TS-CLIP: Time Series Understanding by CLIP

Ziwen Chen[1,2] **Xiaoyuan Zhang**[1,2] **Ming Zhu**[1,2]*
[1]School of EIC, Huazhong University of Science and Technology (HUST)
[2]Hubei Key Laboratory of Smart Internet Technology
{chenziwen, xiaoyuanz, zhuming}@hust.edu.cn

Abstract

Contrastive Language–Image Pre-training (CLIP) has recently demonstrated remarkable success in aligning vision and language. Aligning time series with text leverages the rich semantic cues of language to enhance interpretability and generalization, addressing a largely underexplored area of research. Although applying the CLIP training paradigm to time-series and language pairs is promising, it may result in label collapse due to the sparse semantic annotations and the absence of visual cues in time-series data. To address this, we introduce Time Series CLIP (TS-CLIP), a novel approach that tackles label collapse using a synonym bank mechanism. Synonym bank exploits word analogy phenomena to generate potential synonym embeddings as alignment targets. Specifically, the synonym bank facilitates aligning time series with a word distribution instead of a precise textual description. We conducted extensive zero-shot and few-shot experiments on 128 sub-datasets from the UCR archive. The results show that TS-CLIP achieves state-of-the-art (SOTA) performance in zero-shot settings on 51 datasets. Comprehensive ablation studies and visualization analyzes reveal that TS-CLIP effectively aligns time series with natural language. To the best of our knowledge, this is the first foundational model to achieve general time series and natural language alignment. TS-CLIP introduces a new paradigm for the semantic understanding of time series and opens the possibility of integrating the time series modality into multimodal large models.

1 Introduction

Time series tasks constitute a highly research-intensive field(Sun et al., 2020) with applications across industries(Xu, 2021), healthcare(Sternickel, 2002), meteorology(Bi et al., 2023), and more. As a unique modality, models for time series tasks have evolved from specialized statistical models to general-purpose architectures like RNNs, CNNs, and Transformers. Recently, cross-modal models have excelled in the domains of NLP, image, audio, and video(Radford et al., 2021; Guzhov et al., 2022; Xu et al., 2021). Combining quantitative time-series data with qualitative text (e.g., events and semantics) reveals causal relationships and contextual insights, enabling deeper analysis of complex systems. However, existing methods have seen limited progress due to the absence of foundational models for time-series understanding (Sun et al., 2023), hindering tasks like time series captioning and question answering. This gap motivates our research.

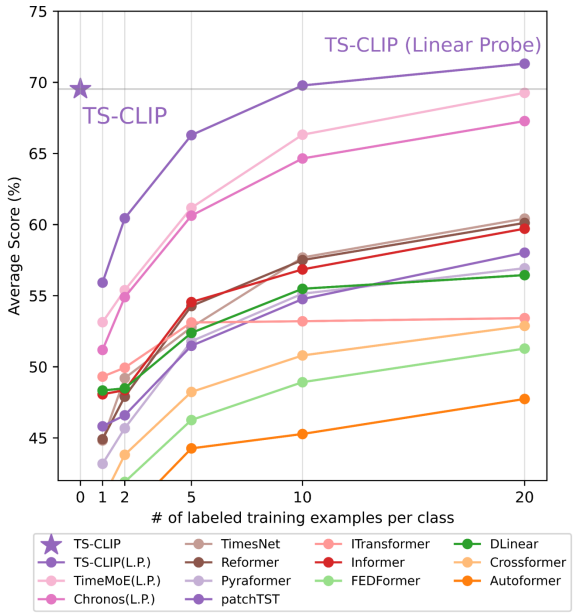

Figure 1: The zero-shot TS-CLIP and its linear probe variants outperform other state-of-the-art models. For a fair comparison, we used the pre-trained TimeMoE with Chronos for the linear probe, while the other baselines utilized fully supervised training.

CLIP aligns text and images in the feature

* Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 4646–4664
November 4-9, 2025

space through large-scale contrastive learning with language-image pairs, demonstrating robust zero-shot capabilities. As a foundational model, CLIP has shown considerable value in downstream tasks such as information retrieval(Luo et al., 2021), data generation(Sanghi et al., 2022; Mirowski et al., 2022; Kwon and Ye, 2022), and 3D modeling(Huang et al., 2023; Wang et al., 2022). Additionally, many studies have emulated CLIP to align supplementary modal data(Guzhov et al., 2022; Zhang et al., 2022). However, despite CLIP's substantial progress in various vision-language and computer vision tasks, it has not yet achieved similar success in time series tasks. Motivated by this, our work seeks to address the issue by investigating the following question: *Is the CLIP architecture effective for the time series modality?*

Time series data, unlike image or audio data, typically lack visual cues and are challenging to annotate with natural language semantics due to their complex features(Sun et al., 2023). Only a small subset of time series data, such as ECG data, includes dense semantic annotations with text descriptions for each segment. Training a CLIP model for time series requires a large number of TS-Text matching pairs, which is challenging to obtain. Currently, several studies aim to achieve alignment between time series data and text or other modalities. Some approaches(Pan et al., 2024; Sun et al., 2023)focus on aligning time series representations with textual prototypes to enable pre-trained language models to perform time series tasks. However, these methods do not align the original semantics of time series segments with natural language representations, limiting their utility for semantic understanding tasks involving time series data. Other approaches(Jin et al., 2023)consider the original semantics of time series data and use general descriptions and terms representing time series trends to reprogram the data. While these methods essentially perform time series forecasting tasks, forcing language models to handle time series prediction may introduce excessive parameters and computational overhead, offering limited contributions to time series modeling. In contrast, understanding the semantic information of time series to extend the modality boundaries of large language models (LLMs) is more meaningful.

We emphasize the importance of paired data in the training process of CLIP. CLIP utilizes 400 million (image, text) pairs to cover as broad a spectrum of visual concepts as possible. Achieving a similar scale for time series datasets is challenging, as a substantial portion of time series data is described in generalized terms. Moreover, unlike images, manual semantic annotation of time series data is inherently difficult. When multiple subclasses are trained using the same major class textual descriptions, CLIP may learn an overly general matching relationship. This excessive generalization can lead to label collapse(Jing et al., 2021), where the model loses sensitivity to the specific characteristics of subclasses, thereby impairing its fine-grained differentiation capabilities. We examine the possibility that, inspired by the evolution from Autoencoder (AE) to Variational Autoencoder (VAE)(Kingma, 2013), time series data might be aligned to a semantic distribution rather than a specific semantic description vector. Thus, we explore the possibility of expanding discrete word vectors into continuous space via word offset. Continuous-space language models exhibit the word analogy phenomenon(Mikolov, 2013), where similarities between word representations extend beyond simple syntactic rules, allowing for algebraic operations on word vectors using word offset techniques(Mikolov et al., 2013). To explore whether CLIP's text encoder has the capability for word analogy, we conducted a toy experiment in Appendix. CLIP demonstrated effective word analogy capabilities on the E-KAR benchmark(Chen et al., 2022).

Consequently, we reassessed the primary challenges of aligning time series with natural language semantics. Time series data typically lack text descriptions, making it difficult to manually annotate with natural language. Based on the observations above, we propose TS-CLIP to address the potential label collapse issue that occurs when aligning time series with natural language. By leveraging the word analogy capabilities of the text encoder, we connected the synonym bank after the text encoder. The synonym bank generates multiple near-synonymous embedding vectors from macro-level category description text embeddings, providing reliable contrastive supervision signals for the time encoder. Pre-trained TS-CLIP can achieve semantic alignment between time series and natural language text in a zero-shot manner. To the best of our knowledge, TS-CLIP is the first general model for aligning time series with natural language. Results on 128 sub-datasets from the UCR archive demonstrate that TS-CLIP achieves state-of-the-art performance in zero-shot scenarios on 51 datasets.

The code will be publicly available at: https://github.com/chenziwenhaoshuai/TS-CLIP

We highlight our contributions as follows:

- To the best of our knowledge, we present the first general method for aligning time series with text. Our proposed synonym bank addresses the difficulties of time series data lacking visual cues and the annotation bottleneck.
- We introduce TS-CLIP, which achieves state-of-the-art performance in zero-shot scenarios on 51 of the 128 sub-datasets in the UCR archive.
- TS-CLIP brings the time series modality to multimodal large language models (MLLMs), enabling the processing and understanding of time series data (e.g., sensor data, financial time series), thus addressing the research gap in aligning time series with natural language modalities.

2 Related Work

The field of Time Series (TS) and Natural Language Processing (NLP) modeling is still in its early stages, with limited research mostly concentrated on time series forecasting. Some approaches(Zhou et al., 2023; Chang et al., 2023; Gao et al., 2024) attempt to model time series using language model frameworks by either designing and training a large language model (LLM) from scratch or fine-tuning an existing one via transfer learning to adapt it to time series tasks such as prediction, imputation, and classification. Benefiting from the well-designed structure of LLMs and the initialization weights derived from large-scale pre-training, these methods often perform well across multiple datasets. However, these approaches frequently lose the capability to process natural language. Other methods(Sun et al., 2023; Jin et al., 2023) retain the language processing abilities of LLMs by adding additional encoders to reprogram time series data and natural language into the same latent space. Also, some methods(Xue and Salim, 2023) treat time series values as words, using prompt learning to adapt LLMs to time series. Further, certain approaches(Li et al., 2024; Chung et al., 2023; Liu et al., 2023a) leverage the dense natural language annotations in the medical field to achieve joint modeling of the true semantics of time series and natural language. Despite these advances, a critical issue remains: these methods rely excessively on using language models directly for time series prediction, without fully capitalizing on the strengths of LLMs in semantic understanding and high-level abstract modeling. Hence, aligning time series data with natural language semantically, and exploiting LLMs' capabilities in semantic comprehension and abstract modeling, can unveil intrinsic patterns of time series data, offering a new perspective and approach for cross-modal data fusion and complex task reasoning.

3 Methods

In this section, we aim to address the potential issue of label collapse that arises when aligning time series with natural language due to a lack of dense textual descriptions. As illustrated in Figure 2, we adopt the structurally validated design paradigm of CLIP and introduce a synonym bank. This innovation generates potential synonym embedding vectors to substitute for dense textual descriptions. We will detail the structural design and training specifics of TS-CLIP in the subsequent sections.

