arXiv:2506.24124v2 [cs.LG] 1 Jul 2025

Teaching Time Series to See and Speak: Forecasting with Aligned Visual and Textual Perspectives

Sixun Dong[1] **Wei Fan**[2] **Teresa Wu**[1] **Yanjie Fu**[1]*
[1]Arizona State University [2]University of Oxford
{sixun.dong, teresa.wu, yanjie.fu}@asu.edu, wei.fan@ox.ac.uk

Abstract

Time series forecasting traditionally relies on unimodal numerical inputs, which often struggle to capture high-level semantic patterns due to their dense and unstructured nature. While recent approaches have explored representing time series as text using large language models (LLMs), these methods remain limited by the discrete nature of token sequences and lack the perceptual intuition humans typically apply, such as interpreting visual patterns. In this paper, we propose a multimodal contrastive learning framework that transforms raw time series into structured visual and textual perspectives. Rather than using natural language or real-world images, we construct both modalities directly from numerical sequences. We then align these views in a shared semantic space via contrastive learning, enabling the model to capture richer and more complementary representations. Furthermore, we introduce a variate selection module that leverages the aligned representations to identify the most informative variables for multivariate forecasting. Extensive experiments on fifteen short-term and six long-term forecasting benchmarks demonstrate that our approach consistently outperforms strong unimodal and cross-modal baselines, highlighting the effectiveness of multimodal alignment in enhancing time series forecasting. Code is available at: https://github.com/Ironieser/TimesCLIP.

1 Introduction

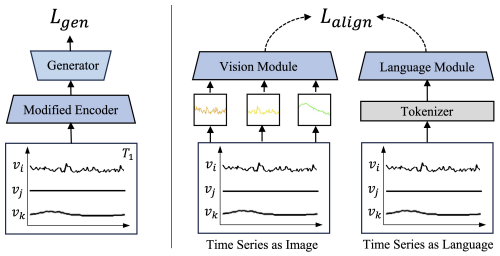
Time series data, consisting of records captured over time, are prevalent in our lives and various application domains [55]. As one of the essential tasks in time series data, time series forecasting, involves predicting future data points by analyzing historical data. Given its widespread demand in real-world scenarios, time series forecasting has attracted considerable research interest from the artificial intelligence and deep learning community, particularly in fields such as meteorology [132, 117, 133], energy [112, 55, 85], healthcare [44, 11], and financial investment [17, 74].

Classic deep learning-based approaches to time series forecasting usually rely on designing unique architectures as different end-to-end forecasting models. Researchers have explored various architectures, including Transformer-based [77, 70], CNN-based [97], MLP-based [79, 131], and RNN-based models [59, 71], to accomplish forecasting. However, these methods mostly treats time series data purely as numerical sequences; in other words, they can be depicted as ***unimodal*** frameworks, as shown in Figure 1(a). Unfortunately, unimodal frameworks are limited in modeling complex patterns, contextual semantics, and long-term dependencies, and, thus, often result into fragile models.

With the success of Large Language Models (LLMs), researchers in time series have brought up the new perspective of ***cross modality***. There are studies that treat time series as a "foreign language" that can be decoded using the language representation strategies of large language models through fine-

*Corresponding author.

Preprint.



(a) Unimodal Frameworks (b) Multimodal Framework

Method	Input	LM	Train Emb.	LM Use	Vision	MM Feat.
Time-LLM	Prompted	LLaMA	✗	LM-based TSF.	✗	✗
Chronos (T5)	Tokenized	T5	✓	LM-based TSF.	✗	✗
PatchTST	Numerical	✗	✓	✗	✗	✗
iTransformer	Numerical	✗	✓	✗	✗	✗
TimesCLIP	**Numerical**	CLIP-Text	✓	Extract Feat.	✓	✓

(c) Comparison of input modalities and modeling strategies.

Figure 1: Comparison of frameworks and modeling strategies. Overview of modeling paradigms for Time Series Forecasting(TSF). The left figure contrasts unimodal and multimodal frameworks, while the right table summarizes representative methods across core modeling dimensions. TimesCLIP uniquely constructs and aligns visual and textual features directly from numerical time series, enabling multimodal learning without requiring external modalities.

tuning [90, 62, 71] or reprogramming dedicated time series encoders [41]. Despite these advances, representing time series as a language is still less intuitive, as humans typically interpret time series data ***visually*** rather than ***textually*** [92]. For example, doctors usually adopt the visual information in ECG to diagnose heart conditions [83]; financial professionals would conduct analysis based on the stock market charts instead of pure numbers [9]. Recent studies have suggested that directly applying LLM techniques to time series forecasting may not always yield effective results [92, 43]. Recent cross-modality studies [56, 86, 7, 57, 47] have converted time series data into images for diverse tasks, but are not performant in time series forecasting [86, 56]. More recent studies have utilized Fourier-transformed spectrograms [7, 111], which, however, don't provide visually interpretative time series representations.

Besides, efforts have been made to extract ***multimodal*** knowledge for time series related domain-specific applications. The studies in [66, 12] attempted to classify different sensors; however, the sensors capture the same type of data-human motion, which does not conform to the true multimodality. In healthcare, the integration of medical time series and disease description texts to construct multimodal frameworks has been explored [83]. However, these applied studies benefit domain-specific data with annotated multimodal information, but are limited in generalized time series analysis and forecasting.

In response to the above challenges, we introduce *TimesCLIP*, a novel Multimodal Contrastive Learning approach for time series modeling and forecasting. Inspired by the effective modeling of CLIP [81] that bridges the gap between text and visual understanding, *TimesCLIP* introduces a novel idea to jointly learn both numeric time series and visual representations. Specifically, it involves converting time series data into images and employing a multimodal contrastive learning framework. Figure 1 (b) shows that, in the vision branch, original numerical variates are transformed into distinct figures using different colors. In the language branch, similar to LLM-based approaches for time series, we treat numerical time series data as a "foreign" language. However, instead of employing a complex projection module to align time series with "real" language, we employ a simpler learning tokenizer to align time series with the feature space of pretrained language model. We further introduce a modified contrastive loss specifically designed for multimodal time series, aiming to align the vision representation with the language representation within a multimodal space. Subsequently, we introduce an innovative variate selection module. We designate the aligned classification feature as "Query" and use a cross-attention layer to identify the most correlative features for downstream tasks in the multivariate feature sequence. This module enables the model to effectively utilize multimodal features and classification information to identify the most significant one among multiple variables for downstream tasks. In brief, we summarize our contribution in threefold:

- We present a novel concept for converting numerical multivariate time series into vision and language representations. We propose an innovative multimodal contrastive learning framework to align time series data with a vision-language multimodal space.
- We introduce a variate selection module designed to leverage aligned multimodal features, enabling the identification of the most relevant variable among the multivariate time series for time series forecasting.
- Extensive experiments demonstrate that our proposed model, TimesCLIP, achieves strong performance on both short-term and long-term forecasting tasks.

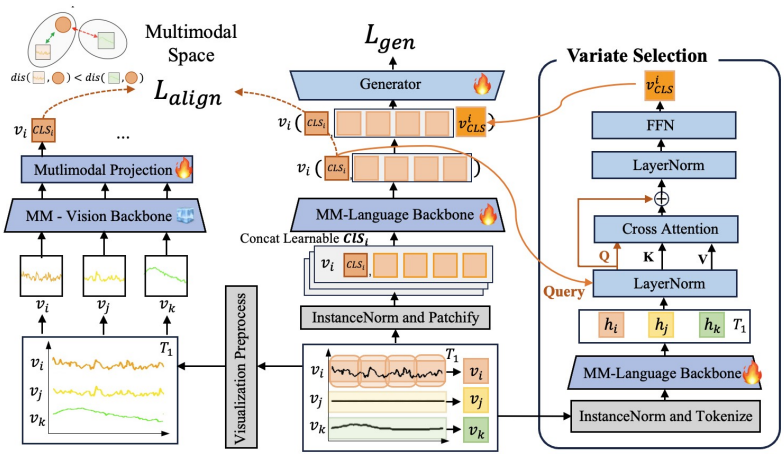


Figure 2: Overview of the TimesCLIP framework. TimesCLIP converts numerical multivariate time series into colorized figures and patchified sequences, which are then processed by pretrained vision and language backbones. A learnable CLS token is appended on the language side to serve as a unified representation for contrastive alignment. A multimodal InfoNCE loss $\mathcal{L}_{\text{align}}$ aligns vision and language features into a shared space. The aligned CLS token is then used to identify task-relevant variates via cross-attention, and passed to a generator for forecasting.

2 Related Work

Multimodal Learning for Time Series. Contrastive learning has significantly advanced the processing and understanding of multimodal data, especially CLIP [81], which illustrates the generalization ability of multimodal contrastive learning. [119] built deep multimodal representation learning from temporal data, which utilizes different modality information, such as audio, image, and language. [87] introduced the Dual Swin Transformer (DuST), a multimodal classification framework, that integrates video with synchronous time series data for driving risk assessment. [50] explored a frozen language model for zero-shot learning on ECG data. [22] integrated structured and unstructured data as different modalities. However, these works require the original dataset to contain more than one modality or do not require aligning different modalities. The benchmarks for time series forecasting [115, 70], only contain recorded numerical data, cannot work with these methods. We summarize representative forecasting paradigms and their use of language models, input modalities, and multimodal strategies in Figure 1(c). [66, 12] claim that they attempt to utilize multimodal learning for human motion classification, a type of time series data. However, they remain constrained by the use of various sensors placed at different positions on the human body for action recognition. They do not adhere to the commonly accepted definition of multimodality in the AI field, which traditionally involves integrating distinct modalities such as vision, audio, language, and depth, rather than using different sensors to record the same modality information. While recent work has explored applying vision-language models (VLMs) to time series data [135], we instead propose a contrastive alignment method specifically designed for numerical time series.

In addition, we include more related work on time series forecasting with language models, vision-language contrastive learning, and time series foundation models in Appendix H.

3 Method

In this section, we first present our proposed model in Section 3.1 for time series forecasting. We explain the modified vision representation module and language module in Section 3.2 and contrastive learning multimodal alignment for time series in Section 3.3, followed by the designed variate selection module in Section 3.4. The aligned language representation of time series data is passed into the generator to forecast future time series data in Section 3.5.

3.1 Overview

Figure 1 shows our general-purpose architecture of multi-modal time series contrastive learning includes two modules: (i) vision module and (ii) language module. To implement the general purpose

architecture, Figure 2 shows our proposed model TimesCLIP. Compared with the proposed general multimodal architecture in Figure 1, TimesCLIP consists of three parts: (i) a multimodal vision module, (ii) a multimodal language module, and (iii) a variate selection module.

