MERIT: Multi-Agent Collaboration for Unsupervised Time Series Representation Learning

Shu Zhou[1*], **Yunyang Xuan**[1*], **Yuxuan Ao**[1*], **Xin Wang**[2], **Tao Fan**[3], **Hao Wang**[1†]
[1]Nanjing University [2]Baidu [3]Nanjing University of Finance & Economics
{shuzhou, yunyangxuan, yxao}@smail.nju.edu.cn
{xinwang2749, fantao0916}@gmail.com, ywhaowang@nju.edu.cn

Abstract

This paper studies the problem of unsupervised time series representation learning, which aims to map unlabeled time series data into a low-dimensional latent space for various downstream tasks. Previous works usually combine a range of augmentation strategies with contrastive learning to generate discriminative representations. However, these augmentation strategies could alter the original semantics of time series data, which could degrade the performance of representation learning. To solve this problem, this paper incorporates the large language model (LLM) agent to guide unsupervised time series representation learning and proposes a novel framework named Multi-Agent Collaboration for Time-series Representation Learning (MERIT). The core of our MERIT is to utilize three LLM agents to collaboratively generate positive views for time series data. In particular, we first design a retrieval agent to automatically identify the relevant time series data from a coarse candidate set. Then, these selected sequences are further utilized to enhance an augmentation agent which automatically selects reliable augmentation strategies from an augmentation strategy library. We also design a review agent to evaluate the quality of generated views and stop the generation process. These three agents are designed to work in a loop for effective time series representation learning. Extensive experiments on various datasets demonstrate the effectiveness of MERIT compared with state-of-the-art baselines.

1 Introduction

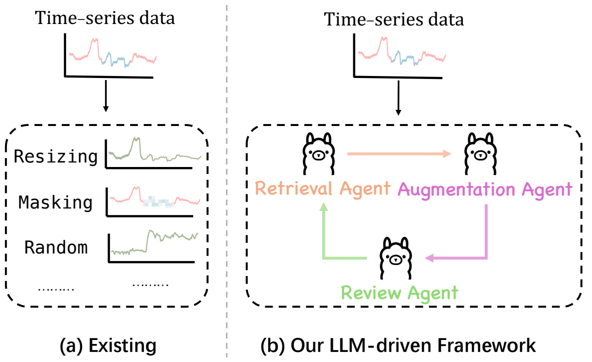
In the data-driven era, time series data exist widely in various fields including finance (Sezer et al., 2020; Liu et al., 2024), healthcare (Caballero Barajas and Akella, 2015), and transportation (Sun et al., 2023). Time series data is naturally high-dimensional, which brings challenges for analytical modeling (Qiu et al., 2011). Traditional feature engineering approaches are time-consuming and expert-dependent, which are difficult to generalize across different tasks. In contrast, unsupervised time series representation learning approaches (Liu and Chen, 2024a; Yang and Hong, 2022; Lafabregue et al., 2022) can map unlabeled time series data to a low-dimensional latent space without expensive annotation. In this way, they can automatically mine latent structural and temporal features to provide downstream tasks such as regression, classification, and anomaly detection (Eldele et al., 2024; Liu and Chen, 2024a).

Recent unsupervised time series representation learning approaches (Trirat et al., 2024; Yue et al., 2022; Eldele et al., 2021; Franceschi et al., 2019a) typically generate positive views using a series of hand-designed transformations, including random dithering, scaling, and cropping. They then utilize contrastive learning to ensure the augmented representations of the same sample are close to the other samples. However, random augmentation could distort key patterns and weaken feature structures, resulting in false positive views that degrade representation learning (Wen et al., 2020). For example, when applied to medical signal processing, excessive smoothing, and interpolation jitter may erase the characteristic peaks and valleys of the electrocardiogram, which hurts the semantics of time series data (Hemakom et al., 2023). In financial trading sequences, adding uncorrelated noise may hide the decision cues embedded in price fluctuations, leading to a decrease in the precision of subsequent forecasting and investment analysis (Huang et al., 2023b). Therefore, it is highly anticipated to have high-quality time series augmentation strategies with crucial semantics preserved in various scenarios to facilitate effective representation learning.