3.1 Encoder

TS Encoder: Given a series of observations across T time steps from multiple domains $\mathbf{X}_{1:T} = (x_1, x_2, \ldots, x_T) \in \mathbb{R}^T$ belonging to category k, our goal is to obtain an embedding vector $Z \in \mathbb{R}^{1\times C}$ of dimension C. We adopt the structural design of TimeMoE(Shi et al., 2024) and apply point-wise tokenization to each input sequence $\mathbf{X}_{1:T}$. Subsequently, we use SwiGLU(Shazeer, 2020)to embed each time series point:

$$h_t^0 = \text{SwiGLU}(x_t) \tag{1}$$

To enhance training stability and support longer sequence lengths, we employ rotary position encoding(Zhang and Sennrich, 2019)and RMSNorm(Su et al., 2024). We replace the feedforward network (FFN) with a Mixture-of-Experts (MoE) layer, where each MoE layer consists of multiple expert networks, each following a standard FFN architecture. As a result, individual time series points can be routed to one or multiple experts based on the top-k selection. The MoE layer is formulated as follows:

$$\text{Mixture}(\mathbf{u}_t^l) = \sum_{i=1}^{N} \left(g_{i,t}\, \text{FFN}_i(\mathbf{u}_t^l) \right), \tag{2}$$

$$\mathbf{g}_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{Topk}(\{s_{j,t} \mid 1 \le j \le N\}, K), \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

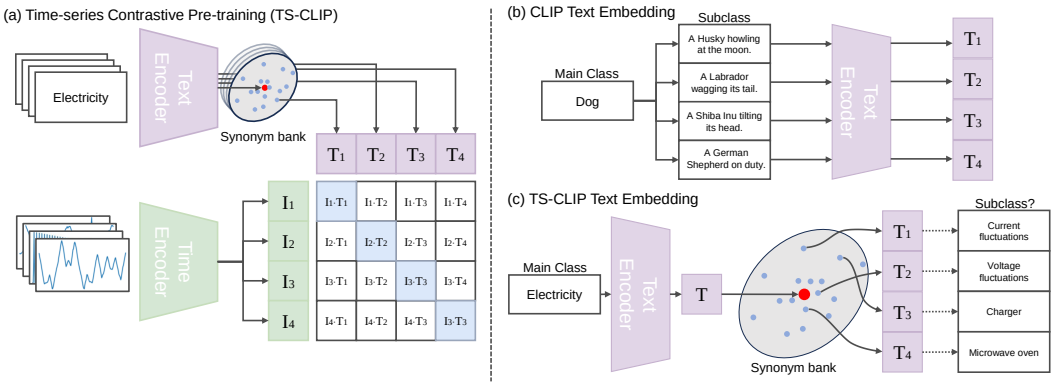

Figure 2: Summary of TS-CLIP: (a) We introduce a synonym bank to enable time series alignment with word distributions, addressing the potential label collapse problem caused by the lack of visual cues. (b) and (c) illustrate how TS-CLIP diverges from CLIP in word embedding.

$$s_{i,t} = \text{Softmax}_i(W_i^l u_t^l), \tag{4}$$

where u_t^l represents the input tensor to the MoE layer, and $W_i^l \in \mathbb{R}^{1\times C}$ denotes the trainable parameters. N and K correspond to the total number of experts and the number of activated experts in each MoE layer, respectively. Finally, we apply averaging to obtain the final embedding vector:

$$Z_\omega = \text{AvgPool}(\mathbf{u}_1^l, \mathbf{u}_2^l, \ldots, \mathbf{u}_t^l) \tag{5}$$

Text Encoder: To ensure effective initialization, we directly incorporate the pre-trained CLIP text encoder, keeping its parameters frozen during the entire training process. This strategy enables the TS-CLIP model's text encoder to leverage the substantial prior knowledge encapsulated within CLIP's corpus, thereby minimizing the necessity for extensive textual data during training. Moreover, freezing the text encoder parameters significantly reduces computational expenses.

During inference, we categorize time series descriptions into two types: fine-grained and coarse category descriptions. Fine-grained descriptions include detailed category names, allowing the use of standard prompt templates to construct supervision signals directly. For coarse category descriptions, we first generate a text embedding from the description text $Z_\omega \in \mathbb{R}^{1\times C}$. Using the category index i as a reference, we then retrieve the corresponding offset vector from the synonym bank $S_i = \text{Synonym}[i] \in \mathbb{R}^{1\times C}$. The final embedding vector is obtained by summing these components, which serves as the supervision signal:

$$Z'_\omega = Z_\omega + S_i \tag{6}$$

The following sections elaborate on the generation of the synonym bank.

3.2 Synonym Bank

The word analogy phenomenon captures a structured relationship between words in natural language processing, where word vectors can perform analogy operations. The most famous example is in continuous-space language model embeddings, where this relationship can be approximated as: $\texttt{vec}_{(\texttt{king})} - \texttt{vec}_{(\texttt{man})} + \texttt{vec}_{(\texttt{woman})} \approx \texttt{vec}_{(\texttt{queen})}$. Building on this concept, we hypothesize that the original word vector in continuous space can obtain its synonym vector through a small offset. Hence, we use vector operations to estimate the synonym distribution for any coarse category description and sample potential synonym embeddings accordingly. We assume that word vectors $Z_\omega \in \mathbb{R}^{1\times C}$ in an embedding space of dimension C follow a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where the mean vector μ represents the central vector of the synonyms for a given word ω, and the covariance matrix σ defines the relationships between different dimensions in the embedding space. For a given word ω, the distribution of its synonyms is formulated as:

$$p(\mathbf{Z}) = \frac{1}{(2\pi)^{\frac{c}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{Z_s}-\mu)^\top \Sigma^{-1}(\mathbf{Z_s}-\mu)\right), \tag{7}$$

where $Z_s \in \mathbb{R}^C$ denotes the embedding vector of a synonym, $\mu \in \mathbb{R}^C$ represents the mean vector derived from ω, and $\Sigma \in \mathbb{R}^{C\times C}$ is the covariance matrix that defines the shape of the synonym distribution. We assume that the embedding vector of the target word ω directly serves as the mean μ:

$$\mu = Z_\omega \tag{8}$$

To simplify computations, we assume that the covariance matrix is diagonal and determined by a

scalar σ^2:

$$\Sigma = \sigma^2 \mathrm{I}, \tag{9}$$

where $\mathrm{I} \in \mathbb{R}^{C \times C}$ is the identity matrix, and σ^2 represents the variance scalar of the distribution. Consequently, the probability density function of the synonym distribution can be simplified as:

$$p(Z) = \frac{1}{(2\pi\sigma^2)^{c/2}} \exp\left(-\frac{\|Z_s - Z_\omega\|^2}{2\sigma^2}\right) \tag{10}$$

To better utilize the geometric properties of the pre-trained text encoder and generate semantically meaningful synonyms, we introduce an anisotropic perturbation term that enhances perturbations along the Z_ω direction:

$$\Sigma = \sigma^2 \left(\mathbf{I} + \alpha \cdot Z_\omega Z_\omega^\top\right), \tag{11}$$

where α is a weighting factor that adjusts the anisotropic perturbation strength. By sampling L times from the above distribution, we construct the synonym bank $Synonym \in \mathbb{R}^{L \times C}$. We provide pseudocode for implementing the synonym bank generation process in the Appendix.

3.3 Dataset

We collected a comprehensive set of time series classification datasets to encompass a variety of time series domains. As illustrated in Table 1, our training data spans several fields, including medical, energy, biological, spectral, gesture, shape, environmental, simulation, and acoustics. For initialization, we employ the TimeMoE-base model, pre-trained on Time-300B, as our time series encoder, and further train it on our dataset. To validate our model, we use the validation sets from 128 sub-datasets within the UCR Time Series Archive(Dau et al., 2019). To ensure fair comparisons and prevent data leakage, we implement a data-cleaning process to eliminate any potential validation samples from the training set. The complete training data will be publicly released.

Prompt Engineering: Constructing prompts from raw data is a critical step in initializing the training pipeline. Once the initial training data is collected, we classify the sub-datasets into two categories:

- Densely annotated datasets – Contain detailed `README` and rich subclass label texts.
- Coarsely annotated datasets – Include only `README` without detailed subclass labels.

The first category comprises 44.6% of the total. For these datasets, we adopt a fixed prompt template: `"Time series of {sub_class}, {class_summary}"`, where `{sub_class}` denotes the specific subclass label, and `{class_summary}` provides a concise dataset description. We use LLM to generate summaries through the `README` of the dataset. For instance, in the AllGestureWiimoteX dataset, we construct the prompt: `"Time series of shake, a subclass of acceleration time series from Wiimote gestures across X axes"`. This prompt is then fed into the text encoder to obtain the corresponding embedding vector. The second category accounts for 55.4% of the total. Since these datasets lack subclass labels and only provide `{class_summary}`, we retrieve an offset vector from a pre-generated synonym bank using the assigned class ID. This offset is then added to the text embedding of `{class_summary}`. To improve efficiency, we precompute all text embeddings prior to training. As the text encoder remains frozen, this approach minimizes computational overhead during training.

3.4 Loss Function

Following the contrastive learning framework, we treat the time series data and its corresponding textual description as positive sample pairs, while all other pairs are designated as negative samples. TS-CLIP is trained by maximizing the contrastive loss for negative pairs and minimizing it for positive pairs. We compute the similarity between two representations using cosine similarity:

$$\mathrm{sim}(ts, t) = \frac{ts^\top \cdot t}{\|ts\|\|t\|} \tag{12}$$

Here, ts denotes the embedding vector of the time series, t is the embedding vector of the text, The terms $\|ts\|$ and $\|t\|$ represent the L2 norms of ts and t, respectively. Next, we calculate the contrastive loss for the i-th TS-to-Text pair as follows:

$$L_i^{(ts \to t)} = -\log\left(\frac{\exp(\mathrm{sim}(t_i, ts_i)/\tau)}{\sum_{k=1}^{N} \exp(\mathrm{sim}(t_i, ts_j)/\tau)}\right), \tag{13}$$

where the initial value of τ is set to 0.07. Similarly, the contrastive loss for Text-to-TS is formulated as:

$$L_i^{(t \to ts)} = -\log\left(\frac{\exp\left(\mathrm{sim}(ts_i, t_i)/\tau\right)}{\sum_{k=1}^{N} \exp\left(\mathrm{sim}(ts_i, t_j)/\tau\right)}\right) \tag{14}$$

	Medical	Energy	Biological	Spectral	Gesture	Shape	Environmental	Simulation	Acoustics	Total
#Seqs.	18,273	27,808	3,663	15,342	32,889	29,062	35,870	16,774	11,477	191,158
#Obs.	10,912,820	8,019,534	4,698,176	13,425,183	18,637,069	10,644,033	2,579,126	4,532,622	6,855,617	80,304,180
%	9.54%	14.54%	1.91%	8.02%	17.20%	15.20%	18.76%	8.77%	6.00%	100.00%

Table 1: Key statistics of the training dataset from various domains.