The left part of Figure 2 shows the vision module of TimesCLIP contains three blocks: the Visualization Preprocess for time series, a pretrained and frozen multimodal vision backbone to extract time series figure features, and a trainable multimodal projection layer. First, the Visualization Preprocess converts the original time series signals into figures, by using a designed normalization step to normalize the original numerical values and visualize them in the figures with different colors. By converting the original time series into images, we can utilize existing image feature encoders to extract time series features from a visual perspective. Additionally, following [21, 127, 81], we feed vision features to a learnable multimodal projection to transform and represent time series vision features into a multimodal time series feature space. Overall, this vision branch plays the most important role in our proposed multimodal framework, which successfully converts the numerical data of the time series into a visual representation space that is to be aligned with language representation space.

The right part of Figure 2 shows that we patchify each variate of time series data and leverage an online learnable time series tokenizer to project the patchified time series signal into language tokens. After concatenating classification class tokens for each variate, we pass them into the pretrained language encoder to obtain the language representation of time series data. Additionally, the introduced class tokens are utilized for multimodal alignment. More explanation about the modules above is in Section 3.2. Specifically, we introduce a special variate selection module, which utilizes classification information from cross-modal space to select the most significant correlative variate feature from multi-variate time series data. Finally, we feed the aligned language representation into the generator to produce the final forecast.

3.2 Vision-Language Module for Time Series

In time series forecasting, given a length-T historical multivariate time series observation $X^{1:T} = \{v_1^{1:T}, \ldots, v_N^{1:T}\} \in \mathbb{R}^{T \times N}$, $v_i^{1:T} = \{v_i^1, \ldots, v_i^T\}, i \in [1, 2, \ldots, N]$. N represents different recorded variates, and T is the known historical time step. We predict the future S time steps of the time series $Y = X^{1+S:T+S} = \{v_1^{1+S:T+S}, \ldots, v_N^{1+S:T+S}\}$.

Figure 2 shows that, with different preprocessing procedures for time series data, its vision representation and language representation are produced by the vision module and language module, respectively.

Vision module. Since original time series data is numerical signals with multivariate, differences in numerical ranges exist. To visualize these data with reasonable value ranges in the figure, we normalize every variable in a fixed window size L_{window} instead of normalizing in all time series data or mini-batch data. In this way, we reduce the impact of outlier maxima or minima compared to normalizing the values in the minibatch or all data. Next, we plot each time series variable with a specific color to help discriminate variable types to align in the multimodal feature space with the language representation of time series. This process can be denoted as Equation (1).

$$I_i = \text{Visualize}(v_i^{1:T}), i \in [1, \ldots, N] \tag{1}$$

As a result, we convert time series into a sequence of images $X_{\text{img}} = \{I_1, I_2, \ldots, I_N\}$. Inspired by the prior works [81, 127], we input the image sequence of the time series X_{Img} into a pretrained and frozen multimodal vision encoder E_v. This process produces a sequence of feature maps $\{f_1, f_2, \ldots, f_N\}$, where each feature map f_i is generated by $f_i^{\text{img}} = \text{E}_{img}(I_i), i \in [1, 2, \ldots, N]$.

Since the multimodal visual backbone is kept frozen during training, we introduce an additional projection layer Proj_{img} to align the visual features of the time series with the multimodal feature space. Consequently, similar to conventions established in [81, 127, 129], the vision representation is referred to as the vision classification token. We denote this vision representation as $\boldsymbol{CLS}_i^{\text{img}} = \text{Proj}_{\text{img}}(f_i), i \in [1, 2, \ldots, N]$.

Finally, for length-T multiple variates time series data $X_i^{1:T}$, we have its vision representation as $X_{\text{feat}}^{\text{img}} = \{\boldsymbol{CLS}_1^{\text{img}}, \ldots, \boldsymbol{CLS}_N^{\text{img}}\}$.

Language module. Inspired by ViT [21] and Bert [45], PatchTST [77] introduces a patchify strategy that segments time series into subseries-level patches, serving as input tokens for the Transformer. Our method applies this strategy to segment the time series $X^{1:T} = \{v_1^{1:T}, \ldots, v_N^{1:T}\}$ into patches as detailed in Equations (2) to (4). Prior to applying the patchify strategy, we follow the studies in [77, 70, 110] and implement layer normalization [2] to address the problem of non-stationarity in time series data.

$$x_{\text{patch}} = \text{Patchify}(\text{LayerNorm}(x^{1:\text{T}})) \tag{2}$$

$$x_{\text{patch}} = \{v_{\text{patch},1}^{1:\text{PL}}, v_{\text{patch},2}^{1:\text{PL}}, \ldots, v_{\text{patch},N}^{1:\text{PL}}\} \tag{3}$$

$$v_{\text{patch},i}^{1:\text{PL}} = \{v_{\text{patch},i}^{1}, \ldots, v_{\text{patch},i}^{M}\}, i \in [1, 2, \ldots, N], \tag{4}$$

where PL represents the patch length, S is the stride, and $M = \lfloor \frac{T-\text{PL}}{S} \rfloor + 2$ is the number of patches.

Unlike previous time series forecasting models based on the Transformer, such as PatchTST [77], we utilize a multimodal pretrained model instead of training the Transformer from scratch. In prior studies [81, 127], words must be tokenized using a pretrained tokenizer before being fed into the language encoder. However, due to the substantial domain gap between time series data and natural language, we cannot use a pretrained language tokenizer, such as *CLIPTokenizer* [81]. To address this issue, inspired by [77, 70], we introduce a random initialized learnable linear layer that functions as an embedding layer, also referred to as the Tokenizer, for time series.

$$v_{\text{token},i}^{1:\text{PL}} = \text{Tokenizer}(v_{\text{patch},i}^{1:\text{PL}}), i \in [1, 2, \ldots, N] \tag{5}$$

$$v_{\text{token},i}^{1:\text{PL}} = \{v_{\text{token},i}^{1}, v_{\text{token},i}^{2}, \ldots, v_{\text{token},i}^{M}\} \tag{6}$$

Subsequently, following [127], to adopt multimodal language encoder to time series domain, we prepend a learnable embedding v_{cls} to each token sequence of variate $v_{\text{token},i}^{1:PL}$, which we call [class] token [21, 127, 81]. Equation (7) shows that our method learns language representation by utilizing pretrained multimodal language model E_{text} to embed time series into multimodal language representation space.

$$f_i^{\text{text}} = \text{E}_{\text{text}}(\left[v_{\text{cls}}, v_{\text{token},i}^{1}, v_{\text{token},i}^{2}, \ldots, v_{\text{token},i}^{M}\right] + e^{\text{pos}}) \tag{7}$$

where $[.,.]$ represents the concatenation of time series token and $[class]$ token, and e^{pos} represents the learnable position embedding. Notably, we use the same $[class]$ token for each given variate $v_{\text{patch},i}^{1:\text{PL}}$. Consequentially, we obtain language representations of time series as Equation (8)

$$X_{\text{feat}}^{\text{text}} = \{f_1^{\text{text}}, f_2^{\text{text}}, \ldots, f_N^{\text{text}}\} \tag{8}$$

Following the contrastive learning strategies in [21, 81, 127], we regard the first token of f_i^{text} as the [class] token, denoted by $CLSi^{\text{text}}$, to align with the vision representation. Consequently, we define $X_{\text{cls}}^{\text{text}} = \{CLS_1^{\text{text}}, \ldots, CLS_N^{\text{text}}\}$.

3.3 Multimodal Contrastive Loss

Our proposed vision-language module (Section 3.2) processes a length-T multivariate time series $X^{1:T}$ to obtain the vision representation $X_{\text{feat}}^{\text{img}}$ and the language classification representation $X_{\text{cls}}^{\text{text}}$ of the time series, where $X_{\text{feat}}^{\text{img}}, X_{\text{cls}}^{\text{text}} \in \mathbb{R}^{N \times D}$, N represents the number of variables, and D is the feature dimension. Then, by using a batch of data to calculate the loss, we obtain the vision representations $V = \{X_{\text{feat},1}^{\text{img}}, \ldots, X_{\text{feat},B}^{\text{img}}\}$ and language classification representations $L = \{X_{\text{cls},1}^{\text{text}}, \ldots, X_{\text{cls},B}^{\text{text}}\}$, where B denotes the batch size.

After that, we incorporate the multimodal contrastive alignment for time series into the standard contrastive framework [81], based on InfoNCE loss [78] as follows:

$$\mathcal{L}_{\text{cont}}(V, L) = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\varphi(X_{\text{cls},i}^{\text{img}}, X_{\text{cls},i}^{\text{text}})/\tau)}{\sum_{j=1}^{B} \exp(\varphi(X_{\text{cls},i}^{\text{img}}, X_{\text{cls},j}^{\text{text}})/\tau)} \tag{9}$$

$$\varphi(X_{\text{cls},i}^{\text{img}}, X_{\text{cls},i}^{\text{text}}) = \frac{X_{\text{cls},i}^{\text{img}}}{\left\|X_{\text{cls},i}^{\text{img}}\right\|} \cdot \frac{X_{\text{cls},i}^{\text{text}\,T}}{\left\|X_{\text{cls},i}^{\text{text}\,T}\right\|} \tag{10}$$

where τ is a learnable temperature parameter [81, 127] and $\varphi(\cdot, \cdot)$ denotes cosine similarity. Each multimodal pair represents N variates from a length-T time series. Positive pairs are formed between CLS_i^{img} and CLS_i^{text} from the same sample, while negatives come from different samples in the batch. We compute the contrastive loss bidirectionally:

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{cont}}(V, L) + \mathcal{L}_{\text{cont}}(L, V). \tag{11}$$

3.4 Variate Selection

We design a novel variate selection module to effectively utilize information through the variates of original time series data across different views. Inspired by [127, 51], we leverage the $[class]$ token from the language module as the "Query" to identify the most correlative time series variate-level token via a cross-attention mechanism. Time series tokens are generated by tokenizing the time series data [70] and then encoding them with the shared multimodal language model described in Section 3.2. Although this feature extraction step is similar to the approach in [70], we utilize a shared pretrained multimodal Transformer model, instead of training one from scratch.