Due to the strong capacity of large language

[*] These authors contributed equally to this work.
[†] Corresponding author

Findings of the Association for Computational Linguistics: ACL 2025, pages 24011–24028
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics



(a) Existing

(b) Our LLM-driven Framework

Figure 1: Comparison between existing time series representation learning approaches (a) and our proposed LLM-driven framework (b).

models (LLMs) (Achiam et al., 2023; Anil et al., 2023; Touvron et al., 2023; Zhou et al., 2025a), we aim to incorporate LLMs into time series representation learning. However, there are still two major challenges in applying LLMs to time series data representation learning. ❶ *LLMs usually have a limited understanding of numerical signals* (Liu et al., 2023a; Liang et al., 2022; Zhou et al., 2025b; Taylor et al., 2022). Therefore, it is hard to require LLMs to directly generate reliable time series representations or augmented time series. LLMs could even generate wrong formats with the principle of next-token predictions. ❷ *Their ability for semantic understanding and feature extraction is insufficient under the zero-shot condition* (Wei et al., 2022; Sanh et al., 2021; Merrill et al., 2024), especially for complicated time series. Therefore, we should introduce contextual search and well-designed instructions to lay a solid foundation for constructing a more flexible, robust, and semantically preserved unsupervised time series representation learning framework.

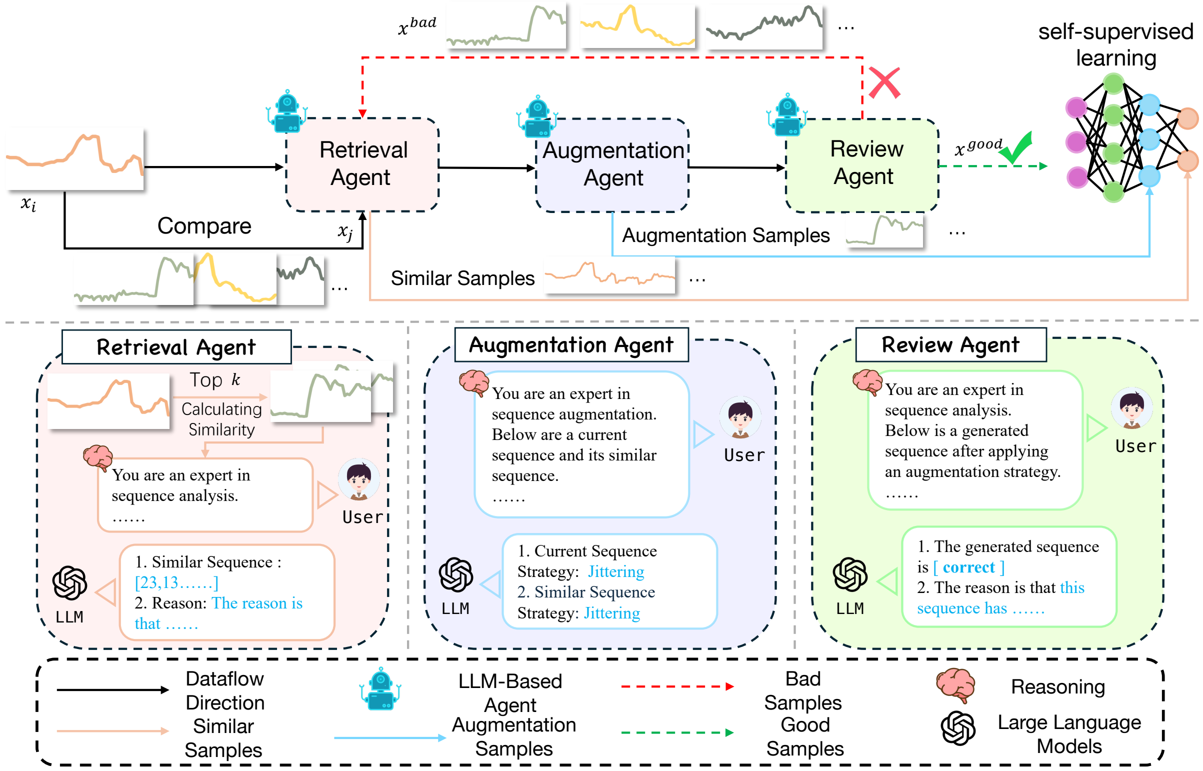
Towards this end, we propose a novel framework named Multi-Agent Collaboration for Time-series Representation Learning (MERIT). Different from existing representation learning methods, our MERIT utilizes three LLM-based agents to automatically generate high-quality positive sample views for time series data (see Figure 1), and thus achieve more reliable representations. In particular, we first introduce a retrieval agent, which utilizes the semantic understanding and association inference ability to select the relevant sequences from the database. To enhance the retrieval efficiency, we also calculate the similarity of time series which generate a coarse candidate. These selected relevant sequences are considered as context data with instructions for an augmentation agent, which automatically identifies reliable augmentation schemes from an augmentation strategy library. In this way, we can generate positive views that match the data characteristics, ensuring that the critical semantics can be preserved as much as possible for effective representation learning. To further ensure the reliability without potential hallucination, we introduce a review agent to evaluate the quality of the generated augmented views and terminate unsuitable augmentations if necessary. Finally, both appropriate augmented views and retrieved sequences are incorporated into a contrastive time series representation learning paradigm. Extensive experiments on a wide range of benchmark datasets validate the effectiveness of our MERIT in comparison with various state-of-the-art approaches.