Finally, the overall training loss is computed as the average of all positive TS-Text pairs within each batch:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \frac{L_i^{(ts \to t)} + L_i^{(t \to ts)}}{2} \tag{15}$$

4 Experiments

4.1 Implementation Details

We utilize TimeMoE-base(Shi et al., 2024) as the time encoder. To leverage prior knowledge, we initialize the pre-trained weights of TimeMoE from Time-300B. Following this, we continued pre-train on our dataset and use the validation sets from the 128 sub-datasets in the UCR Time Series Archive(Dau et al., 2019) for evaluation. A linear projection layer is added to align the output dimensions, with the output dimension set to 512. The temperature parameter τ is initialized to 0.07. The time encoder is optimized using the AdamW optimizer, with an initial learning rate of 1e-5 and a weight decay of 0.01. During the pre-training phase, we train for 200 epochs with a batch size of 192. For the linear probe, we use a batch size of 16 and train for 100 epochs with early stopping and a patience of 10. All experiments are conducted on an NVIDIA Tesla A100 GPU.

4.2 Experimental Results

To evaluate the performance of TS-CLIP, we compare our method against several established baselines: TimeMoE(Shi et al., 2024), Chronos(Ansari et al., 2024), Autoformer(Wu et al., 2021), Crossformer(Zhang and Yan, 2023), Dlinear(Zeng et al., 2023), FEDFormer(Zhou et al., 2022), Informer(Zhou et al., 2021), iTransformer(Liu et al., 2023b), patchTST(Nie et al., 2022), Pyraformer(Liu et al., 2022), Reformer(Kitaev et al., 2020), and TimesNet(Wu et al., 2022). The training protocols for these baselines adhere to the default parameter settings in the Time Series Library[1].

Zero-shot learning refers to a pretrained model's capability to generalize to unseen data during classification. Within the UCR archive, we generated candidate prompts for each of the 128 sub-datasets using predefined templates. We then compared our model with a fully supervised baseline on each sub-dataset. As illustrated in Table 2, TS-CLIP outperformed the baseline in 51 out of the 128 datasets (see the detailed table in the Appendix). This finding underscores TS-CLIP's robust zero-shot learning capabilities and demonstrates the effectiveness of incorporating a synonym Bank to prevent label collapse when aligning time series with natural language modalities. In Figure 3, we analyze the datasets where TS-CLIP performed well and those where it did not perform well. We observed that TS-CLIP typically performs better on datasets with larger volumes. For instance, TS-CLIP excels in datasets related to Energy (FreezerSmallTrain, PLAID, LargeKitchenAppliances). Conversely, its performance is subpar in domains like Biological (InsectEPGRegularTrain) and Simulation (TwoPatterns). Additionally, TS-CLIP shows a preference for datasets with strong regularities, such as Medical (PigArtPressure, PigCVP) and Shape (OSULeaf). However, its performance is limited on datasets with high complexity, irregular patterns, and significant noise, such as those in Environmental (Crop, Chinatown, MelbournePedestrian) and Gesture (CricketZ, InlineSkate). These results suggest that TS-CLIP is effective at leveraging natural language descriptions to align with data characterized by clear structures and well-defined class boundaries. However, its mapping capability is constrained by noisy datasets with high variability.

Few-shot learning provides a more direct metric for assessing TS-CLIP's adaptability to domain-specific data. Unlike zero-shot learning, few-shot learning allows the model to observe a limited number of samples to adapt to a new domain. For time series data, a robust few-shot learning capability can counterbalance TS-CLIP's deficiencies in fine-grained class differentiation. In Figure 1, we compare the performance of TS-CLIP using linear probing against baseline methods. Since both TimeMoE and Chronos are pretrained models, we evaluated them using linear probing, while fully supervised methods were applied to the other, non-

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

Dataset	TS-CLIP	Autoformer	Crossformer	DLinear	FEDFormer	Informer	iTransformer	patchTST	Pyraformer	Reformer	TimesNet
ACSF1	**77.00**	38.00	63.00	42.00	54.00	57.00	42.00	60.00	46.00	55.00	63.00
Adiac	**78.01**	42.20	54.73	70.59	70.08	47.83	29.92	64.71	28.13	38.62	43.22
AllGestureWiimoteX	58.71	21.57	**59.57**	22.00	34.29	49.29	46.29	53.14	45.43	45.29	49.71
AllGestureWiimoteY	**64.57**	25.71	62.43	35.00	38.71	50.00	41.71	61.14	49.71	52.71	55.14
AllGestureWiimoteZ	**54.14**	25.29	49.14	22.57	33.57	40.29	35.29	42.00	28.43	39.00	39.57
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Wafer	**99.94**	99.32	99.40	93.79	99.68	99.63	99.56	99.51	99.51	99.35	99.58
Wine	**74.07**	44.44	44.44	57.41	51.85	55.56	50.00	53.70	51.92	57.41	55.56
WordSynonyms	**62.38**	26.96	61.44	40.75	42.79	54.39	52.66	57.05	44.67	54.86	57.52
WormsTwoClass	54.55	48.05	**61.04**	55.84	54.55	50.65	54.55	50.65	**64.47**	55.84	58.44
Worms	**66.23**	44.16	54.55	38.96	53.25	37.66	48.05	50.65	48.68	38.96	53.25
Yoga	**85.77**	61.37	78.20	64.63	64.50	74.67	69.30	76.67	64.07	72.77	77.27
AVG.	69.65	54.74	71.69	64.02	60.72	70.13	67.31	**72.08**	66.52	68.80	69.90
1^{st} **Count**	**51**	3	23	5	5	12	13	19	12	8	16

Table 2: Performance comparison of Zero-shot TS-CLIP and supervised baselines on UCR datasets. TS-CLIP achieves the best results in 51 out of 128 datasets. All values are accuracy percentages (%), with the highest accuracy in **bold** and the second highest number is underlined.

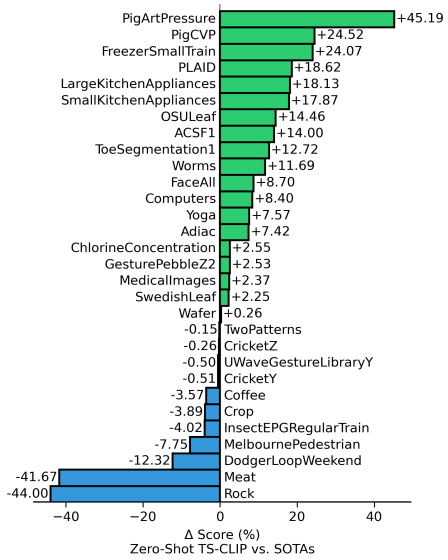

Figure 3: Comparison of zero shot with previous SOTAs on different datasets. TS-CLIP is more biased towards data with strong regularity and distinct category boundaries, and performs weakly on complex datasets.

pretrained models. Our results indicate that the zero-shot performance of TS-CLIP is on par with its 10-shot performance. This consistency may be attributed to the detailed natural language descriptions used to generate zero-shot classifiers, which directly express visual cues, whereas supervised training infers these cues from training samples. Upon using 20 samples, TS-CLIP with linear probing surpassed all baselines. Notably, although TS-CLIP and TimeMoE share the same architecture and initial weights, the incorporation of natural language contrastive learning substantially enhanced the model's few-shot capabilities. We hypothesize that a single time series could encompass multiple concepts, and the inclusion of natural language facilitated the model's understanding of these complex concepts.

5 Ablation Studies

To validate the efficacy of our TS-CLIP design, we conducted a comprehensive ablation study on critical hyperparameters and architectural components across all experimental benchmarks.

Ablation	1^{st}	**Avg. Acc**
TS-CLIP	51	69.65
TS-CLIP (Rewrite)	49	66.84

Table 3: Ablation study of prompt rewrite.

Prompt Rewrite. Label collapse is often observed when a model performs well with a fixed prompt but suffers substantial performance loss upon slight modifications (e.g., altering sentence structure or using synonyms). Therefore, we employed an LLM to rewrite the pre-generated prompts, preserving the original sentence meanings. Table 3 shows our zero-shot test results on TS-CLIP after these rewrites. The results indicate that TS-CLIP did not experience significant performance degradation, demonstrating that TS-CLIP, trained with a synonym bank, possesses robust generalization ability to handle diverse language descriptions.

Ablation	1^{st}	**Avg. Acc**
ID Text Bank	48	69.08%
Synonym Bank	51	69.65%

Table 4: Ablation study of synonym bank generation.

Ablation of synonym bank generation. To validate the sampling process in the Synonym bank, we employed a prompt template using the subclass IDs' text (e.g., "`Type {Sub-CLASS-ID} of time`

series, {CLASS-Summary}") to retrain and test TS-CLIP, referred to as the ID Text bank. Table 4 presents the results comparing both methods, demonstrating that the Synonym bank achieved higher accuracy. We posit that the natural language embeddings of subclass IDs (such as "1", "2") typically display specific semantic distributions within the language model. In models like BERT or word2vec, these numerical embeddings may represent certain semantic features (such as order or category) rather than being random. When these embeddings are combined with the original class embeddings, the resulting synonym distribution might skew towards a particular geometric structure instead of a genuinely diverse distribution. This fixed pattern could restrict the alignment capability between time series features and these distributions. Conversely, the sampling method of the Synonym bank uniformly perturbs text embeddings in multi-dimensional space, thereby more effectively generating a uniform "synonym distribution," which facilitates better alignment of time series with natural language.