Variate Tokenizer. Note that in our model, given a multivariate time series $X^{1:T}$ of length T, the approach diverges from the patchify strategy used in the language module, as described in Section 3.2. Each variable $v_i \in \mathbb{R}^{1\times T}$ is treated as an individual token, without applying the patchification process, where N denotes the number of variables, and T denotes the time steps. As a result, $X^{1:T}$ can be represented as $X \in \mathbb{R}^{N\times T}$. Post-tokenization, the time series is transformed to $X_{\text{var}}^{1:T} \in \mathbb{R}^{N\times D}$, where D is the hidden dimension. We pass $X_{\text{var}}^{1:T} = \{v_{\text{var}}^1, v_{\text{var}}^2, \ldots, v_{\text{var}}^N\}$ to the pretrained multimodal language encoder $\text{E}_{\text{text}}^{var}$ and produce variate-level time series representations $H = \{h_1, h_2, \ldots, h_N\}$. This process is denoted as Equation (12).

$$h_i = \text{E}_{\text{text}}^{var}(v_{\text{var}}^i), i \in [1, 2, \ldots, N] \quad (12) \qquad v_{\text{CLS}}^i = CLS_i^{\text{text}} + \text{Softmax}\left(\frac{(W_q CLS_i^{\text{text}})(W_k H)^T}{\sqrt{d_k}}\right) W_v H \quad (13)$$

The variate-level time series representation H and CLS_i^{text}, named Query in Figure 2, from the multimodal language model discussed in Section 3.2, are normalized using layer normalization [2]. Subsequently, we select the most correlative variate feature by employing a cross-attention layer, an adaptation of Transformer decoder[95], with CLS_i^{text} serving as the Query and H as both the Key and Value in the cross-attention layer (Figure 2). Formally, this procedure is defined as:

where W_q, W_k, and W_v are the weight matrices for the Query, Key, and Value projections, respectively, in the cross-attention mechanism. d_k represents the dimensionality of the per-head dimension of multi-head attention[95, 51]. Finally, v_{CLS}^i is passed through a LayerNorm and a two-layer FFN with nonlinearity. Together with the cross-attention layers in Equation (13), these components form a modified Transformer decoder [95].

3.5 Generator

After obtaining v_{CLS}^i for each variate from the variate selection module, we combine them with the multimodal language representation of the time series. Rather than simply concatenating them at the end of the feature sequence, we replace the last token of the multimodal language representation with v_{CLS}^i. This replacement is motivated by the padding strategy used in the Patchify process described in Equation (2). Further details and ablation studies on this fusion strategy can be found in Appendix D.2. Finally, following [77], we apply a flattening layer followed by a linear head to generate the forecasting result $\hat{Y} = \hat{X}^{1+S:T+S}$.

3.6 Training Loss

We adopt the mean squared error (MSE) loss for all forecasting tasks, except for the M4 dataset where we follow prior work[116] and use the Symmetric Mean Absolute Percentage Error (SMAPE) loss:$\mathcal{L}_{\text{SMAPE}} = \frac{200}{B}\sum_{i=1}^{B}\frac{|Y_i - \hat{Y}_i|}{|Y_i| + |\hat{Y}_i|}$. We refer to both $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{SMAPE}}$ as the predictive loss $\mathcal{L}_{\text{gen}}$. The overall training objective combines predictive and contrastive terms:$\mathcal{L} = \lambda_1\mathcal{L}_{\text{gen}} + \lambda_2\mathcal{L}_{\text{align}}$, where λ_1 and λ_2 are weighting coefficients.

4 Experiments

We introduce the implementation details and evaluation metrics in Section 4.1. We then evaluate the effectiveness of our proposed method for short-term forecasting in Section 4.2 and long-term forecasting in Section 4.3.

Table 1: Full results for the short-term forecasting task in the M4 dataset. X. in the Transformers indicates the name of X-former. *Stationary* means the Non-stationary Transformer.

Models		MM-Based	LLM-Based		Uni-modality Models (End to End)															
		TimesCLIP (Ours)	Time-LLM* [2024]	GPT4TS* [2023]	iTrans. [2024]	TimesNet [2022]	N-HiTS [2023]	N-BEATS* [2019]	ETS. [2022]	LightTS [2023]	DLinear [2023]	FED. [2022]	Stationary [2022]	Auto. [2021]	Pyra. [2022]	In. [2021]	LogTrans [2019]	Re. [2020]	LSTM [1997]	TCN [2019]	S4 [2021]
Yearly	SMAPE↓	**13.188**	13.419	15.110	14.141	13.387	13.418	13.436	18.009	14.247	16.965	13.728	13.717	13.974	15.530	14.727	17.107	16.169	176.040	14.920	61.675
	MASE↓	**2.95**	3.005	3.565	3.179	2.996	3.045	3.043	4.487	3.109	4.283	3.048	3.078	3.134	3.711	3.418	4.177	3.800	31.033	3.364	19.953
	OWA↓	**0.775**	0.789	0.911	0.833	0.786	0.793	0.794	1.115	0.827	1.058	0.803	0.807	0.822	0.942	0.881	1.049	0.973	9.290	0.880	4.397
Quarterly	SMAPE↓	**10.007**	10.110	10.597	10.739	10.100	10.202	10.124	13.376	11.364	12.145	10.792	10.958	11.338	15.449	11.360	13.207	13.313	172.808	11.122	65.999
	MASE↓	**1.166**	1.178	1.253	1.282	1.182	1.194	1.169	1.906	1.328	1.520	1.283	1.325	1.365	2.350	1.401	1.827	1.775	19.753	1.360	17.662
	OWA↓	**0.880**	0.889	0.938	0.955	0.890	0.899	0.886	1.302	1.000	1.106	0.958	0.981	1.012	1.558	1.027	1.266	1.252	15.049	1.001	9.436
Monthly	SMAPE↓	**12.502**	12.980	13.258	13.727	12.670	12.791	12.677	14.588	14.014	13.514	14.260	13.917	13.958	17.642	14.062	16.149	20.128	143.237	15.626	64.664
	MASE↓	**0.929**	0.963	1.003	1.082	0.933	0.969	0.937	1.368	1.053	1.037	1.102	1.097	1.103	1.913	1.141	1.660	2.614	16.551	1.274	16.245
	OWA↓	**0.870**	0.903	0.931	0.984	0.878	0.899	0.880	1.149	0.981	0.956	1.012	0.998	1.002	1.511	1.024	1.340	1.927	12.747	1.141	9.879
Others	SMAPE↓	**4.587**	4.795	6.124	5.569	4.891	5.061	4.925	7.267	15.880	6.709	4.954	6.302	5.485	24.786	24.460	23.236	32.491	186.282	7.186	121.844
	MASE↓	**3.116**	3.178	4.116	4.052	3.302	3.216	3.391	5.240	11.434	4.953	3.264	4.064	3.865	18.581	20.960	16.288	33.355	119.294	4.677	91.650
	OWA↓	**0.974**	1.006	1.259	1.225	1.035	1.040	1.053	1.591	3.474	1.487	1.036	1.304	1.187	5.538	5.879	5.013	8.679	38.411	1.494	27.273
Weighted Average	SMAPE↓	**11.642**	11.983	12.690	12.699	11.829	11.927	11.851	14.718	13.525	13.639	12.840	12.780	12.909	16.987	14.086	16.018	18.200	160.031	13.961	67.156
	MASE↓	**1.560**	1.595	1.808	1.761	1.585	1.613	1.599	2.408	2.111	2.095	1.701	1.756	1.771	3.265	2.718	3.010	4.223	25.788	1.945	21.208
	OWA↓	**0.837**	0.859	0.940	0.929	0.851	0.861	0.855	1.172	1.051	1.051	0.918	0.930	0.939	1.480	1.230	1.378	1.775	12.642	1.023	8.021

∗ For N-BEATS[79], we follow the experiment result in [115], which remove the special ensemble method. We adopt the performance of Time-LLM and GPT4TS from [41].

Table 2: Average results on PEMS and illness datasets. Full results are provided in Table 10.

Model	TimesCLIP Ours		iTransformer [2024]		PatchTST [2022]	
Metric	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓
PEMS03	**0.176**	**0.261**	0.250	0.306	0.299	0.363
PEMS04	**0.098**	**0.216**	0.121	0.232	0.312	0.377
PEMS03	**0.115**	**0.226**	0.128	0.237	0.202	0.293
PEMS07	**0.358**	**0.773**	1.334	1.480	1.009	1.351
illness	**1.986**	**0.878**	2.261	0.961	2.137	0.892

Table 3: Forecasting results on EPF datasets (multivariate, 168→24).

Model	TimesCLIP Ours		iTransformer [2024]		PatchTST [2022]	
Metric	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓
Nord Pool	**0.293**	**0.325**	0.338	0.352	0.307	0.336
PJM	**0.076**	**0.178**	0.079	0.179	0.081	0.187
EPEX-BE	**0.142**	**0.162**	0.149	0.174	0.158	0.179
EPEX-FR	**0.143**	**0.151**	0.148	0.154	0.159	0.167
EPEX-DE	**0.190**	**0.245**	0.212	0.261	0.210	0.258

4.1 Experimental Details

We employ the pretrained CLIP vision and language encoders, both based on ViT-B [81], as our vision and language backbones, respectively. In our experiments, we freeze the vision backbone within our vision module and fine-tune the pretrained CLIP text encoder during training. Additionally, we replace CLIP's original token embedding with a new embedding trained from scratch. Unlike end-to-end models [77, 70, 115], which require modifications for different benchmarks, our proposed method maintains the same architecture, including hidden dimensions and module layers. To ensure a fair evaluation of method effectiveness, we implement our model without any fine-tuning techniques, such as Parameter-Efficient Fine-Tuning (PEFT) [34], which are commonly used to accelerate training and enhance performance. More implementation details are provided in the Appendix A.

Evaluation Metrics. Following [70], we adopt mean absolute error(MAE) and mean square error(MSE) for long-term forecasting. Following [115, 79], we use symmetric mean absolute percentage error(SMAPE), mean absolute scaled error(MASE) and overall weighted average (OWA) for short-term forecasting. More details are provided in Appendix B.

4.2 Short-term Forecasting

M4 dataset. Following [70, 41], we select the M4 dataset as the benchmark for short-term forecasting. More detailed dataset description and model hyperparameters are provided in the Appendix E.1.

Baselines. Following [116], we extensively compared 19 models, categorized into three architectures: (1) Multimodal (MM)-based model, our proposed TimesCLIP (2) LLM-based models such as Time-LLM [41] and GPT4TS [140]. (3) Unimodal end-to-end models: LSTM [1997], S4 [2021], TCN [2019], TimesNet [115], N-HiTS [2023], N-BEATS [2019], LightTS [2023], , DLinear [2023], iTransformer [2024], Reformer [2020], Informer [2021], Pyraformer [2022], Autoformer [2021], FEDformer [2022], Non-stationary Transformer [2022], and ETSformer [2022].

As shown in Table 1, our MM-based model, TimesCLIP, achieves the best performance across all datasets and metrics compared to all baselines.