The contributions of this paper are as follows: ❶ *Problem Connection.* We are the first to connect LLM agents with time series representation learning, which utilizes the reasoning capability of LLMs to enhance the time series learning paradigm. ❷ *Novel Methodology.* Our multi-agent collaborate framework MERIT utilizes a retrieval agent to extract context from the data, which enhances the reasoning ability of an augmentation agent for effective time series positive views. A review agent is also adopted to ensure the reliability. ❸ *Extensive Experiments.* Extensive experiments on several publicly available time series datasets results show that MERIT significantly outperforms existing comparative learning methods.

2 Related Work

2.1 Time Series Representation Learning

Time series representation learning has become increasingly important for various downstream tasks. Early approaches leveraged traditional dimensionality reduction techniques like Principal Component Analysis (PCA) (Pearson, 1901) and autoencoders (Kramer, 1991) to learn compact representations. Recent self-supervised learning advancements have introduced contrastive learning-based methods (Liu and Chen, 2024b; Luo et al., 2023; Wu et al., 2023; Yue et al., 2022), which generate positive pairs through data augmentation and maximize their agreement in the representation space. However, many methods rely on fixed or random augmentation, potentially distorting the semantics of time series. Some studies attempt to address this through carefully designed augmentations (Zhang



You are an expert in sequence analysis.
……

1. Similar Sequence : [23,13……]
2. Reason: The reason is that ……

You are an expert in sequence augmentation. Below are a current sequence and its similar sequence.
……

1. Current Sequence Strategy: Jittering
2. Similar Sequence Strategy: Jittering

You are an expert in sequence analysis. Below is a generated sequence after applying an augmentation strategy.
……

1. The generated sequence is [correct]
2. The reason is that this sequence has ……

Figure 2: An overview of MERIT. Our retrieval agent first selects neighborhood candidates with similarity, followed by refinement using an LLM. Our augmentation agent then selects suitable data augmentation strategies from the augmentation strategy library. Finally, our review agent evaluates the quality of the augmented sequences, which approves positive samples into a memory bank and returns to the retrieval agent for refinement if rejected. The augmented views and neighboring samples are utilized as positives for time series representation learning.

et al., 2023, 2022b; Eldele et al., 2021; Yang et al., 2021), but they lack adaptive selection based on data characteristics. More recent methods (Xu et al., 2023; Liu et al., 2023b) leverage masked autoencoding and use transformer architectures, but they still face challenges with maintaining semantic consistency during augmentation.