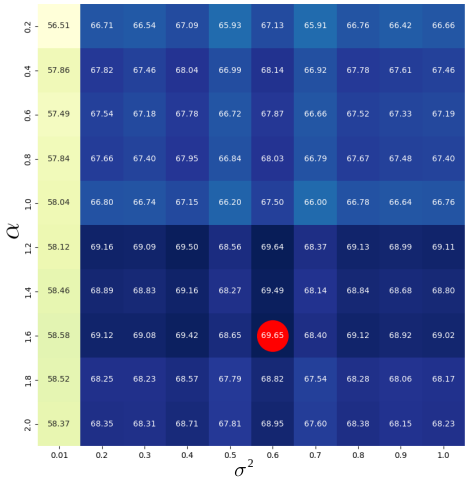

Figure 4: Grid search accuracy percentages (%) for the parameters σ^2 and α in the synonym bank sampling process.

Ablation of σ^2 and α. In the sampling process of the Synonym bank, the parameter σ^2 controls the isotropic perturbation. If σ^2 is too small, it generates indistinguishable synonym vectors, whereas if σ^2 is too large, it introduces semantically unrelated word vectors that are closely positioned in the latent space. To address this issue, we introduced the anisotropic term hyperparameter α , which adds controllable perturbations in the semantic direction, thereby generating more reasonable synonyms. Figure 4 illustrates the grid search results for σ^2 and α. As shown, when σ^2 is set to 0.6 and α is set to 1.6, TS-CLIP achieves its highest performance.

Prompt	Zero-Shot
"a time series of a [Sub-CLASS]."	68.41
"Time series representation of a [Sub-CLASS]."	67.63
"A dynamic time series of [Sub-CLASS]."	68.68
"Time series of [Sub-CLASS], [CLASS-Summary]."	69.65

Table 5: Ablation study of prompt design.

Ablation of Prompt Design. Table 5 presents our experiments with four different prompt designs. We found that employing solely subclass descriptions as prompts did not fully exploit the capabilities of TS-CLIP. However, when we included a summary of the dataset, TS-CLIP achieved optimal performance. This indicates that certain datasets may contain similar subclass descriptors belonging to different major categories. For example, "mobile phone" can appear in both "Electricity" and "Shape" datasets. Incorporating the dataset summary allows the model to better contextualize the current time series, leading to more accurate classification of sample types.

Downstream Tasks. TS-CLIP is pre-trained on the time-series-to-text retrieval task. Verifying whether TS-CLIP can effectively transfer to downstream tasks serves as an important sanity check and proof of concept. Therefore, we report the experimental results of TS-CLIP on downstream tasks related to time-series understanding in the Appendix, providing insights and inspiration for future research.

6 Conclusions

In this work, we propose TS-CLIP, which, to the best of our knowledge, is the first foundational model for aligning general time series with natural language. By introducing the synonym bank mechanism, TS-CLIP addresses the issue of label collapse caused by the lack of dense textual descriptions in time series data. Experimental results show that TS-CLIP exhibits strong zero-shot learning capabilities and few-shot generalization. However, we wish to emphasize that TS-CLIP currently achieves coarse-grained alignment and may have limitations when handling samples requiring precise textual descriptions. Despite this, TS-CLIP provides a new paradigm for the semantic understanding of time series data and cross-modal learning, laying the groundwork for multimodal large language models to process time series modalities.

Limitations

TS-CLIP lays the foundation for aligning time series with natural language. However, efficiently integrating alignment models into large language models (LLMs) to support complex multimodal reasoning requires further research. A foreseeable limitation is that, in real-world scenarios, time series from different sources may exhibit highly similar waveform characteristics; for instance, electrical current and voltage signals of appliances often appear as sine waves. This phenomenon might lead to semantic conflicts during the alignment of time series with natural language, where time series with different semantics could be erroneously mapped to similar or identical linguistic expressions, impairing semantic understanding and inference accuracy in downstream tasks. To address this issue, employing methods such as prompt learning or one-shot learning when integrating with LLMs may be necessary. These methods involve initially teaching the model based on exemplary samples. However, this solution heavily relies on the availability of high-quality samples and increases the complexity of both model design and usage, thereby limiting its generalizability in certain scenarios. Secondly, constructing a pre-training dataset in the time series domain that is comparable in scale to CLIP's (400 million samples) is challenging, making efficient data utilization a current limitation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China, grant number 62201220.

References

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. 2023. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538.

Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. 2023. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469*.

Jiangjie Chen, Rui Xu, Ziquan Fu, Wei Shi, Zhongqiao Li, Xinbo Zhang, Changzhi Sun, Lei Li, Yanghua Xiao, and Hao Zhou. 2022. E-kar: A benchmark for rationalizing natural language analogical reasoning. *arXiv preprint arXiv:2203.08480*.

Hyunseung Chung, Jiho Kim, Joon-myoung Kwon, Ki-Hyun Jeon, Min Sung Lee, and Edward Choi. 2023. Text-to-ecg: 12-lead electrocardiogram synthesis conditioned on clinical text reports. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2019. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2021. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. 2024. Units: Building a unified time series model. *arXiv preprint arXiv:2403.00131*.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Zero-shot detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2(3):4.

Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. 2022. 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.

Mariya Hendriksen, Maurits Bleeker, Svitlana Vakulenko, Nanne Van Noord, Ernst Kuiper, and Maarten De Rijke. 2022. Extending clip for category-to-image retrieval in e-commerce. In *European Conference on Information Retrieval*, pages 289–303. Springer.

Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. 2023. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2021. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*.

Diederik P Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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

Gihyun Kwon and Jong Chul Ye. 2022. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18062–18071.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7331–7341.

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

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

Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. 2023a. Large language models are few-shot health learners. *arXiv preprint arXiv:2305.15525*.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023b. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502.

Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

Piotr Mirowski, Dylan Banarse, Mateusz Malinowski, Simon Osindero, and Chrisantha Fernando. 2022. Clip-clop: Clip-guided collage and photomontage. *arXiv preprint arXiv:2205.03146*.

Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

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

David Nukrai, Ron Mokady, and Amir Globerson. 2022. Text-only training for image captioning using noise-injected clip. *arXiv preprint arXiv:2211.00575*.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. 2024. s^2 ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *Forty-first International Conference on Machine Learning*.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2085–2094.

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

Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. 2022. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18603–18613.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. 2024. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*.

Karsten Sternickel. 2002. Automatic pattern recognition in ecg time series. *Computer methods and programs in biomedicine*, 68(2):109–115.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Chenxi Sun, Shenda Hong, Moxian Song, and Hongyan Li. 2020. A review of deep learning methods for irregularly sampled medical time series data. *arXiv preprint arXiv:2010.12493*.

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

Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. 2022. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844.

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

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

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*.

Jiehui Xu. 2021. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*.

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

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

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

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

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

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

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

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

A A Revisit of CLIP

CLIP(Radford et al., 2021)is a foundational method known for its robust scalability, designed to learn extensive visual concepts through natural language supervision. The key contribution of CLIP is its ability to align image and text representations in a zero-shot manner(Gu et al., 2021), enabling various vision-language downstream tasks. CLIP's training objective is to link image representations to text representations, thus facilitating tasks such as information retrieval and image captioning(Hendriksen et al., 2022; Nukrai et al., 2022; Mokady et al., 2021). Additionally, several works(Luo et al., 2021; Lei et al., 2021) have extended its retrieval capabilities to video retrieval. Controllable image generation tasks also benefit from natural language guidance; various studies(Wang et al., 2022; Patashnik et al., 2021; Sanghi et al., 2022; Michel et al., 2022) have used CLIP-aligned vision-language representations to guide 2D or 3D generation tasks. Beyond the visual domain, certain works have explored aligning other modalities (such as audio, point clouds, ECG series) with language(Zhang et al., 2022; Guzhov et al., 2022; Li et al., 2024). These alignment tasks are typically feasible due to the availability of dense natural language identifiers for corresponding modalities, obtained via automated tools or human effort. However, achieving such alignment for time series data remains challenging.

CLIP is trained using a contrastive learning framework, which learns cross-modal representations by maximizing the similarity of correct image-text pairs while minimizing the similarity of incorrect ones. Specifically, CLIP employs a dual-encoder structure (image encoder and text encoder) to project images and text into the same vector space, optimized using the InfoNCE (Noise Contrastive Estimation) loss. For a batch of (N) image-text pairs, the model computes an $(N \times N)$ similarity matrix and uses softmax normalization to calculate the cross-entropy loss, ensuring that the correct image-text pairs have the highest similarity. Through this method, CLIP is pretrained on large-scale image-text data, learning general cross-modal representations that enable it to perform open-class recognition tasks in a zero-shot setting during inference.

Within CLIP, two independent encoders generate embedding vectors for images and text. During training, CLIP employs a contrastive loss to align images and texts within a given batch in the embedding space.

Method	Acc.
Word2Vec	25.6
GloVe	27.8
FastText	28.2
BERT-Base	30.4
CLIP	30.77

Table 6: Performance comparison of different methods on word analogy tasks.

For a given set of K unknown categories, CLIP places all category names into predefined prompt templates. The text encoder then generates C-dimensional text embeddings $Z_t \in \mathbb{R}^{K \times C}$, where K row vectors encode the pre-trained category representations. For an input test image, the image encoder generates the corresponding image embeddings $Z_i \in \mathbb{R}^{1 \times C}$. Classification $\text{logits}_i \in \mathbb{R}^{1 \times K}$ is performed by calculating the predicted probabilities of the K categories using a softmax operation. The process can be formulated as follows:

$$p = \text{SoftMax}(Z_i Z_t^\top) \tag{16}$$

In this procedure, no new training images are required. The pre-trained encoders alone enable robust zero-shot performance.

To explore whether CLIP's text encoder has the capability for word analogy, we conducted a toy experiment. We compared CLIP's text encoder with three static word embedding methods—word2vec, GloVe, and FastText—which have been proven effective for word analogy tasks, and with context-aware embeddings from pre-trained language models such as BERT. As shown in Table 6, CLIP demonstrated effective word analogy capabilities on the E-KAR benchmark(Chen et al., 2022).

B Visualization

To evaluate the alignment effectiveness of TS-CLIP, we performed an intuitive visualization of its embeddings. We selected 14 datasets, embedding their class summaries with TS-CLIP's text encoder and several time series samples with the time encoder. Figure 6 displays the t-SNE visualization results. We observed that the representations of time series samples (denoted as points) closely aligned with the corresponding text representations (denoted as stars), illustrating the cross-modal alignment capability of TS-CLIP.