EPF dataset. We follow [65] in using the EPF dataset, but extend the original 2→1 setting to a multivariate-to-multivariate configuration (3 inputs → 3 outputs) to better evaluate short-term multivariate forecasting. Only two baselines are included due to the revised setup. Additionally, we

treat multi-scenario datasets (e.g., M4, EPF, PEMS) as multiple independent short-term forecasting tasks, resulting in a total of fifteen short-term benchmarks in our evaluation.

4.3 Long-term Forecasting

We conduct experiments on six standard long-term forecasting datasets: Electricity (ECL), Exchange, Traffic, Weather, Illness, ETTm1, and ETTm2. While our setup is based on the widely-adopted benchmark of TimesNet [115], we intentionally omit prediction lengths of 336 and 720, as these represent forecasting horizons up to 7.5 times longer than the context length. Instead, we fix the input sequence length to 96 and evaluate on horizons of 96 and 192, which reflect more realistic forecasting scenarios. This choice is motivated by recent studies [8, 1, 77] that emphasize the importance of aligning context and prediction ranges in practical applications.

Baselines. Following [70], we choose 13 time series forecasting models as our benchmark, including: (1) Our method, TimesCLIP, is the only MM-based method. (2) LLM-based methods: Time-LLM[42] and GPT4TS[139]. (3) Uni-modality models: iTransformer[70], Autoformer[114], FEDformer[137], Stationary[69], Crossformer[134], PatchTST[77], DLinear[128], TiDE[18], RLinear[58], SCINet[65], and TimesNet[116]. The results of uni-modal models and GPT4TS from [70, 139], however, the results of Time-LLM[2024] from [92]. More details and discussion are in Appendix E.2.

The results are illustrated in Table 4, **Bold** indicates the best performance, and Underline represents the second-best. Our multimodal(MM)-based model, TimesCLIP, achieves the best performance on six datasets and secures the most first-place rankings in long-term forecasting tasks.

Table 4: Full results of the long-term forecasting task. The prediction lengths are set to {96,192}, following the setting of TimesNet [2022]. The input sequence length is set to 96 for all baselines. *Avg* denotes the average results across all four prediction lengths.

Models		MM-Based		LLM-Based				Uni-Models (End-to-End)																			
		TimesCLIP Ours		Time-LLM [2024]		GPT4TS [2023]		iTransformer [2024]		RLinear [2023]		PatchTST [2022]		Crossformer [2023]		TiDE [2023]		TimesNet [2022]		DLinear [2023]		SCINet [2022]		FEDformer [2022]		Stationary [2022]		Autoformer [2021]	
Metric		MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓
Exch.	96	**0.083**	**0.204**	0.123	0.251	0.096	0.218	0.086	0.206	0.093	0.217	0.088	0.205	0.256	0.367	0.094	0.218	0.107	0.234	0.088	0.218	0.267	0.396	0.148	0.278	0.111	0.237	0.197	0.323
	192	**0.172**	**0.298**	0.224	0.344	0.182	0.307	0.177	0.299	0.184	0.307	0.176	0.299	0.470	0.509	0.184	0.307	0.226	0.344	0.176	0.315	0.351	0.459	0.271	0.315	0.219	0.335	0.300	0.369
	Avg	**0.128**	**0.251**	0.174	0.298	0.139	0.263	0.132	0.253	0.139	0.262	0.132	0.252	0.363	0.438	0.139	0.263	0.167	0.289	0.132	0.267	0.309	0.428	0.210	0.297	0.165	0.286	0.249	0.346
Traffic	96	**0.390**	**0.262**	0.392	0.267	0.388	0.282	0.395	0.268	0.649	0.389	0.462	0.295	0.522	0.290	0.805	0.493	0.593	0.321	0.650	0.396	0.788	0.499	0.587	0.366	0.612	0.338	0.613	0.388
	192	**0.403**	**0.274**	0.409	0.271	0.407	0.290	0.417	0.276	0.601	0.366	0.466	0.296	0.530	0.293	0.756	0.474	0.617	0.336	0.598	0.370	0.789	0.505	0.604	0.373	0.613	0.340	0.616	0.382
	Avg	**0.397**	**0.268**	0.401	0.269	0.398	0.286	0.406	0.272	0.625	0.378	0.464	0.296	0.526	0.292	0.781	0.484	0.605	0.329	0.624	0.383	0.789	0.502	0.596	0.370	0.613	0.339	0.615	0.385
Weather	96	**0.153**	**0.199**	0.155	0.199	0.162	0.212	0.174	0.214	0.192	0.232	0.177	0.218	0.158	0.230	0.202	0.261	0.172	0.220	0.196	0.255	0.221	0.306	0.217	0.296	0.173	0.223	0.266	0.336
	192	**0.202**	**0.246**	0.223	0.261	0.204	0.248	0.221	0.254	0.240	0.271	0.225	0.259	0.206	0.277	0.242	0.298	0.219	0.261	0.237	0.296	0.261	0.340	0.276	0.336	0.245	0.285	0.307	0.367
	Avg	**0.178**	**0.223**	0.189	0.230	0.183	0.230	0.198	0.234	0.216	0.252	0.201	0.239	0.182	0.254	0.222	0.280	0.196	0.241	0.217	0.276	0.241	0.323	0.247	0.316	0.209	0.254	0.287	0.352
ETTm1	96	**0.319**	**0.358**	0.294	0.341	0.300	0.340	0.334	0.368	0.355	0.376	0.329	0.367	0.404	0.426	0.364	0.387	0.338	0.375	0.345	0.372	0.418	0.438	0.379	0.419	0.386	0.398	0.505	0.475
	192	**0.367**	0.387	0.341	0.369	0.343	0.000	0.377	0.391	0.391	0.392	**0.367**	**0.385**	0.450	0.451	0.398	0.404	0.374	0.387	0.380	0.389	0.439	0.450	0.426	0.441	0.459	0.444	0.553	0.496
	Avg	**0.343**	**0.373**	0.318	0.355	0.322	0.170	0.356	0.380	0.373	0.384	0.348	0.376	0.427	0.439	0.381	0.396	0.356	0.381	0.363	0.381	0.429	0.444	0.403	0.430	0.423	0.421	0.529	0.486
ETTm2	96	0.176	**0.258**	0.162	0.248	0.163	0.249	0.180	0.264	0.182	0.265	**0.175**	0.259	0.287	0.366	0.207	0.305	0.187	0.267	0.193	0.292	0.286	0.377	0.203	0.287	0.192	0.274	0.255	0.339
	192	0.242	**0.302**	0.255	0.304	0.222	0.291	0.250	0.309	0.246	0.304	0.241	**0.302**	0.414	0.492	0.290	0.364	0.249	0.309	0.284	0.362	0.399	0.445	0.269	0.328	0.280	0.339	0.281	0.340
	Avg	0.209	**0.280**	0.199	0.276	0.193	0.270	0.215	0.287	0.214	0.285	**0.208**	0.281	0.351	0.429	0.249	0.335	0.218	0.288	0.239	0.327	0.343	0.411	0.236	0.308	0.236	0.307	0.268	0.340
Illness	96	**0.203**	0.248	-	-	-	-	0.206	**0.237**	0.322	0.339	0.208	0.251	0.310	0.331	0.312	0.399	0.250	0.292	0.290	0.378	0.237	0.344	0.242	0.342	0.215	0.249	0.884	0.711
	192	0.244	**0.261**	-	-	-	-	0.241	0.264	0.359	0.356	0.239	0.272	0.734	0.725	0.339	0.416	0.296	0.318	0.320	0.398	0.280	0.380	0.285	0.380	0.254	0.272	0.834	0.692
	Avg	**0.224**	0.254	-	-	-	-	**0.224**	0.250	0.341	0.348	0.224	0.262	0.522	0.528	0.326	0.408	0.273	0.305	0.305	0.388	0.259	0.362	0.264	0.361	0.235	0.261	0.859	0.702

Few-shot Learning. We conduct experiments with few-shot learning on {5%, 10%, 20%} of the data from the Exchange dataset. The results are shown in Table 8.

5 Analysis

In this section, we first analyze the impact of multimodal alignment and variate selection module in Sections 5.1 and 5.2. To comprehensively evaluate our proposed multimodal contrastive learning framework, we conduct ablation studies of different vision and language backbones in Section 5.3. Furthermore, we discuss the limitations and broader impact of our work.

5.1 Ablation of Multimodal Alignment

We perform a comprehensive ablation study on both short-term (M4) and long-term (Exchange) forecasting benchmarks to evaluate the contributions of key components in TimesCLIP. Since the M4 dataset contains only a single variate, colorization is not applicable there; thus, the ablation of colorization is conducted on Exchange.

As shown in Table 5, the multimodal contrastive loss $\mathcal{L}_{\text{align}}$ and Variate Selection module are essential for strong performance. In particular, colorization, the assignment of fixed colors to each variate, enables the vision encoder to visually distinguish variables, which is crucial for effective contrastive learning. Without colorization, the selection mechanism based on the visual 'Query' becomes

significantly less effective. Further analysis of the normalization and colorization steps is provided in Appendix D.1.

We further validate the general effectiveness of multimodal alignment across different vision and language backbones in Section 5.3 and Table 6, where the alignment consistently improves forecasting performance regardless of the specific encoder choice.

Table 5: Ablation study of our proposed module.

Method	Vision Module		Variable	M4 Weighted Avg.			Exchange	
	L_{align}	Col.	Selection	SMAPE↓	MASE↓	OWA↓	MSE↓	MAE↓
Ours	✗	✗	✗	11.782	1.578	0.847	0.389	0.414
	✓	✗		-	-	-	0.383	0.413
	✓	✓		11.725	1.572	0.843	0.376	0.410
	✓	✗	✓	-	-	-	0.383	0.424
	✓	✓	✓	**11.642**	**1.560**	**0.837**	**0.335**	**0.394**

Table 6: Results with different pretrained vision and language backbones on M4.