2.2 Multi-agent Systems

Multi-agent systems (MAS) (Baroni et al., 2022; Li et al., 2023; Park et al., 2024) attract growing interest in machine learning tasks, where agents interact to achieve individual or shared goals. Early works have focused on cooperative settings, with agents learning a shared representation through communication and coordination (Sunehag et al., 2018). More recent approaches explore competitive and adversarial settings, where agents learn distinct representations through competition (Lowe et al., 2017). Researchers have also applied MAS to multi-agent reinforcement learning and decentralized approaches for learning representations in multi-agent environments (Anschel et al., 2018; Gupta et al., 2017). Additionally, MAS has been explored in graph representation learning (Wang et al., 2024; Zhang et al., 2022a), unsupervised representation learning (Zhu et al., 2022), and cross-modal representation learning (Zhang et al., 2024), highlighting its potential in addressing complex representation learning challenges. Towards this end, this paper proposes a novel approach MERIT, which uses LLMs agents to collaboratively generate positive views for time series data.

3 The Proposed MERIT

This paper studies the problem of time series representation learning and proposes a multi-agent framework named MERIT, which incorporates three LLM-based intelligent agents in a closed-loop collaboration mechanism to dynamically generate high-quality positive views while ensuring the preservation of semantics. In particular, our framework consists of three LLM-based agents: (1) *Retrieval agent*, which first selects the candidates for the input sequence and then identifies the relevant context; (2) *Augmentation agent*, which selects appropriate augmentation strategies based on context inforamtion; (3) *Review agent*, which evaluates the quality of the generated views. The whole process enables the MERIT framework to dynamically generate views of positive samples and to improve the discrimination and transferability of time series representations. The overview of our MERIT can be found in Figure 2.

3.1 Problem Definition

Given a dataset $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ of N time series $\boldsymbol{x}_i$ of length T with C channels, the objective of self-supervised learning is to learn a function f_θ, such that $\forall i \in [1, N]$, $\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i)$. Each $\boldsymbol{z}_i \in \mathbb{R}^d$ is a d-dimensional representation of time series $\boldsymbol{x}_i$, which should preserve as much information of the original data as possible. In this work, $f_\theta(\cdot)$ is learned fully from unlabeled data X.

3.2 Retrieval Agent for Context Mining

In zero-shot scenarios, directly applying LLMs to raw time series data can be problematic due to their limited understanding of numerical signals (Liu et al., 2023a; Liang et al., 2022; Taylor et al., 2022). Therefore, our framework first utilizes a retrieval agent to provide LLM with high-quality context from the dataset, which is beneficial to the subsequent process. Our retrieval agent takes a two-step paradigm, which first selects a set of candidate sequences based on their similarity to the target sequence and then further refines this set using the semantic reasoning capabilities of LLMs.

In particular, we first identify a coarse candidate set $\mathcal{C}_i$ for the target sequence $\boldsymbol{x}_i \in \mathbb{R}^{T \times C}$, which can greatly save the cost of LLMs. By calculating the similarity scores between the target sequence and other sequences in the dataset, we select the top K sequences that are similar to $\boldsymbol{x}_i$ as the candidate set as follows:

$$\mathcal{C}_i = \{\boldsymbol{x}_j | \boldsymbol{x}_j \in \text{Top-K}(\text{Sim}(\boldsymbol{x}_i, \boldsymbol{x}_j)), \boldsymbol{x}_j \neq \boldsymbol{x}_i\} \tag{1}$$

where $\text{Sim}(\cdot, \cdot)$ is the function used to measure sequence similarity and $\text{Top-K}(\cdot)$ returns the set of samples with top K scores.