To visually demonstrate the sampling effectiveness of the synonym bank, we employed the Clip-

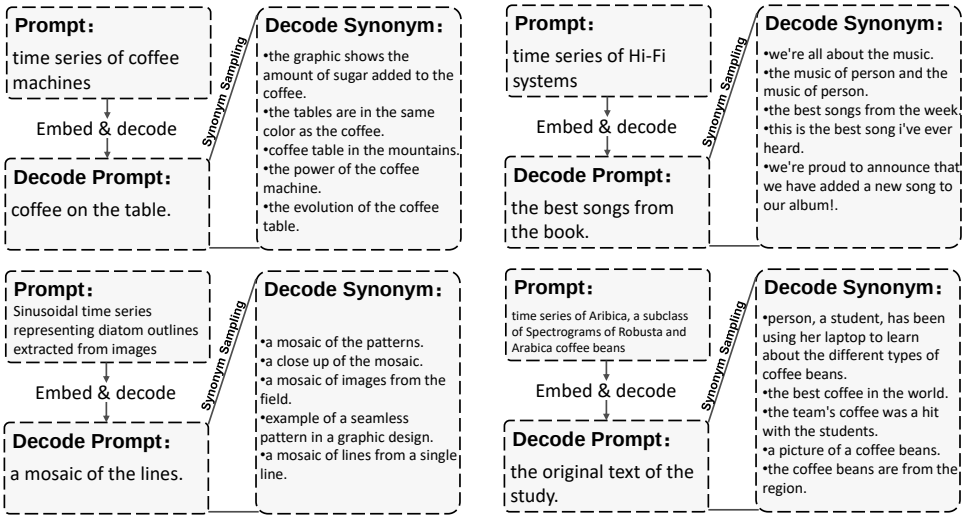

Figure 5: Decoded results of the original prompt and sampled embeddings from the synonym bank.

Cap(Mokady et al., 2021) decoder to decode five samples from the synonym bank. This approach indirectly showcased the potential distribution of the aligned texts. Figure 5 displays the original prompt, the decoded result of the original prompt embedding vector, and the decoded results of the sampled embedding vectors from the synonym bank. Notably, while ClipCap cannot achieve perfectly accurate vector decoding, the synonym bank preserves the essential hints from the original prompt and exhibits a degree of generalization.

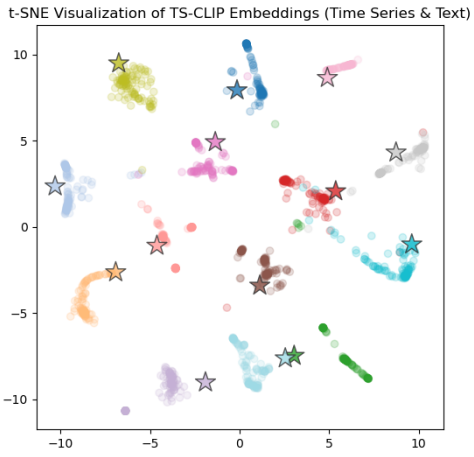

Figure 6: t-SNE visualization of TS-CLIP embeddings for 14 datasets. Time series samples (points) and their corresponding class summaries (stars) are closely aligned.

Since we directly used CLIP's text encoder during training without updating its parameters, TS-CLIP aligns time series with natural language and inherently retains the ability to align with images. In Figure 7, we utilized CLIP's image encoder to process images as retrieval targets. It can be observed that the embedding derived from a spectrum sequence of beef naturally exhibits higher similarity with the image embedding of beef. Similarly, a car-related time series displays higher similarity with the image of an SUV. This demonstrates that TS-CLIP not only establishes effective alignment between text and time series but also enables efficient matching between time series and images via the existing CLIP model.

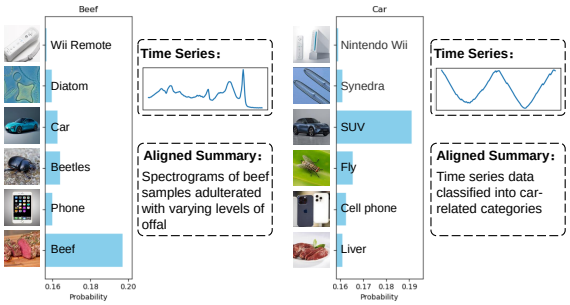

Figure 7: TS-CLIP aligns time series with images. Aligned Summary refers to the raw text of the time series alignment during training.

C Downstream Tasks

TS-CLIP is pre-trained for the task of time series-text retrieval. Although the focus of this paper is on representation learning and pre-trained models, verifying whether TS-CLIP can be transferred to downstream tasks serves as an important sanity check and proof of concept. Therefore, we evaluate TS-CLIP on downstream tasks related to time

series understanding, providing insights for future research.

C.1 Time Series Caption

Time series captioning is a fundamental task in time series language understanding, where the model predicts textual captions for a given input time series. While pre-trained foundation models have achieved significant progress in vision-language captioning, there has been limited work reported on time series captioning due to the lack of foundational models similar to CLIP. To demonstrate the transferability of TS-CLIP in downstream tasks, we adapt the ClipCap framework (Mokady et al., 2021) with minor modifications to implement time series captioning. Leveraging the inherent alignment of TS-CLIP with image-based representations, we replace the image encoder in ClipCap with the time series encoder from TS-CLIP to enable caption generation for time series data. Figure 9 illustrate the captioning results of TS-CLIP for different time series inputs. As shown, TS-CLIP effectively extracts semantic information embedded in various time series. However, since no modules were retrained in our experiments and ClipCap was originally trained on visual data, the generated captions tend to describe sequences as visual scenes. Despite this limitation, the results provide strong evidence for TS-CLIP's capability as a foundational model in enabling time series captioning tasks.

C.2 Time Series Question Answering

Time Series Question Answering (TQA) is a fundamental task in downstream applications of time series understanding models, where the model is required to answer a question related to the input time series. Among various forms of TQA, multiple-choice TQA is particularly challenging, as it requires the model to distinguish between multiple seemingly plausible answers. To demonstrate the transferability of TS-CLIP as a foundational model to TQA tasks, we designed a pipeline for implementing TQA, as illustrated in Figure 8.

In this pipeline, each option for the input question is split and combined with the question to form multiple textual segments. These segments are then paired with the input time series to compute cosine similarity scores. The computed results are aggregated as auxiliary information, which, along with the original question, is formatted using prompt templates and fed into a large language model (LLM), enabling TQA without requiring additional training. Figure 10 present TS-CLIP's responses to different input time series and question pairs. The results demonstrate that TS-CLIP's zero-shot inference capability effectively supports the implementation of TQA tasks.

D More ablation studies and experiments

D.1 Initialization methods

We compared the impact of using TimeMoE pretraining weights on the performance of TS-CLIP. Table 7 presents a comparison of results when initializing TS-CLIP with TimeMoE pretraining weights versus training without pretraining, under the same number of epochs. It is evident that, under identical training conditions, TS-CLIP with random initialization performs worse. By increasing the number of pretraining epochs, the randomly initialized TS-CLIP achieves comparable performance. We also experimented with using a vanilla Transformer with a similar parameter size as the time series encoder. The results indicate that the performance of the Transformer-based encoder is comparable to that of TimeMoE. However, the mixture-of-experts mechanism in TimeMoE significantly reduces computational overhead during inference, which motivates our choice of TimeMoE as the time series encoder for TS-CLIP.

Initialization Methods	Zero-Shot
TimeMoE-Pretrain-200epoch	69.53
TimeMoE-Random-200epoch	52.79
TimeMoE-Random-400epoch	69.51
Transformer-Random-400epoch	69.47

Table 7: Ablation study of initialization methods.

D.2 Different LLMs Used for Summary

When creating the dataset, we utilized large language models (LLMs) to generate summaries. For the selection of LLMs, we conducted ablation experiments, as shown in Table 8. The results indicate that differences in the summarization capabilities of current advanced LLMs do not significantly affect the training process of TS-CLIP. This is because the embedding vectors generated during text embedding are sampled from the summaries produced by LLMs within the synonym bank. As a result, the synonym distributions generated after embedding are similar across different LLMs. Therefore, all existing advanced LLMs are suitable for use in the TS-CLIP training process.

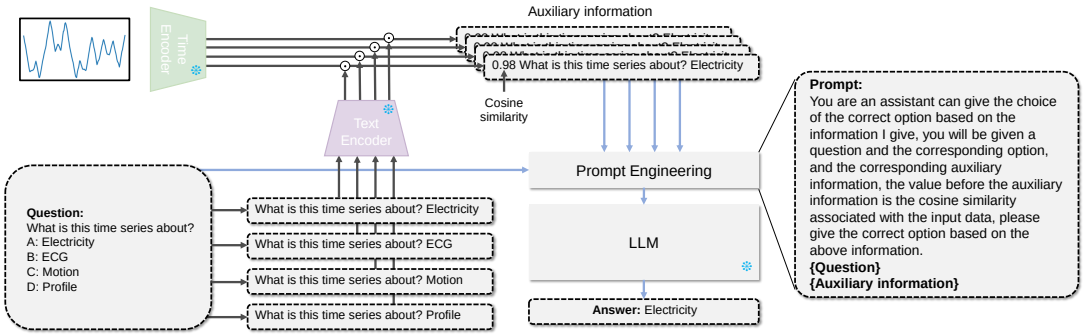

Figure 8: Overview of the Time Series Question Answering pipeline.

LLMs	Zero-Shot
ChatGPT-4o	69.53
ChatGPT-4o mini	69.51
Qwen 2.5	69.50
Claude 3.5 sonnet	69.52
Claude 3.5 Haiku	69.52

Table 8: Ablation study of LLMs.