Language Backbone	Vision Backbone	MM Pretrained	L_{align}	M4 Weighted Avg. SMAPE↓	MASE↓	OWA↓
BERT-base	-	✗	✗	16.356	2.234	1.187
BERT-base	ResNet50		✓	17.898	2.611	1.342
BERT-base	ViT-B/16		✓	17.927	2.634	1.350
BERT-base	SwinT-B		✓	17.954	2.643	1.353
T5	-	✗	✗	11.798	1.579	0.848
T5	CLIP-ViT		✓	11.775	1.579	0.847
CLIP-Text	-	✗	✗	11.782	1.578	0.847
CLIP-Text	ResNet50		✓	11.681	1.565	0.840
CLIP-Text	ViT-B/16		✓	**11.640**	**1.557**	**0.836**
CLIP-Text	SwinT-B		✓	11.676	1.566	0.840
CLIP-Text	CLIP-Vision	✓	✓	<u>11.642</u>	<u>1.560</u>	<u>0.837</u>

5.2 Ablation of Variate Selection

We present ablation studies to assess the effectiveness of variate selection module within our proposed framework. Because our method needs a vision branch to obtain the $class$ token, which will serves as the "Query" for the Variable Selection module, we only evaluate variable selection when the vision module is working. The results in Table 5 indicate that variable selection can help our method achieve better performance. It also benefits from colorization. Since we leverage a shared language backbone for the Variable Selection module, enabling the variable selection module does not introduce extra parameters except for additional learnable positional embedding.

5.3 Ablation of Backbone

We investigate the impact of different pretrained backbones on forecasting performance, including vision encoders such as ViT-B/16 [2020], Swin Transformer [2021], ResNet50 [2016], and CLIP-Vision [81], as well as language encoders like BERT [2019], T5 [2020], and CLIP-Text [81]. As shown in Table 6, CLIP-Text consistently outperforms BERT and T5, even without the vision module. This suggests that language representations learned in a multimodal space are more effective for time series forecasting than those trained on textual corpora alone.

Prior work [93] shows that prompting-based methods like Time-LLM, which use frozen LLMs (e.g., LLaMA) for autoregressive generation, perform poorly in forecasting tasks. More recent approaches such as Chronos [1] improve performance by re-learning the embedding layer of T5 and formulating forecasting as a token-level generation problem.

In contrast, our method takes a fundamentally different approach: instead of treating forecasting as a language modeling task, we align time series with a multimodal representation space via contrastive learning. This avoids the limitations of both prompting and generation, and leads to more robust and generalizable forecasting performance. This alignment-based strategy avoids reliance on discrete token sequences and preserves continuous semantics across views.

Broader Impact and Limitations. In multivariate time series forecasting, our proposed model requires converting each variate into an image and requires significant GPU memory for extracting image features. Additionally, GPU memory requirements increase as the token length and the number of variables grow. Moreover, similar to CLIP [2021], our proposed multimodal framework can align numerical data with both vision and language representations. It has the potential to extend to vision-based large language models and time series foundation models.

6 Conclusion

We aim to address the limitations of previous works that solely relied on unimodal approaches for time series forecasting. We present a novel multimodal contrastive learning model, called TimesCLIP. This approach successfully aligns time series data with a multimodal vision-language space by converting the original numerical data points into colorized images and interpreting the time series data as a "foreign" language. Moreover, to efficiently leverage the aligned multimodal features, we designed a variate selection module to identify the most correlative variate feature from the multivariate time series feature sequence. Extensive experiments demonstrate that our proposed method surpasses established baselines in both short-term and long-term forecasting.

References

[1] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

[2] Jimmy Lei Ba. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Haoyue Bai, Wangyang Ying, Nanxu Gong, Xinyuan Wang, Hao Liu, and Yanjie Fu. Privacy preserving generative feature transformation.

[4] Haoyue Bai, Min Hou, Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, and Meng Wang. Gorec: a generative cold-start recommendation framework. In *Proceedings of the 31st ACM international conference on multimedia*, pages 1004–1012, 2023.

[5] Haoyue Bai, Le Wu, Min Hou, Miaomiao Cai, Zhuangzhuang He, Yuyang Zhou, Richang Hong, and Meng Wang. Multimodality invariant learning for multimedia-based new item recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 677–686, 2024.

[6] Haoyue Bai, Guodong Chen, Wangyang Ying, Xinyuan Wang, Nanxu Gong, Sixun Dong, Giulia Pedrielli, Haoyu Wang, Haifeng Chen, and Yanjie Fu. Brownian bridge augmented surrogate simulation and injection planning for geological co _2 storage. *arXiv preprint arXiv:2505.18204*, 2025.

[7] Silvio Barra, Salvatore Mario Carta, Andrea Corriga, Alessandro Sebastian Podda, and Diego Reforgiato Recupero. Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica*, 7(3):683–692, 2020.

[8] Christoph Bergmeir. Fundamental limitations of foundational forecasting models: The need for multimodality and rigorous evaluation. Invited Talk at the NeurIPS 2024 Workshop on Time Series in the Age of Large Models, December 2024. https://neurips.cc/virtual/2024/108471.

[9] Konstantinos Bisiotis, Stelios Psarakis, and Athanasios N Yannacopoulos. Control charts in financial applications: An overview. *Quality and Reliability Engineering International*, 38(3): 1441–1462, 2022.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] C Bui, N Pham, A Vo, A Tran, A Nguyen, and T Le. Time series forecasting for healthcare diagnosis and prognostics with the focus on cardiovascular diseases. In *6th International Conference on the Development of Biomedical Engineering in Vietnam (BME6) 6*, pages 809–818. Springer, 2018.

[12] Ruichu Cai, Zhifang Jiang, Zijian Li, Weilin Chen, Xuexin Chen, Zhifeng Hao, Yifan Shen, Guangyi Chen, and Kun Zhang. From orthogonality to dependency: Learning disentangled representation for multi-modal time-series sensing signals. *arXiv preprint arXiv:2405.16083*, 2024.

[13] David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S Jensen. Lightts: Lightweight time series classification with adaptive ensemble distillation. *Proceedings of the ACM on Management of Data*, 1(2):1–27, 2023.

[14] Meng Cao, Tianyu Yang, Junwu Weng, Can Zhang, Jue Wang, and Yuexian Zou. Locvtp: Video-text pre-training for temporal localization. *arXiv preprint arXiv:2207.10362*, 2022.

[15] Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 6989–6997, 2023.

[16] Zhenrui Chen, Zhibo Dai, Huiyan Xing, Junyu Chen, Menghao Huo, and Kuan Lu. Multi-model approach for stock price prediction and trading recommendations. *Preprints*, May 2025. doi: 10.20944/preprints202501.1003.v3. URL https://doi.org/10.20944/preprints202501.1003.v3.

[17] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218, 2022.

[18] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

[19] Jiaxiang Dong, Haixu Wu, Yuxuan Wang, Yunzhong Qiu, Li Zhang, Jianmin Wang, and Mingsheng Long. Timesiam: A pre-training framework for siamese time-series modeling. *arXiv preprint arXiv:2402.02475*, 2024.

[20] Sixun Dong, Huazhang Hu, Dongze Lian, Weixin Luo, Yicheng Qian, and Shenghua Gao. Weakly supervised video representation learning with unaligned text for sequential videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2447, 2023.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[22] Sayna Ebrahimi, Sercan O Arik, Yihe Dong, and Tomas Pfister. Lanistr: Multimodal learning from structured and unstructured data. *arXiv preprint arXiv:2305.16556*, 2023.

[23] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

[24] Nanxu Gong, Sixun Dong, Haoyue Bai, Xinyuan Wang, Wangyang Ying, and Yanjie Fu. Agentic feature augmentation: Unifying selection and generation with teaming, planning, and memories. *arXiv preprint arXiv:2505.15076*, 2025.

[25] Nanxu Gong, Zijun Li, Sixun Dong, Haoyue Bai, Wangyang Ying, Xinyuan Wang, and Yanjie Fu. Sculpting features from noise: Reward-guided hierarchical diffusion for task-optimal feature transformation. *arXiv preprint arXiv:2505.15152*, 2025.

[26] Nanxu Gong, Chandan K Reddy, Wangyang Ying, Haifeng Chen, and Yanjie Fu. Evolutionary large language model for automated feature transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16844–16852, 2025.

[27] Nanxu Gong, Xinyuan Wang, Wangyang Ying, Haoyue Bai, Sixun Dong, Haifeng Chen, and Yanjie Fu. Unsupervised feature transformation via in-context generation, generator-critic llm agents, and duet-play teaming. *arXiv preprint arXiv:2504.21304*, 2025.

[28] Nanxu Gong, Wangyang Ying, Dongjie Wang, and Yanjie Fu. Neuro-symbolic embedding for short and effective feature selection via autoregressive generation. *ACM Transactions on Intelligent Systems and Technology*, 16(2):1–21, 2025.

[29] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: a family of open time-series foundation models.. 2024. *arXiv preprint arXiv:2402.03885*, 2024.

[30] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

[31] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[32] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 976–980. IEEE, 2022.

[33] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2906–2916, 2022.

[34] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] Zhuangzhuang He, Yifan Wang, Yonghui Yang, Peijie Sun, Le Wu, Haoyue Bai, Jinqi Gong, Richang Hong, and Min Zhang. Double correction framework for denoising recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1062–1072, 2024.

[37] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.

[38] Huazhang Hu, Sixun Dong, Yiqun Zhao, Dongze Lian, Zhengxin Li, and Shenghua Gao. Transrac: Encoding multi-scale temporal correlation with transformers for repetitive action counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19013–19022, 2022.

[39] Menghao Huo, Kuan Lu, Yuxiao Li, and Qiang Zhu. Ct-patchtst: Channel-time patch time-series transformer for long-term renewable energy forecasting. *arXiv preprint arXiv:2501.08620*, 2025. URL https://arxiv.org/abs/2501.08620.

[40] 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.

[41] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models, 2024. URL https://arxiv.org/abs/2310.01728.

[42] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models, 2024. URL https://arxiv.org/abs/2310.01728.

[43] Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan, and Qingsong Wen. Position: What can large language models tell us about time series analysis. In *Forty-first International Conference on Machine Learning*, 2024.

[44] Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.

[45] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota, 2019.

[46] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[47] Haoran Li, Junqi Liu, Zexian Wang, Shiyuan Luo, Xiaowei Jia, and Huaxiu Yao. Lite: Modeling environmental ecosystems with multimodal large language models. *arXiv preprint arXiv:2404.01165*, 2024.

[48] Haozhou Li, Qinke Peng, Xinyuan Wang, Xu Mou, and Yonghao Wang. Sehf: A summary-enhanced hierarchical framework for financial report sentiment analysis. *IEEE Transactions on Computational Social Systems*, 11(3):4087–4101, 2023.

[49] Haozhou Li, Xinyuan Wang, Hongkai Du, Wentong Sun, and Qinke Peng. Sade: A speaker-aware dual encoding model based on diagbert for medical triage and pre-diagnosis. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12712–12716. IEEE, 2024.

[50] Jun Li, Che Liu, Sibo Cheng, Rossella Arcucci, and Shenda Hong. Frozen language model helps ecg zero-shot learning. In *Medical Imaging with Deep Learning*, pages 402–415. PMLR, 2024.