Then, we adopt an LLM to further narrow down the scope by designing a prompt containing inference instructions, which semantically filter these candidate sequences for the similar sequences. In formulation, given a target sequence $\boldsymbol{x}_i$ and a candidate sequence $\boldsymbol{x}_j$, the prompt contains an inference requirement that instructs the LLM to explain why $\boldsymbol{x}_j$ is similar to $\boldsymbol{x}_i$ and generates a semantic relevance score $\text{Rel}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ based on its explanation:

$$\text{Rel}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \text{LLM}(\boldsymbol{x}_i, \boldsymbol{x}_j, \mathcal{P}) \tag{2}$$

where the prompt $\mathcal{P}$ includes the description of the target sequence $\boldsymbol{x}_i$ and the context of the candidate sequence $\boldsymbol{x}_j$, and the LLM is asked to provide the reason for the similarity. Finally, based on the semantic relevance scores, we select the highest scoring M sequences to form the final reference set $\mathcal{R}_i$:

$$\mathcal{R}_i = \{\boldsymbol{x}_j | \boldsymbol{x}_j \in \text{Top-M}(\text{Rel}(\boldsymbol{x}_i, \boldsymbol{x}_j)) \cap \mathcal{C}_i\} \tag{3}$$

In this way, we are able to construct reference context with high semantic relevance $\mathcal{R}_i$ for the target time series, which provides a data-centric view of contextual support for the subsequent paradigm.

3.3 Augmentation Agent with Adaptive Strategy Selection

Positive views from augmentation are the key to unsupervised time series representation learning (Yue et al., 2022; Eldele et al., 2021). Existing time series augmentation approaches often rely on a fixed set of transformations, which may potentially distort crucial semantic information in diverse scenarios (Zhang et al., 2022b; Yang et al., 2021; Eldele et al., 2021). To address this, our augmentation agent utilizes contextual information provided by the retrieval agent to dynamically select appropriate augmentation strategies in a data-centric way. By adapting to the specific characteristics of the time series data, we can ensure that the generated positive views are semantically reliable.

In detail, from the reference set $\mathcal{R}_i$, our augmentation agent is required to choose a suitable augmentation strategy for the target sequence $\boldsymbol{x}_i$. Towards this end, we require an LLM to generate the set of augmentation strategies, which are applicable to the current data, i.e.,

$$\mathcal{S}_i = \text{LLM}(\boldsymbol{x}_i, \mathcal{R}_i, \mathcal{S}), \tag{4}$$

where $\mathcal{S}$ = {Sailing, Resizing, Jittering, Flipping, Permutation, Time Masking, Frequency Masking, Time Neighboring} is the augmentation strategy library and $\mathcal{S}_i \subseteq \mathcal{S}$. Then, we apply these strategies to the target sequence $\boldsymbol{x}_i$ to generate the corresponding augmented samples:

$$\hat{\boldsymbol{x}}_i^s = s(\boldsymbol{x}_i), \quad \forall s \in \mathcal{S}_i, \tag{5}$$

where $\hat{\boldsymbol{x}}_i^s$ is the augmented sample with the strategy s. A set of positive views can be obtained by combining the augmented views of $\boldsymbol{x}_i$ and its similar sequence $\boldsymbol{x}_j \in \mathcal{R}_i$ respectively:

$$\hat{X}_i^P = \{\hat{\boldsymbol{x}}_i^s | s \in \mathcal{S}_i\} \cup \{\boldsymbol{x}_j | j \in \mathcal{R}_i\}. \tag{6}$$

We also conduct augmentation on $\boldsymbol{x}_j$ to promote diverse positive views.

3.4 Review Agent for Reliable Actions

Although we have generated adaptive positive views for specific target sample, there is still a chance that the LLM may generate semantically inappropriate or distorted augmented samples due to LLM hallucination (Ji et al., 2023; Huang et al., 2023a). To mitigate this risk and further enhance the reliability of our framework, we introduce a review agent to assess the quality of augmentation actions and provide feedback. By carefully reviewing each augmented sample, we can ensure that only high-quality views are used for time series representation learning.

Specifically, after generating the augmented samples $\hat{\boldsymbol{x}}_i^s$, our review agent evaluates our augmentation action. Here, we feed the target sequence $\boldsymbol{x}_i$ and the augmented sequence $\hat{\boldsymbol{x}}_i^s$ into an LLM, and the LLM outputs a quality assessment result, $\text{Quality}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i^s)$, along with the reasoning process:

$$\text{Quality}(\hat{\boldsymbol{x}}_i^s) = \text{LLM}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i^s), \tag{7}$$

where the output of quality assessment is 'Correct' or 'Error'. Based on the assessment results, we store the approved positive views in a memory bank $\mathcal{M}$ each time and re-start the searching procedure:

$$\mathcal{M} \leftarrow \mathcal{M} \cup \{\hat{\boldsymbol{x}}_i^s | s \in \mathcal{S}_i, \text{Quality}(\hat{\boldsymbol{x}}_i^s) = \text{'Correct'}\}. \tag{8}$$

We will stop the procedure when no augmented view is rejected by the review agent or the memory bank is full. This iterative process ensures that only reliable and semantically consistent views are used for representation learning.