D.3 Comparison of Synonym Bank and ID Text Bank

In the previous sections, we compared the performance of TS-CLIP trained with the ID text bank and observed similar results in linear probe experiments. However, does this imply that the Synonym bank can be replaced? To address this question, we conducted further experiments. Table 9 illustrates the prompt words generated by the two methods for the description "A power usage curve of a household appliance." As shown, the Synonym bank generates prompt words that are semantically relevant to the input description, whereas the ID text bank merely distinguishes between different categories without providing meaningful semantic differentiation. For linear probe experiments, distinguishing between different embedding vectors is relatively straightforward, which explains why the linear probe results showed no significant difference. However, the ID text bank is not meaningful for understanding time series data. To further validate this, we conducted zero-shot classification tests on several datasets. As shown in Table 10, TS-CLIP trained with the ID text bank predicts nearly equal probabilities for each category, rendering it incapable of performing zero-shot classification and limiting its applicability to downstream tasks. In contrast, TS-CLIP trained with the Synonym bank successfully performs zero-shot classification, demonstrating its effectiveness in downstream applications.

CLASS ID	Synonym Bank	ID Text Bank
1	Time series of **TV**, A power usage curve of a household appliance.	Time series of **class 1**, A power usage curve of a household appliance.
2	Time series of **phone**, A power usage curve of a household appliance.	Time series of **class 2**, A power usage curve of a household appliance.
3	Time series of **computer**, A power usage curve of a household appliance.	Time series of **class 3**, A power usage curve of a household appliance.
4	Time series of **oven**, A power usage curve of a household appliance.	Time series of **class 4**, A power usage curve of a household appliance.

Table 9: Ablation study of the prompt words generated by the synonym bank and ID text bank for the description "A power usage curve of a household appliance."

D.4 Few-Shot Fine-Tuning Performance Analysis

In the zero-shot experimental results, TS-CLIP achieves the best performance on 51 out of 128 datasets. However, its average accuracy is lower than that of the supervised baseline. TS-CLIP does not require any post-training after pre-training, allowing us to evaluate its performance using a zero-shot approach in comparison to fully supervised baseline models. While the zero-shot method achieves competitive performance without relying on extensive labeled data, it inherently sacrifices

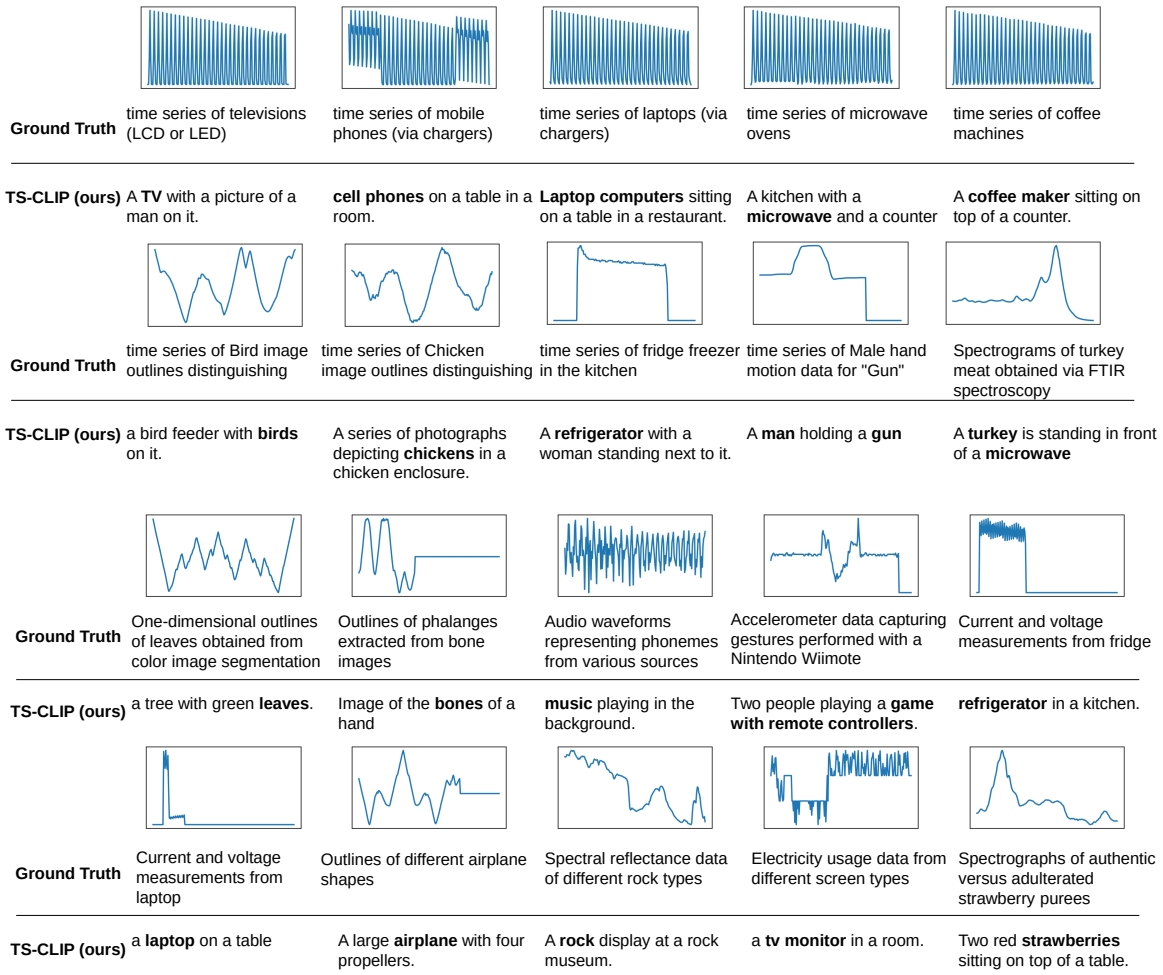

Figure 9: The visualization of TS-CLIP's time series captioning results for different input sequences.

Dataset(num_class)	Synonym Bank	ID Text Bank
ACSF1 (10)	77.00	12.00
Adiac (37)	78.01	3.42
CBF (3)	98.00	34.25
ChlorineConcentration (3)	72.24	31.55
FaceAll (14)	90.41	6.92
HandOutlines (2)	94.32	54.62
LargeKitchenAppliances (3)	84.27	37.33
MiddlePhalanxOutlineCorrect (2)	82.82	51.89
ProximalPhalanxOutlineCorrect (2)	86.69	42.80

Table 10: Ablation study of synonym bank and ID text bank in zero-shot learning.

some accuracy in exchange for enhanced generalization and application flexibility. Therefore, although the average accuracy is lower than that of fully supervised models, this outcome aligns with the anticipated trade-off in performance. In Table 11, we report the performance of TS-CLIP under few-shot fine-tuning. It can be observed that TS-CLIP surpasses all supervised baselines with as few as 20 samples for fine-tuning. When trained with the full dataset, TS-CLIP significantly outperforms the previous supervised baselines.

Training Setting	Accuracy
Previous Supervised SOTA (patchTST)	72.08
1-shot (TS-CLIP)	55.92
2-shot (TS-CLIP)	60.44
5-shot (TS-CLIP)	66.29
10-shot (TS-CLIP)	69.78
20-shot (TS-CLIP)	72.31
Fully Supervised (TS-CLIP)	**76.52**

Table 11: Ablation study of few-shot and fully supervised fine-tuning.

D.5 Comparison with Additional Methods

We compared zero-shot and few-shot performance with several self-supervised methods and other alignment approaches. Specifically, we selected self-supervised baselines such as PatchTST (Nie et al., 2022), TS-TCC (Eldele et al., 2021), and GPT4TS (Zhou et al., 2023) for comparison. Additionally, we compared TS-CLIP with other time-

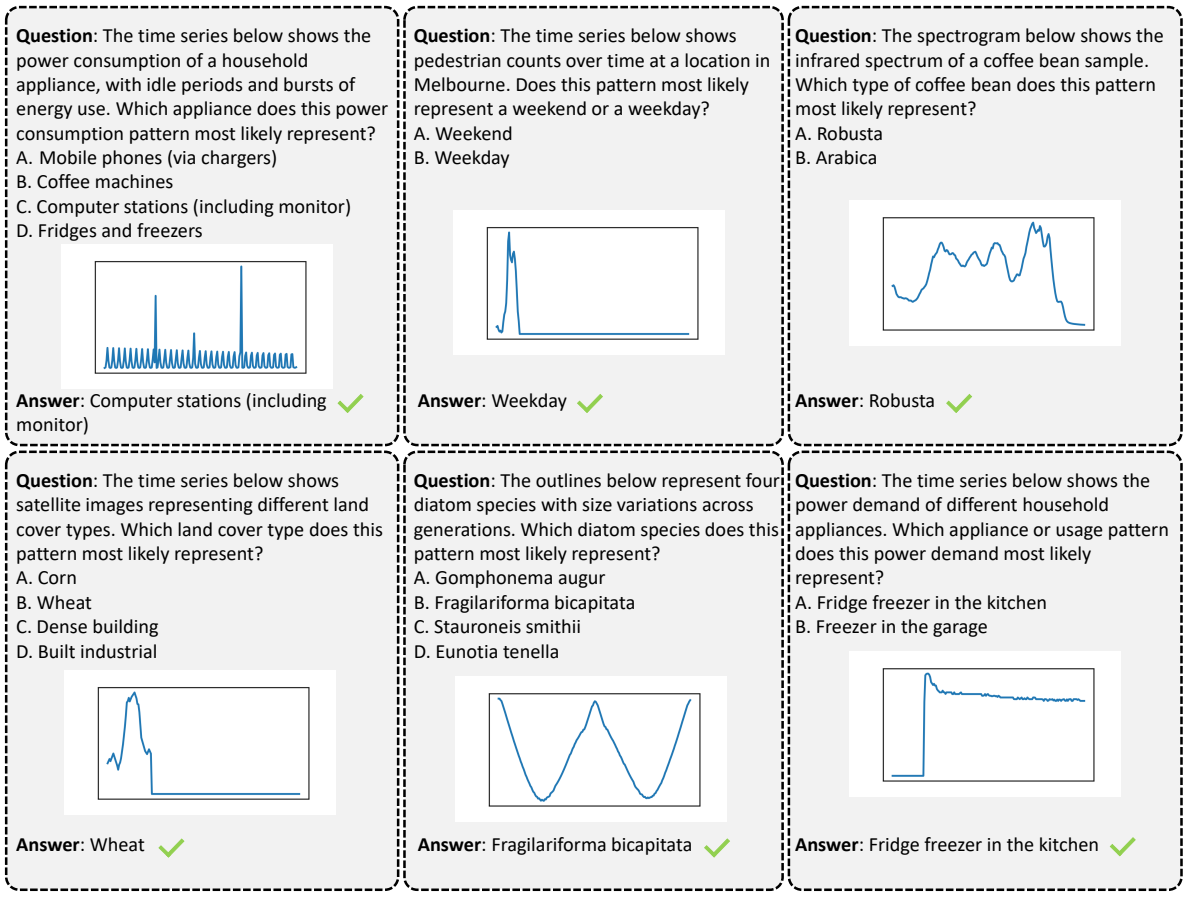

Figure 10: The visualization of TS-CLIP's time series question answering results for different input sequences.

series and natural language alignment methods, including TimeLLM (Jin et al., 2023) and TEST (Sun et al., 2023). As shown in Table 12, none of the aforementioned baselines can perform zero-shot classification, as they do not fundamentally align the semantics of time series with natural language. In contrast, the 20-shot fine-tuning results highlight the effectiveness of TS-CLIP, further demonstrating its capability in bridging the gap between time-series data and natural language understanding.