[51] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[52] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[53] Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. Clip-event: Connecting text and images with event structures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16420–16429, 2022.

[54] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[55] Wenxiang Li and KL Eddie Law. Deep learning models for time series forecasting: a review. *IEEE Access*, 2024.

[56] Xixi Li, Yanfei Kang, and Feng Li. Forecasting with time series imaging. *Expert Systems with Applications*, 160:113680, 2020.

[57] Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series. *Advances in Neural Information Processing Systems*, 36, 2024.

[58] Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

[59] Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv preprint arXiv:2308.11200*, 2023.

[60] Hanghang Liu, Linyi Liu, and Clifford J Rosen. Pth and the regulation of mesenchymal cells within the bone marrow niche. *Cells*, 13(5):406, 2024.

[61] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[62] 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.

[63] Linyi Liu, Sha Leng, Junli Yue, Qian Lu, Weizhe Xu, Xiaowei Yi, Dingming Huang, and Lan Zhang. Edta enhances stromal cell–derived factor 1α–induced migration of dental pulp cells by up-regulating chemokine receptor 4 expression. *Journal of Endodontics*, 45(5):599–605, 2019.

[64] Linyi Liu, Phuong T Le, J Patrizia Stohn, Hanghang Liu, Wangyang Ying, Roland Baron, and Clifford J Rosen. Calorie restriction in mice impairs cortical but not trabecular peak bone mass by suppressing bone remodeling. *Journal of Bone and Mineral Research*, 39(8):1188–1199, 2024.

[65] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

[66] Shengzhong Liu, Tomoyoshi Kimura, Dongxin Liu, Ruijie Wang, Jinyang Li, Suhas Diggavi, Mani Srivastava, and Tarek Abdelzaher. Focal: Contrastive learning for multimodal time-series sensing signals in factorized orthogonal latent space. *Advances in Neural Information Processing Systems*, 36, 2024.

[67] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *# PLACEHOLDER_PARENT_METADATA_VALUE#*, 2022.

[68] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM on Web Conference 2024*, pages 4095–4106, 2024.

[69] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.

[70] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024. URL https://arxiv.org/abs/2310.06625.

[71] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *arXiv preprint arXiv:2402.02370*, 2024.

[72] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*, 2024.

[73] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[74] Wenjie Lu, Jiazheng Li, Jingyang Wang, and Lele Qin. A cnn-bilstm-am method for stock price prediction. *Neural Computing and Applications*, 33(10):4741–4753, 2021.

[75] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019.

[76] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022.

[77] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

[78] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[79] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

[80] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[81] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[82] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[83] Antônio H Ribeiro, Manoel Horta Ribeiro, Gabriela MM Paixão, Derick M Oliveira, Paulo R Gomes, Jéssica A Canazart, Milton PS Ferreira, Carl R Andersson, Peter W Macfarlane, Wagner Meira Jr, et al. Automatic diagnosis of the 12-lead ecg using a deep neural network. *Nature communications*, 11(1):1760, 2020.

[84] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[85] Jason Runge and Radu Zmeureanu. A review of deep learning techniques for forecasting energy use in buildings. *Energies*, 14(3):608, 2021.

[86] Artemios-Anargyros Semenoglou, Evangelos Spiliotis, and Vassilios Assimakopoulos. Image-based time series forecasting: A deep convolutional neural network approach. *Neural Networks*, 157:39–53, 2023.

[87] Liang Shi, Yixin Chen, Meimei Liu, and Feng Guo. Dust: Dual swin transformer for multi-modal video and time-series modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4537–4546, 2024.

[88] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019.

[89] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019.

[90] Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*, 2023.

[91] Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu. Long-form video-language pre-training with multimodal temporal contrastive learning. *arXiv preprint arXiv:2210.06031*, 2022.

[92] Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[93] Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[94] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[95] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[96] Dongjie Wang, Yanyong Huang, Wangyang Ying, Haoyue Bai, Nanxu Gong, Xinyuan Wang, Sixun Dong, Tao Zhe, Kunpeng Liu, Meng Xiao, et al. Towards data-centric ai: A comprehensive survey of traditional, reinforcement, and generative approaches for tabular data transformation. *arXiv preprint arXiv:2501.10555*, 2025.

[97] 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.

[98] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14010–14020, 2022.

[99] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021.

[100] Xinyuan Wang, Qinke Peng, Xu Mou, Haozhou Li, and Ying Wang. A hierarchal bert structure for native speaker writing detection. In *2022 China Automation Congress (CAC)*, pages 3705–3710. IEEE, 2022.

[101] Xinyuan Wang, Haozhou Li, Dingfang Zheng, and Qinke Peng. Lcmdc: Large-scale chinese medical dialogue corpora for automatic triage and medical consultation. *arXiv preprint arXiv:2410.03521*, 2024.

[102] Xinyuan Wang, Dongjie Wang, Wangyang Ying, Rui Xie, Haifeng Chen, and Yanjie Fu. Knockoff-guided feature selection via a single pre-trained reinforced agent. *arXiv preprint arXiv:2403.04015*, 2024.

[103] Xinyuan Wang, Liang Wu, Liangjie Hong, Hao Liu, and Yanjie Fu. Llm-enhanced user-item interactions: Leveraging edge information for optimized recommendations. *arXiv preprint arXiv:2402.09617*, 2024.

[104] Xinyuan Wang, Haoyue Bai, Nanxu Gong, Wangyang Ying, Sixun Dong, Xiquan Cui, and Yanjie Fu. Llm-ml teaming: Integrated symbolic decoding and gradient search for valid and stable generative feature transformation. *arXiv preprint arXiv:2506.09085*, 2025.

[105] Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. Mixllm: Dynamic routing in mixed large language models. *arXiv preprint arXiv:2502.18482*, 2025.

[106] Xinyuan Wang, Dongjie Wang, Wangyang Ying, Haoyue Bai, Nanxu Gong, Sixun Dong, Kunpeng Liu, and Yanjie Fu. Efficient post-training refinement of latent reasoning in large language models. *arXiv preprint arXiv:2506.08552*, 2025.

[107] Xinyuan Wang, Liang Wu, and Yanjie Fu. Enhanced whole page optimization via mixed-grained reward mechanism-adapted language models. *arXiv preprint arXiv:2506.09084*, 2025.

[108] Ying Wang, Qinke Peng, Xu Mou, Xinyuan Wang, Haozhou Li, Tian Han, Zhao Sun, and Xiao Wang. A successful hybrid deep learning model aiming at promoter identification. *BMC bioinformatics*, 23(Suppl 1):206, 2022.

[109] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. 2024.

[110] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *arXiv preprint arXiv:2402.19072*, 2024.

[111] Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. *arXiv preprint arXiv:1506.00327*, 2015.

[112] Rafał Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International journal of forecasting*, 30(4):1030–1081, 2014.

[113] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

[114] 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.

[115] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[116] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[117] Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 5(6):602–611, 2023.

[118] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4563–4567. IEEE, 2022.

[119] Xitong Yang, Palghat Ramesh, Radha Chitta, Sriganesh Madhvanath, Edgar A Bernal, and Jiebo Luo. Deep multimodal representation learning from temporal data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5447–5455, 2017.

[120] Wangyang Ying, Lei Zhang, and Hongli Deng. Sichuan dialect speech recognition with deep lstm network. *Frontiers of Computer Science*, 14(2):378–387, 2020.

[121] Wangyang Ying, Dongjie Wang, Kunpeng Liu, Leilei Sun, and Yanjie Fu. Self-optimizing feature generation via categorical hashing representation and hierarchical reinforcement crossing. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 748–757. IEEE, 2023.

[122] Wangyang Ying, Dongjie Wang, Haifeng Chen, and Yanjie Fu. Feature selection as deep sequential generative learning. *ACM Transactions on Knowledge Discovery from Data*, 18(9): 1–21, 2024.

[123] Wangyang Ying, Dongjie Wang, Xuanming Hu, Yuanchun Zhou, Charu C Aggarwal, and Yanjie Fu. Unsupervised generative feature transformation via graph contrastive pre-training and multi-objective fine-tuning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3966–3976, 2024.

[124] Wangyang Ying, Haoyue Bai, Nanxu Gong, Xinyuan Wang, Sixun Dong, Haifeng Chen, and Yanjie Fu. Bridging the domain gap in equation distillation with reinforcement feedback. *arXiv preprint arXiv:2505.15572*, 2025.

[125] Wangyang Ying, Cong Wei, Nanxu Gong, Xinyuan Wang, Haoyue Bai, Arun Vignesh Malarkkan, Sixun Dong, Dongjie Wang, Denghui Zhang, and Yanjie Fu. A survey on data-centric ai: Tabular learning from reinforcement learning and generative ai perspective. *arXiv preprint arXiv:2502.08828*, 2025.

[126] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

[127] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[128] 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.

[129] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[130] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8552–8562, 2022.

[131] Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022.

[132] Wenqing Zhang, Junming Huang, Ruotong Wang, Changsong Wei, Wenqian Huang, and Yuxin Qiao. Integration of mamba and transformer-mat for long-short range time series forecasting with application to weather dynamics. In *2024 International Conference on Electrical, Communication and Computer Engineering (ICECCE)*, pages 1–6. IEEE, 2024.

[133] Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 619(7970):526–532, 2023.

[134] 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.

[135] Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting, 2025. URL https://arxiv.org/abs/2502.04395.

[136] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[137] 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.

[138] 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.

[139] 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.

[140] 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.

[141] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020.

A Implementation Details

For model initialization, we adopt Kaiming uniform initialization for 1D convolutional layers and Xavier initialization for linear layers. Layer normalization layers are initialized with the bias set to 0 and the weights set to 1. Additionally, the introduced $class$ token is normalized from a normal distribution, $N(0, 0.02)$. We set the weights in $\lambda_1 \mathcal{L}_{gen} + \lambda_2 \mathcal{L}_{align}$ as $\lambda_1 = 1$ and $\lambda_2 = 0.1$, unless otherwise specified. We implement our experiments using on PyTorch[80] and Time Series Library (TSLib)[109, 115]. All experiments are conducted on a single NVIDIA RTX 6000 Ada Generation with a fixed random seed of 2024.

B Benchmarks and Evaluation Metrics

Dim. denotes the number of variables. The dataset sizes are presented as in (Train, Validation, Test). The dataset descriptions are in Table 7, adopted from [115]. For evaluation, following[115], we

Table 7: Dataset statistics from [115].

