province (No. 2408085QF193), Guangdong S&T Programme (No. 2025B0101120004), USTC Research Funds of the Double First-Class Initiative (No. YD2150002501), the Fundamental Research Funds for the Central Universities of China (No. WK2150110032).

References

[Cao *et al.*, 2024] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[Chang *et al.*, 2025] Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, 16(3):1–20, 2025.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[Cheng and others, 2025a] Mingyue Cheng et al. A comprehensive survey of time series forecasting: Concepts, challenges, and future directions, 2025. TechRxiv preprint.

[Cheng and others, 2025b] Mingyue Cheng et al. Instructime: Advancing time series classification with multimodal language modeling. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 792–800, 2025.

[Cheng *et al.*, 2025a] Mingyue Cheng, Xiaoyu Tao, Qi Liu, Hao Zhang, Yiheng Chen, and Defu Lian. Cross-domain pre-training with language models for transferable time series representations. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 175–183, 2025.

[Cheng *et al.*, 2025b] Mingyue Cheng, Jiqian Yang, Tingyue Pan, Qi Liu, Zhi Li, and Shijin Wang. Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 171–180, 2025.

[Dong *et al.*, 2023] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36:29996–30025, 2023.

[Feng *et al.*, 2019] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–30, 2019.

[Gasthaus *et al.*, 2019] Jan Gasthaus, Konstantinos Benidis, Yuyang Wang, Syama Sundar Rangapuram, David Salinas, Valentin Flunkert, and Tim Januschowski. Probabilistic forecasting with spline quantile function rnns. In *The 22nd international conference on artificial intelligence and statistics*, pages 1901–1910. PMLR, 2019.

[Holt, 2004] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

[Hu *et al.*, 2025] Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-alignment: Activating and enhancing llms capabilities in time series. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Hyndman and Khandakar, 2008] Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of statistical software*, 27:1–22, 2008.

[Jia *et al.*, 2024] Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23343–23351, 2024.

[Jiang *et al.*, 2025] Yushan Jiang, Kanghui Ning, Zijie Pan, Xuyang Shen, Jingchao Ni, Wenchao Yu, Anderson Schneider, Haifeng Chen, Yuriy Nevmyvaka, and Dongjin Song. Multi-modal time series analysis: A tutorial and survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6043–6053, 2025.

[Jin *et al.*, 2024] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *International conference on learning representations*, volume 2024, pages 23857–23880, 2024.

[Kalekar and others, 2004] Prajakta S Kalekar et al. Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi school of information Technology*, 4329008(13):1–13, 2004.

[Kim *et al.*, 2021] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

[Lai *et al.*, 2018] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[Lim *et al.*, 2021] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4):1748–1764, 2021.

[Lin *et al.*, 2007] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144, 2007.

[Liu *et al.*, 2024] Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B Aditya Prakash. Lstprompt: Large language models as zero-shot time series forecasters by long-short-term prompting. *arXiv preprint arXiv:2402.16132*, 2024.

[McCracken and Ng, 2016] Michael W McCracken and Serena Ng. Fred-md: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

[Panagopoulos *et al.*, 2021] George Panagopoulos, Giannis Nikolentzos, and Michalis Vazirgiannis. Transfer graph neural networks for pandemic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4838–4845, 2021.

[Poyatos *et al.*, 2020] Rafael Poyatos, Víctor Granda, Víctor Flo, Mark A Adams, Balázs Adorján, David Aguadé, Marcos PM Aidar, Scott Allen, M Susana Alvarado-Barrientos, Kristina J Anderson-Teixeira, et al. Global transpiration data from sap flow measurements: the sapfluxnet database. *Earth System Science Data Discussions*, 2020:1–57, 2020.

[Qiu *et al.*, 2024] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S Jensen, Zhenli Sheng, et al. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proceedings of the VLDB Endowment*, 17(9):2363–2377, 2024.

[Salinas *et al.*, 2020] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

[Shi *et al.*, 2025] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, volume 2025, pages 34635–34667, 2025.

[Sun *et al.*, 2024] Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. In *International Conference on Learning Representations*, volume 2024, pages 37854–37881, 2024.

[Wang *et al.*, 2019] Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International conference on machine learning*, pages 6607–6617. PMLR, 2019.

[Wang *et al.*, 2023] Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

[Wang *et al.*, 2024] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pages 38626–38652, 2024.

[Wang *et al.*, 2025] Daoyu Wang, Mingyue Cheng, Zhiding Liu, and Qi Liu. Timedart: A diffusion autoregressive transformer for self-supervised time series representation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 62627–62651. PMLR, 2025.

[Williams *et al.*, 2025] Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information. In *Forty-second International Conference on Machine Learning*, 2025.

[Winters, 1960] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[Xue and Salim, 2023] Hao Xue and Flora D Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6851–6864, 2023.

[Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

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

[Zhao *et al.*, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.

[Zhou *et al.*, 2023] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.