Methods	Zero-shot	20-shot
patchTST	-	58.02
TS-TCC	-	61.47
GPT4TS	-	62.27
TimeLLM	-	69.72
TEST	-	67.54
TS-CLIP(ours)	69.53	72.31

Table 12: Ablation study of additional methods.

D.6 Experimental details

In Table 13 and Table 14, we show the complete experimental results of zero shot TS-CLIP on 128 datasets from UCR. Figure 11 shows the code for pytorch like generated by synonym bank.

```
# z_w (Tensor): Base embedding vector (d-dimensional)
# sigma (float): Global perturbation magnitude
# alpha (float): Anisotropic perturbation weight
# K (int): Number of synonyms to generate

d = z_w.size(-1)
# Optional: Normalization
z_w = z_w / z_w.norm(dim=-1, keepdim=True)

# Isotropic noise (K x d)
epsilon_iso = torch.randn(K, d) * sigma

# Anisotropic noise (K x 1) -> (K x d)
z = torch.randn(K, 1)

# Scalar noise
epsilon_aniso = (sigma * torch.sqrt(alpha)) * z * z_w

# Combined perturbation
epsilon = epsilon_iso + epsilon_aniso

# Generate synonym embeddings
synonym = z_w + epsilon
```

Figure 11: Pytorch-like pseudocode for the core of an implementation of synonym bank.

Dataset	TS-CLIP	Autoformer	Crossformer	Dlinear	FEDFormer	Informer	iTransformer	patchTST	Pyraformer	Reformer	TimesNet
ACSF1	**77.00**	38.00	63.00	42.00	54.00	57.00	42.00	60.00	46.00	55.00	63.00
Adiac	**78.01**	42.20	54.73	70.59	70.08	47.83	29.92	64.71	28.13	38.62	43.22
AllGestureWiimoteX	58.71	21.57	**59.57**	22.00	34.29	49.29	46.29	53.14	45.43	45.29	49.71
AllGestureWiimoteY	**64.57**	25.71	62.43	35.00	38.71	50.00	41.71	61.14	49.71	52.71	55.14
AllGestureWiimoteZ	**54.14**	25.29	49.14	22.57	33.57	40.29	35.29	42.00	28.43	39.00	39.57
ArrowHead	61.14	56.00	55.43	69.14	73.71	72.00	68.00	**77.14**	69.14	68.57	57.14
Beef	20.00	56.67	63.33	63.33	46.67	**73.33**	70.00	**73.33**	63.33	66.67	60.00
BeetleFly	75.00	55.00	80.00	80.00	40.00	80.00	**85.00**	**85.00**	80.00	75.00	70.00
BirdChicken	80.00	65.00	55.00	75.00	60.00	55.00	80.00	**90.00**	55.00	60.00	50.00
BME	88.67	85.33	96.67	84.00	96.00	90.00	95.33	94.67	**97.33**	91.33	64.00
Car	85.00	33.33	66.67	76.67	43.33	75.00	81.67	**86.67**	65.00	75.00	76.67
CBF	**98.00**	42.56	87.44	78.11	52.22	93.78	92.89	87.00	95.44	93.11	83.44
Chinatown	30.61	97.08	**98.54**	81.63	96.79	97.38	97.08	90.96	96.50	77.26	59.18
ChlorineConcentration	**72.24**	69.69	57.84	57.94	64.87	63.07	53.10	59.48	55.31	64.95	61.64
CinCECGTorso	66.96	42.03	89.93	45.14	60.07	84.20	80.22	90.36	**90.72**	85.00	84.42
Coffee	96.43	85.71	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	89.29	92.86	85.71
Computers	**72.00**	59.20	60.00	51.20	52.00	53.20	56.00	59.60	63.20	61.20	63.60
CricketX	**65.13**	21.03	63.59	23.33	31.03	54.36	41.28	60.26	42.05	52.31	61.28
CricketY	62.31	14.36	61.54	31.54	36.92	55.90	43.33	**62.82**	42.05	45.64	60.00
CricketZ	69.74	19.49	**70.00**	24.10	36.41	47.44	45.38	63.33	45.64	50.00	61.03
Crop	71.96	71.54	73.66	68.33	74.32	75.24	70.00	73.98	65.29	74.82	**75.85**
DiatomSizeReduction	89.54	43.14	82.35	39.54	61.76	93.46	93.14	**95.10**	86.27	90.85	58.82
DistalPhalanxOutlineAgeGroup	55.40	**77.70**	74.10	66.19	71.22	69.78	74.10	76.26	66.91	74.82	70.50
DistalPhalanxOutlineCorrect	61.59	**78.99**	74.64	68.12	74.64	75.36	61.23	74.28	65.58	72.10	76.81
DistalPhalanxTW	51.80	67.63	67.63	70.50	59.71	69.78	68.35	69.06	69.06	61.15	**71.22**
DodgerLoopDay	18.75	30.00	**56.25**	47.50	40.00	52.50	53.75	55.00	55.00	52.50	**56.25**
DodgerLoopGame	52.17	55.80	84.06	81.16	64.49	81.88	**86.96**	84.78	84.78	84.06	76.81
DodgerLoopWeekend	85.51	75.36	**97.83**	96.38	81.88	**97.83**	**97.83**	**97.83**	**97.83**	**97.83**	**97.83**
Earthquakes	74.82	74.82	74.10	58.99	69.78	72.66	64.03	72.66	74.82	71.22	**76.26**
ECG200	87.00	90.00	89.00	83.00	**92.00**	85.00	84.00	89.00	85.00	84.00	84.00
ECG5000	93.93	94.07	**94.64**	92.87	93.53	93.00	92.64	93.07	92.64	92.53	93.60
ECGFiveDays	96.98	97.44	97.44	57.72	**97.79**	91.99	93.38	82.46	73.95	84.55	75.84
ElectricDevices	**75.06**	65.48	59.28	47.75	62.56	68.64	57.62	63.01	64.88	71.04	64.41
EOGHorizontalSignal	15.19	16.30	**48.90**	41.44	20.99	43.92	33.98	40.06	36.19	42.27	45.58
EOGVerticalSignal	9.94	10.77	41.71	12.43	31.22	41.71	22.10	**46.13**	30.94	30.94	37.85
EthanolLevel	50.00	35.60	25.40	37.00	28.20	**62.00**	24.80	30.20	28.20	58.40	56.80
FaceAll	**90.41**	75.09	79.70	81.72	78.46	74.79	81.01	76.69	74.97	74.02	74.56
FaceFour	67.05	65.91	85.23	80.68	71.59	81.82	84.09	**88.64**	85.23	79.55	81.82
FacesUCR	16.88	77.07	**89.76**	76.24	79.37	84.88	77.12	88.68	83.12	84.24	85.22
FiftyWords	70.99	24.40	**74.29**	49.23	49.01	58.24	60.88	68.13	51.65	58.24	69.01
Fish	36.00	34.29	80.00	84.00	72.00	78.86	80.57	**86.29**	72.00	76.00	81.71
FordA	**93.64**	79.32	86.05	48.86	86.59	63.41	54.95	90.63	51.59	56.78	92.99
FordB	**80.99**	64.94	70.99	50.25	68.77	54.57	52.96	63.95	57.04	69.75	77.28
FreezerRegularTrain	**99.72**	88.74	99.44	92.81	96.00	98.91	90.25	98.84	97.82	95.44	92.07
FreezerSmallTrain	**99.72**	60.42	67.44	69.82	70.53	70.21	70.18	67.89	67.44	69.72	75.65
Fungi	30.65	74.19	88.71	66.13	78.49	95.70	82.80	87.63	84.41	**96.77**	93.01
GestureMidAirD1	46.92	30.00	63.08	56.92	35.38	61.54	49.23	53.85	56.92	60.77	**63.85**
GestureMidAirD2	48.46	20.00	**54.62**	51.54	33.85	44.62	52.31	50.77	41.54	44.62	53.08
GestureMidAirD3	23.08	19.23	23.08	17.69	20.77	7.69	23.08	**37.69**	9.23	13.85	10.00
GesturePebbleZ1	81.98	32.56	**88.95**	76.16	56.98	87.79	72.67	86.63	80.81	84.30	86.63
GesturePebbleZ2	**80.38**	25.32	77.85	60.13	53.16	76.58	66.46	75.32	75.95	70.25	70.89
GunPointAgeSpan	**98.67**	67.41	96.52	86.39	80.70	92.41	87.97	94.62	81.33	88.29	92.41
GunPointMaleVersusFemale	87.66	76.90	98.73	94.94	55.38	99.68	98.73	97.47	**100.00**	99.37	95.89
GunPointOldVersusYoung	93.35	58.73	**100.00**	**100.00**	94.29	**100.00**	**100.00**	92.38	**100.00**	**100.00**	**100.00**
GunPoint	62.86	77.33	89.33	78.00	81.33	**97.33**	94.00	89.33	73.33	94.00	76.00
Ham	68.57	76.19	76.19	64.76	70.48	70.48	**77.14**	68.57	68.27	71.43	**77.14**
HandOutlines	**94.32**	55.95	88.92	88.92	65.41	87.03	89.19	90.00	88.11	88.92	85.68
Haptics	**51.30**	25.00	38.31	45.45	28.57	40.91	43.51	47.40	29.22	36.69	40.26
Herring	60.94	56.25	59.38	**64.06**	56.25	51.56	62.50	57.81	53.13	48.44	62.50
HouseTwenty	**94.12**	63.87	83.19	75.63	68.07	85.71	75.63	82.35	82.35	85.71	87.39
InlineSkate	16.36	19.45	25.64	25.27	16.91	24.55	26.55	27.45	22.91	27.64	**30.73**
InsectEPGRegularTrain	95.98	50.20	**100.00**	**100.00**	47.39	**100.00**	**100.00**	77.91	**100.00**	**100.00**	**100.00**
InsectEPGSmallTrain	95.98	59.44	**100.00**	86.25	49.17	**100.00**	**100.00**	71.67	**100.00**	**100.00**	**100.00**
InsectWingbeatSound	47.17	48.59	61.26	57.53	53.18	59.24	59.49	57.63	55.96	60.15	**62.93**
ItalyPowerDemand	93.00	96.79	**97.08**	95.72	95.72	95.53	96.02	96.31	96.21	95.53	96.02
LargeKitchenAppliances	**84.27**	52.53	61.33	37.07	47.47	49.60	46.67	66.13	51.47	53.07	64.00
Lightning2	62.30	59.02	68.85	59.02	**73.77**	67.21	68.85	68.85	71.67	65.57	68.85
Lightning7	42.47	30.14	71.23	60.27	46.58	**73.97**	56.16	67.12	59.72	71.23	**73.97**
Mallat	91.22	72.67	89.04	93.43	79.70	89.34	**96.59**	94.12	83.75	71.17	92.88
Meat	51.67	86.67	46.67	86.67	43.33	78.33	75.00	63.33	60.00	90.00	**93.33**
MedicalImages	**75.00**	61.05	71.71	59.61	65.53	68.68	60.79	72.63	64.34	66.18	69.08
MelbournePedestrian	88.23	85.94	89.50	88.48	93.32	94.75	85.98	87.00	87.13	94.71	**95.98**
MiddlePhalanxOutlineAgeGroup	48.70	64.94	62.34	59.09	52.60	50.65	**65.58**	57.79	58.44	51.95	63.64
MiddlePhalanxOutlineCorrect	**82.82**	81.44	57.39	59.11	79.38	71.48	51.89	57.39	48.80	69.07	80.41
MiddlePhalanxTW	28.57	55.84	**62.34**	58.44	47.40	51.95	59.74	54.55	55.84	51.95	55.19
MixedShapesRegularTrain	**96.29**	25.32	85.86	81.73	49.32	83.30	81.77	84.70	84.65	79.96	86.47
MixedShapesSmallTrain	**96.54**	24.95	80.54	75.84	33.15	79.51	80.62	79.51	70.01	75.51	80.37
MoteStrain	79.47	77.72	84.35	84.98	85.62	91.61	85.78	88.02	88.42	**91.77**	81.63
NonInvasiveFetalECGThorax1	**93.03**	52.47	84.83	87.53	75.52	82.85	84.53	86.41	73.37	78.98	84.53
NonInvasiveFetalECGThorax2	**93.94**	77.56	88.04	91.20	84.94	88.80	90.13	89.16	81.06	85.04	89.21
OliveOil	**86.67**	50.00	40.00	43.33	63.33	66.67	40.00	43.33	40.00	70.00	40.00
OSULeaf	**72.73**	21.90	54.96	43.39	43.80	41.74	53.31	58.26	40.91	45.45	49.59