Tasks	Dataset	Dim	Context → Predict Length	Dataset Size	Information (Frequency)
Forecasting (Long-term)	ETTm1, ETTm2	7	96 →{96, 192 }	(34465, 11521, 11521)	Electricity (15 mins)
	Traffic	862		(12185, 1757, 3509)	Transportation (Hourly)
	Weather	21		(36792, 5271, 10540)	Weather (10 mins)
	Exchange	8		(5120, 665, 1422)	Exchange rate (Daily)
	Solar-Energy	137		(36601,5161,10417)	Energy(10 mins)
Forecasting (short-term)	M4-Yearly	1	6	(23000, 0, 23000)	Demographic
	M4-Quarterly	1	8	(24000, 0, 24000)	Finance
	M4-Monthly	1	18	(48000, 0, 48000)	Industry
	M4-Weakly	1	13	(359, 0, 359)	Macro
	M4-Daily	1	14	(4227, 0, 4227)	Micro
	M4-Hourly	1	48	(414, 0, 414)	Other
	illness	7	36 →{24,36,48, 60 }	(617,74,170)	Illness(Weekly)
	PEMS-03	358	96 → {12,24,48, 96 }	(15617, 5135, 5135)	Transportation(5 mins)
	PEMS-04	307		(10172, 3375, 3375)	Transportation(5 mins)
	PEMS-07	883		(16911, 5622, 5622)	Transportation(5 mins)
	PEMS-08	170		(10690, 3548, 3548)	Transportation(5 mins)
	EPF-NordPool	3	168 → 24	(36500, 5219, 10460)	Electricity Price (1 Hour)
	EPF-PJM	3		(36500, 5219, 10460)	Electricity Price (1 Hour)
	EPF-EPEX-BE	3		(36500, 5219, 10460)	Electricity Price (1 Hour)
	EPF-EPEX-FR	3		(36500, 5219, 10460)	Electricity Price (1 Hour)
	EPF-EPEX-DE	3		(36500, 5219, 10460)	Electricity Price (1 Hour)

utilize the commonly used metrics in previous work[115, 70], including mean square error (MSE) and mean absolute error (MAE) for long-term forecasting. For short-term forecasting, following N-BEATS [79, 115], we adopt the symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MASE), and overall weighted average (OWA) as the metrics. The detailed

calculations are as follows:

$$\text{MAE} = \frac{1}{H}\sum_{i=1}^{H}|\mathbf{X}_i - \widehat{\mathbf{X}}_i|, \qquad \text{MSE} = \frac{1}{H}\sum_{i=1}^{H}(\mathbf{X}_i - \widehat{\mathbf{X}}_i)^2,$$

$$\text{SMAPE} = \frac{200}{H}\sum_{i=1}^{H}\frac{|\mathbf{X}_i - \widehat{\mathbf{X}}_i|}{|\mathbf{X}_i| + |\widehat{\mathbf{X}}_i|}, \qquad \text{MAPE} = \frac{100}{H}\sum_{i=1}^{H}\frac{|\mathbf{X}_i - \widehat{\mathbf{X}}_i|}{|\mathbf{X}_i|},$$

$$\text{MASE} = \frac{1}{H}\sum_{i=1}^{H}\frac{|\mathbf{X}_i - \widehat{\mathbf{X}}_i|}{\frac{1}{H-m}\sum_{j=m+1}^{H}|\mathbf{X}_j - \mathbf{X}_{j-m}|}, \qquad \text{OWA} = \frac{1}{2}\left[\frac{\text{SMAPE}}{\text{SMAPE}_{\text{Naïve2}}} + \frac{\text{MASE}}{\text{MASE}_{\text{Naïve2}}}\right],$$

where m is the periodicity of the data. $\mathbf{X}, \widehat{\mathbf{X}} \in \mathbb{R}^{H\times C}$ are the ground truth and prediction results of the future with H time pints and C dimensions. $\mathbf{X}_i$ means the i-th future time point.

C Few-shot Learning

The experiments in [93] indicate that LLM-based models do not benefit from pretrained large language models. To further evaluate the effectiveness of our multimodal model, we select three well-known end-to-end models as baselines. The results indicate that our method achieves comparable performance in 10%-shot learning. However, due to the feature space gap between the language domain and time series data, our model fails in 5%-shot learning. Additionally, we observe that our model, when using a pretrained backbone, tends to overfit on the training and validation sets when trained with limited data, leading to lower performance on the test set.

Table 8: Few-shot learning on Exchange dataset.

Model		TimesCLIP		PatchTST		iTransformer		TimesNet	
Few-shot	Metric	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓
20%	96	0.121	0.241	0.087	0.208	0.100	0.226	0.153	0.275
	192	0.192	0.314	0.174	0.300	0.196	0.319	0.260	0.366
	336	0.368	0.447	0.290	0.393	0.370	0.446	0.417	0.472
	720	0.938	0.733	0.717	0.647	0.926	0.734	0.962	0.747
	Avg.	0.405	0.434	0.317	0.387	0.398	0.431	0.448	0.465
10%	96	0.102	0.227	0.089	0.210	0.114	0.241	0.152	0.275
	192	**0.187**	0.316	0.194	0.315	0.220	0.339	0.272	0.383
	336	**0.322**	**0.417**	0.325	0.417	0.388	0.458	0.400	0.466
	Avg.	0.204	0.320	0.203	0.314	0.241	0.346	0.275	0.375
5%	96	0.141	0.235	0.108	0.231	0.133	0.262	0.187	0.303
	192	0.671	0.619	0.237	0.351	0.239	0.356	0.271	0.376
	Avg.	0.406	0.427	0.173	0.291	0.186	0.309	0.229	0.340

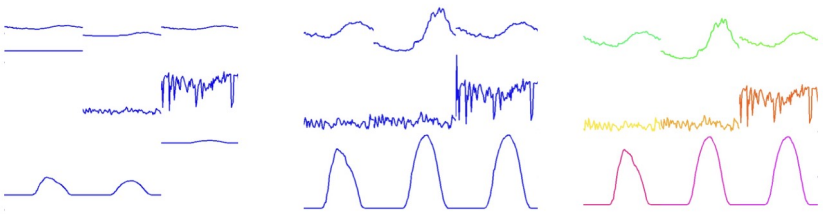
D Extra Ablation Study

D.1 Ablation of Visualization Preprocess

As shown in Figure 3, we visualize the results under two conditions: Norm and Colorize, as introduced in Section 3.2. Figure 3(a) indicates that the original numerical time series data spans a wide range. After normalization, as illustrated in Figure 3(b), we observe that variates with smaller values are magnified, making their periodicity more apparent. While the bottom three variates exhibit similar patterns, colorization helps the model distinguish different variates despite their structural similarity.

D.2 Variate Fusion Strategy

In this section, we conducted an ablation study on the fusion strategy for fusing the obtained v_{CLS} to time series representation sequence as referred to in Section 3.5.



(a) w/o Norm, w/o Colorize (b) Norm, w/o Colorize (c) Norm, Colorize

Figure 3: The visualization of Visualization Preprocess. Sample data is from Weather dataset.

Strategy explanation. As introduced in Section 3.5, we propose a new variate fusion strategy to integrate the extracted correlative variate into the language representation of the time series. In detail, following [77], the patchify process includes a padding step, as detailed in Equation (2). The original length-T time series is padded to T + Padding, where the padding length is half of the stride. Consequently, the last patch contains $X^{T-\frac{S}{2}:T}$ and half padding data, while the second to last patch contains $X^{T-S:T}$. Replacing the last token with v^i_{CLS} helps mitigate the influence of padding on the processing and avoids introducing additional parameters in the linear head. We conduct an ablation study of this specific fusion process in Table 9.

Ablation of fusion strategy. We conducted ablation studies on three different fusion strategies designed to integrate the extracted correlative variate into the language representation of time series as the generator's input. These strategies include concatenating at the end of the feature sequence, replacing the $class$ token, and replacing the last feature of the language representation. As shown in Table 9, the results demonstrate that replacing the last feature of the language representation sequence is the most effective strategy for integrating extracted correlative variable features. As explained in Section 3.5, we think the padding step, introduced by [77], employs "same" padding, and replicating the last length-*Padding* of the time series features may compromise the model's forecasting accuracy.

Table 9: Ablation study on variate fusion strategy

Method	Variate Fusion		M4 Weighted Avg.		
	Strategy	Position	SMAPE↓	MASE↓	OWA↓
Ours	Replace	First	11.687	1.563	0.840
	Concat	End	11.682	1.562	0.839
	Replace	Last	**11.642**	**1.560**	**0.837**

E Experimental Details

In this section, we provide more experimental details and analysis. For experiments result in Tables 1 and 4, if not otherwise stated, we adopt the baselines results from original paper and official code, except Time-LLM[42]. We noticed the short term forecasting results of other baselines in Time-LLM[42] are different with baselines' original paper, such as TimesNet have lower performance in [42], although [42] claim their experimental setups followed [115]. For fair comparison, we adopt Time-LLM results for long-term forecasting from [93], which re-implements three LLM-based models and introduce a novel view that LLMs may not help time series analysis better. Additionally, we adopt the experiment results of Time-LLM for short-term forecasting from [42], because these results have no advantage compared with other baselines' original results. Additionally, all the baselines that we reproduced are implemented based on the configurations of the original paper and official code. It is also notable that no multimodal-based methods are proposed for general time series analysis. Please refer to our code, which shows more experimental details.

E.1 short-term forecasting

We implement our model for short-term forecasting with the hyper-parameters in Table 14. For M4 datasets, we conduct 17 uni-modality end-to-end baselines for short-term forecasting, which

are classified into four architectures: (1)RNN-based models like LSTM [1997] and S4 [2021]; (2) CNN-based models, specifically TCN [2019] and TimesNet [115]; (3) MLP-based models such as N-HiTS [2023], N-BEATS [2019], LightTS [2023], and DLinear [2023]; (4) Transformer-based models including iTransformer [2024], Reformer [2020], Informer [2021], Pyraformer [2022], Autoformer [2021], FEDformer [2022], Non-stationary Transformer [2022], and ETSformer [2022]. For PEMS, illness, and EPF datasets, we compared TimesCLIP with iTransformer and PatchTST. The full results are shown in Tables 1, 3 and 10.

Table 10: PEMS and illness, input 96 for PEMS, input 36 for illness

Model		TimesCLIP Ours		iTransformer [2024]		PatchTST [2022]	
Metric		MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓
PEMS08	12	0.094	0.207	0.088	0.193	0.107	0.221
	24	**0.130**	**0.244**	0.138	0.243	0.167	0.279
	48	**0.212**	**0.276**	0.324	0.348	0.322	0.394
	96	**0.269**	**0.319**	0.450	0.439	0.599	0.557
	Avg	**0.176**	**0.261**	0.250	0.306	0.299	0.363
PEMS04	12	**0.080**	0.201	0.081	0.189	0.109	0.223
	24	**0.088**	**0.207**	0.100	0.212	0.179	0.293
	48	**0.100**	**0.216**	0.135	0.248	0.344	0.414
	96	**0.124**	**0.240**	0.167	0.279	0.615	0.580
	Avg	**0.098**	**0.216**	0.121	0.232	0.312	0.377
PEMS03	12	**0.069**	0.181	0.069	0.174	0.082	0.192
	24	**0.091**	**0.206**	0.098	0.209	0.125	0.235
	48	**0.142**	**0.258**	0.167	0.279	0.220	0.318
	96	**0.157**	**0.257**	0.178	0.287	0.381	0.428
	Avg	**0.115**	**0.226**	0.128	0.237	0.202	0.293
PEMS07	12	**0.063**	**0.175**	0.066	0.164	0.087	0.207
	24	**0.074**	**0.183**	0.087	0.192	0.145	0.264
	48	**0.106**	**0.204**	0.262	0.356	0.284	0.374
	96	**0.116**	**0.210**	0.918	0.768	0.493	0.506
	Avg	**0.358**	**0.773**	1.334	1.480	1.009	1.351
illness	24	**2.007**	**0.869**	2.317	0.941	2.199	0.889
	36	**2.160**	**0.910**	2.199	0.950	2.362	0.917
	48	**1.849**	**0.845**	2.243	0.964	2.028	0.876
	60	**1.927**	**0.889**	2.287	0.990	1.960	0.888
	Avg	**1.986**	**0.878**	2.261	0.961	2.137	0.892

E.2 long-term forecasting

To evaluate our method on the Traffic and ECL datasets, our CLIP-based model results in GPU memory consumption increasing with the number of variables. For instance, if we input 800 variates, the resulting language feature will have the shape $[B \times 800, 12, 512]$, requiring the processing of $B \times 800$ time series figures, which can lead to memory overflow. To address this issue, we modify our model to embed numerical time series using the tokenizer, as described in Section 3.4, and exclude the multimodal contrastive learning loss. The hyper-parameters are shown in Table 11. As a result, the language feature shape becomes $[B, 800, 512]$, reducing the feature size by 12 times compared to the original design. Furthermore, our model relies solely on the pretrained multimodal language model for long-term forecasting.

As discussed in Section 5.3, multimodal language models exhibit strong generalization ability for time series forecasting tasks. The experimental results, shown in Table 4, demonstrate that our simplified multimodal pretrained model still achieves the best performance on the Traffic dataset. We believe that with an efficient fine-tuning strategy, training our proposed model with contrastive loss would further improve its performance.

For Weather and Exchange dataset, we have different hyper-parameters setting depend on prediction length. The details are shown in Tables 12 and 13.

Table 11: Hyper-parameters on Traffic and ECL

Hyper-parameter	Long-term Forecasting	
	Traffic	ECL
Optimizer	AdamW[126]	
LR decay schedule	Cosine Decaying to 0	
Pred Length	96, 192, 336, 720	
train epoch	50	
Batch Size	64	
Pretrained Encoder LR	1e-4	
Others LR	1e-4	
early stop	20	

Table 12: Hyper-parameters on Weather

Hyper-parameter	Weather (long-term)			
Optimizer	AdamW[126]			
LR decay schedule	Exponential Decay, $\gamma = 0.5$			
Pred Length	96	192	336	720
train epoch	10			
Batch Size	8	16	64	4
Pretrained Encoder LR	1e-6	5e-6	1e-6	1e-6
Others LR	1e-4			
early stop	3			
λ_1 (weight of L_{gen})	1			
λ_2 (weight of L_{align})	0.1			

Table 13: Hyper-parameters on Exchange

Hyper-parameter	Exchange (long-term)			
Optimizer	AdamW[126]			
LR decay schedule	Exponential Decay, $\gamma = 0.5$			
Pred Length	96	192	336	720
train epoch	10			
Batch Size	4	8	8	4
Pretrained Encoder LR	1e-5	1e-6	5e-5	5e-6
Others LR	1e-4			
early stop	3			
λ_1 (weight of L_{gen})	1			
λ_2 (weight of L_{align})	0.5	0.1	0.1	0.9

Table 14: Hyper-parameters on short forecasting.

Hyper-parameter	M4, PMES, illness, EPF
Optimizer	AdamW[126]
LR decay schedule	Cosine Decay to 0
train epoch	100
Batch Size	64
Pretrained Encoder LR	1e-4
Others LR	1e-3
λ_1 (weight of L_{gen})	1
λ_2 (weight of L_{align})	0.1
early stop	30

F FLOPs and Parameters Analysis

We compare the efficiency of our model in terms of FLOPs, trainable parameters, and total parameters against two Transformer-based models, one CNN-based model, and two LLM-based models. Note that we implement the same model architecture, based on CLIP-ViT-B [81], across all datasets. The slight increase in parameters is due to the longer learnable position embeddings and the additional class token. In contrast, other baselines are modified for different datasets, leading to variations in model parameters. It is important to note that LLM-based models have a large number of total parameters; however, through Parameter-Efficient Fine-Tuning(PEFT) [34], they fine-tune only a small subset of parameters. This fine-tuning technique is also can be applied to our Transformer-based model to reduce the number of trainable parameters. However, since this work focuses on multimodal contrastive learning, we do not employ such techniques to accelerate training.

Table 15: Comparison of FLOPs and parameters across different methods. Evaluate models with the prediction length set to 96. We adopt the results of Time-LLM and GPT4TS from [93]. Other baselines are evaluated based on their official codes and papers. The *dim/Channel* denotes the hidden dimension of attention layer or CNN's channel depend on model's architecture. Due to Transformer-based model have different dimension of FFN, denoted as *Dim. of FFN*, for instance, PatchTST has 2028 for Weather and 512 for Traffic, which cause different parameters.

Method	Dataset	Enc. Layers	Dim/Channel	Dim. of FFN	FLOPs(G)	Trainable Param.(M)	Total Param.(M)
TimesNet [2022]	Exchange	2	64	64	4.54	4.71	4.71
	Weather	2	32	32	1.13	1.19	1.19
	Traffic	2	512	512	288.12	301.69	301.69
PatchTST [2022]	Exchange	2	512	2048	0.61	6.90	6.90
	Weather	2	512	2048	1.61	6.90	6.90
	Traffic	2	512	512	33.52	3.76	3.76
iTransformer [2024]	Exchange	2	128	128	2.77	0.22	0.22
	Weather	3	512	512	0.123	4.83	3.83
	Traffic	4	512	512	8.63	6.41	6.41
GPT4TS [2023]*	Weather	6 Layers of GPT2			-	-	86
Time-LLM [2024]*	Weather	Llama-7B			-	-	6642
TimesCLIP (Ours)	Exchange	CLIP-ViT-B (12 layers)			39.44	65.85	153.31
	Weather				103.21	65.87	153.32
	Traffic				262.93	66.14	153.59

G Limitation and Further Improvement

Our method is based on pretrained multimodal language encoder, a standard Transformer-based framework. However, the computational complexity of traditional Transformers, which scales as $O(n^2)$, poses a significant challenge when handling multiple variates as extremely long token sequences. This issue became apparent during our evaluation of the model on a traffic dataset, where we were compelled to omit the vision module to conserve GPU resources. Consequently, the number of trainable parameters and the demand for GPU resources increase as the token length grows with the number of variables. Fortunately, recent advancements have addressed these efficiency challenges; studies such as those on the Linear Transformer and lower parameter fine-tuning strategies like LoRA offer promising solutions. We think that designing a more efficient time series imaging strategy and alignment method to reduce resource requirements will extend this work as a foundational multimodal time series representation framework for more downstream tasks.

Furthermore, mapping time-series signals into a unified vision–language embedding space can benefit other modalities as well. For instance, repetitive action counting in videos can be framed as a time-series analysis problem [38]. The proposed multimodal alignment framework can also be applied to other downstream applications such as data-centric AI [96, 125, 3, 24–28, 102, 104, 121–123], linguistics [120, 100], business [36, 103, 4, 5, 48, 107], and medicine [63, 108, 64, 101, 60, 49]. Additionally, the reinforcement learning strategy [124], reasoning process [106], and the rule-based enhancement [6] could also be applied to enhance the robustness of the alignment process.

H Related work

Time Series Forecasting with Language Models. Recent developments in multimodal large language models (LLMs), such as [94, 61], demonstrate the growing multimodal understanding capabilities of LLMs. For instance, [30] directly leverages the sequential processing capabilities of LLMs for numerical forecasting. Furthermore, several researchers have treated time series as a language [40, 139, 42, 90, 62], employing pretrained large language models [10, 94] to enhance forecasting methodologies. Notably, [90] introduced TEST, which employs text prototype-aligned embeddings to augment the ability of LLMs to process time series data. Similarly, [41] proposed Time-LLM, utilizing a novel training strategy named 'reprogramming,' designed to tailor LLMs' capabilities specifically for time series forecasting. However, a recent study [92] suggests that LLM-based time series forecasting models might achieve better performance by replacing their LLM modules with a simpler encoder.

Vision-Text Multimodal Learning In recent years, vision-text multimodal learning [75, 88, 81, 89, 141, 14, 33, 99, 98, 127, 16] has attracted increasing attention within the computer vision community. One of the most notable contributions is CLIP [81], which successfully learns multimodal visual representations guided by natural language. CLIP utilizes a contrastive learning loss, specifically the InfoNCE loss [78], to align visual representations with language representations within a shared multimodal space. The zero-shot ability of multimodal contrastive learning has spurred a multitude of follow-up works across various domains to align different modalities to a unified multimodal space, such as image domain [51, 127, 76], video domain [89, 99, 20, 33, 53, 91], 3D domain [84, 130, 39], and audio domain [32, 118]. Moreover, recent multimodal large language models[61, 52] continue to employ similar frameworks to integrate diverse modal representations with language representation spaces, unlocking the potential of LLMs. LLM routing can be applied to leverage the power of different LLMs [105].

Time Series Foundation Modals. Inspired by the concept of foundation models, such as LLMs, and moving away from training each model per task, some studies have also explored building foundational models for time series. For instance, [29] introduced MOMENT, a family of pretrained time-series foundation models, and collected a new time series benchmark. [68] introduced a unified model trained for cross-domain learning with natural language as domain instructions to provide domain-specific information. In contrast to [77], which adapts pre-training strategies from neutral language processing and computer vision, [72, 19] introduced effective pre-training frameworks for time series forecasting. These efforts aim to construct robust time series foundation models.