Table 13: Performance comparison of Zero-shot TS-CLIP and supervised baselines on UCR datasets. TS-CLIP achieves the best results in 51 out of 128 datasets. All values are accuracy percentages (%), with the highest accuracy in **bold** and the second highest number is underlined.

Dataset	TS-CLIP	Autoformer	Crossformer	Dlinear	FEDFormer	Informer	iTransformer	patchTST	Pyraformer	Reformer	TimesNet
PhalangesOutlinesCorrect	75.52	75.29	64.45	65.73	**82.17**	67.25	63.52	65.97	63.52	66.43	77.16
Phoneme	**17.83**	13.71	14.72	9.86	12.76	13.61	10.76	15.93	9.18	13.29	16.46
PickupGestureWiimoteZ	46.00	34.00	62.00	**64.00**	48.00	**64.00**	58.00	60.00	60.00	58.00	52.00
PigAirwayPressure	**13.94**	4.81	10.10	9.13	2.88	6.73	8.65	5.77	5.29	5.77	9.62
PigArtPressure	**62.98**	8.17	17.79	14.90	4.81	13.94	13.94	9.13	12.50	12.50	17.31
PigCVP	**38.46**	4.33	9.13	8.65	2.88	5.77	9.13	8.65	9.62	13.94	12.98
PLAID	**83.99**	34.64	39.85	35.38	36.87	42.83	35.01	65.36	35.82	40.60	40.97
Plane	**100.00**	99.05	96.19	99.05	99.05	97.14	97.14	98.10	96.15	97.14	98.10
PowerCons	95.00	68.33	**100.00**	98.89	93.89	**100.00**	**100.00**	97.78	99.44	**100.00**	99.44
ProximalPhalanxOutlineAgeGroup	73.17	86.34	**86.83**	81.95	85.37	85.85	85.85	84.39	86.76	85.85	85.37
ProximalPhalanxOutlineCorrect	**89.69**	82.82	70.79	83.16	89.00	86.25	74.23	77.32	75.95	81.10	80.76
ProximalPhalanxTW	39.02	**79.51**	79.02	78.54	74.15	79.02	**79.51**	78.54	79.41	76.10	79.02
RefrigerationDevices	**53.33**	36.00	44.27	37.07	33.87	48.27	38.40	41.33	42.40	45.60	46.40
Rock	44.00	58.00	80.00	76.00	48.00	52.00	78.00	**88.00**	48.00	48.00	56.00
ScreenType	**47.73**	37.33	34.67	36.80	37.07	46.40	38.40	38.40	37.07	46.93	42.40
SemgHandGenderCh2	83.17	74.17	**95.00**	81.50	81.67	89.83	88.67	89.00	92.00	85.17	89.00
SemgHandMovementCh2	44.22	22.22	**72.67**	49.33	30.22	42.44	46.67	42.67	44.00	42.44	50.67
SemgHandSubjectCh2	35.33	41.33	**90.22**	80.89	53.78	75.78	80.22	86.00	88.22	72.00	78.22
ShakeGestureWiimoteZ	44.00	36.00	66.00	56.00	52.00	68.00	56.00	68.00	**74.00**	70.00	64.00
ShapeletSim	**66.11**	57.22	50.56	48.33	51.67	48.89	51.11	48.89	45.56	48.89	53.33
ShapesAll	**82.50**	12.50	72.00	57.33	22.67	59.17	67.33	66.67	41.50	57.67	68.83
SmallKitchenAppliances	**78.93**	59.47	56.27	38.40	59.73	52.27	36.27	54.13	61.07	52.80	57.87
SmoothSubspace	33.33	98.00	88.67	83.33	98.67	**99.33**	89.33	88.00	97.30	98.00	97.33
SonyAIBORobotSurface1	70.55	67.05	73.21	66.72	65.89	72.71	73.21	71.55	**73.50**	71.21	51.08
SonyAIBORobotSurface2	85.10	85.73	85.31	83.21	84.58	84.37	82.58	85.31	**88.24**	85.20	81.53
StarLightCurves	**97.20**	54.20	89.32	90.91	28.29	91.72	85.64	93.14	85.68	92.11	91.33
Strawberry	90.00	91.35	95.41	94.59	95.14	95.41	77.03	94.86	74.46	**96.49**	93.24
SwedishLeaf	**93.28**	78.37	89.26	81.41	88.78	88.62	72.44	91.03	78.85	86.38	85.58
Symbols	82.71	35.58	85.73	79.70	60.20	81.21	**86.13**	85.43	77.72	80.60	82.41
SyntheticControl	98.00	79.67	97.00	86.33	93.67	98.33	89.00	98.33	98.67	98.33	**99.33**
ToeSegmentation1	**79.39**	49.12	60.09	55.70	50.44	60.96	57.89	66.23	58.33	61.84	66.67
ToeSegmentation2	75.38	63.85	53.08	54.62	50.00	73.85	75.38	**87.69**	64.62	72.31	61.54
Trace	**100.00**	57.00	82.00	69.00	79.00	95.00	50.00	98.00	97.00	87.00	85.00
TwoLeadECG	**99.39**	79.46	88.76	57.68	86.57	65.85	83.23	85.95	74.28	76.47	58.56
TwoPatterns	99.48	80.25	99.35	83.00	67.55	88.13	82.08	**99.63**	98.03	86.43	98.25
UMD	90.97	84.03	**100.00**	92.36	95.83	98.61	94.44	98.61	**100.00**	72.22	73.61
UWaveGestureLibraryAll	**94.00**	35.71	92.04	80.51	75.46	90.23	89.22	91.26	89.64	91.32	92.46
UWaveGestureLibraryX	68.62	34.67	74.71	62.65	57.12	67.76	65.41	**77.61**	68.96	64.71	72.72
UWaveGestureLibraryY	69.68	27.28	63.85	59.10	49.50	58.21	57.79	**70.18**	62.95	58.01	63.54
UWaveGestureLibraryZ	63.74	31.99	68.23	54.33	52.46	60.30	58.68	**70.32**	60.75	54.08	64.54
Wafer	**99.94**	99.32	99.40	93.79	99.68	99.63	99.56	99.51	99.51	99.35	99.58
Wine	**74.07**	44.44	44.44	57.41	51.85	55.56	50.00	53.70	51.92	57.41	55.56
WordSynonyms	**62.38**	26.96	61.44	40.75	42.79	54.39	52.66	57.05	44.67	54.86	57.52
WormsTwoClass	54.55	48.05	61.04	55.84	54.55	50.65	54.55	50.65	**64.47**	55.84	58.44
Worms	**66.23**	44.16	54.55	38.96	53.25	37.66	48.05	50.65	48.68	38.96	53.25
Yoga	**85.77**	61.37	78.20	64.63	64.50	74.67	69.30	76.67	64.07	72.77	77.27

Table 14: (continued) Performance comparison of Zero-shot TS-CLIP and supervised baselines on UCR datasets. TS-CLIP achieves the best results in 51 out of 128 datasets. All values are accuracy percentages (%), with the highest accuracy in **bold** and the second highest number is underlined.