3.5 Time Series Representation Learning

After generating reliable positive views, we incorporate them into a contrastive learning framework, which maximizes the similarity between the original time series and its augmented views while minimizing the similarity with other samples (Hu et al., 2024; He et al., 2020; Tian et al., 2020). In this way, we can learn effective time series representations for different downstream tasks.

In formulation, for each target sequence $\boldsymbol{x}_i \in X$, the set of positive samples $\hat{X}_i^P$ can be rewritten as:

$$\hat{X}_i^P = \mathcal{M} \cup \{\boldsymbol{x}_j | j \in \mathcal{R}_i\}. \tag{9}$$

where $\mathcal{M}$ is positive views approved by the memory bank and, $\mathcal{R}_i$ is the set of semantically similar sequences selected by the retrieval agent. We utilize an encoder f_θ to map the original sequence and its positive samples to the representation space:

$$\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i), \quad \hat{\boldsymbol{z}}_{ir}^s = f_\theta(\hat{\boldsymbol{x}}_{ir}), \hat{\boldsymbol{x}}_{ir} \in \hat{X}_i^P. \tag{10}$$

The contrastive learning loss $\mathcal{L}$ is written as:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\hat{X}_i^P|} \sum_{j=1}^{\hat{X}_i^P} - \log \frac{\exp(\frac{\text{sim}(\boldsymbol{z}_i, \hat{\boldsymbol{z}}_{ir})}{\tau})}{\sum_{k=1}^{N} \exp(\frac{\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)}{\tau})} \tag{11}$$

where $\text{sim}(\boldsymbol{z}_a, \boldsymbol{z}_b) = \frac{\boldsymbol{z}_a^\top \boldsymbol{z}_b}{\|\boldsymbol{z}_a\| \|\boldsymbol{z}_b\|}$ denotes the cosine similarity of the two representation vectors. τ is a temperature parameter that regulates the scale of the similarity. In this way, we maximize the mutual information between target data and its positive views for effective time series representations.

4 Experiments

Baselines. To evaluate the performance of MERIT, we compare it with a wide range of state-of-the-art baselines, including unsupervised, self-supervised, and fully supervised methods: DTW (Müller, 2007), DONUT (Xu et al., 2018), SR (Ren et al., 2019), N-BEATS (Oreshkin et al., 2019), LogTrans (Li et al., 2019), TS-TCC (Eldele et al., 2021), TNC (Tonekaboni et al., 2021), T-Loss (Eldele et al., 2021), Informer (Zhou et al., 2021), TST (Zerveas et al., 2021), CoST (Woo et al., 2022), TS2Vec (Yue et al., 2022), InfoTS (Luo et al., 2023) and TimesURL (Liu and Chen, 2024b). These methods are evaluated across various downstream tasks, including classification, imputation, forecasting, anomaly detection, and transfer learning. More details can be found in Appendix C.

Implementation Details. For MERIT, we implement a multi-agent collaboration framework consisting of three specialized agents. Each agent is designed with specific roles and responsibilities in the representation learning process. More details can be found in the Appendix E.

4.1 Classification

Setups. To evaluate the effectiveness of MERIT on time series classification and transfer learning, we

Method	AWB	AF	BM	CT	CR	CDG	EW	EP	EC	ER	FD	FM	HMD	HW	HB	Average
DTW	98.0	22.0	97.5	96.9	94.4	27.5	55.7	96.4	29.3	13.3	52.8	55.0	27.8	28.5	71.7	62.9
TST	98.3	26.7	97.5	97.3	95.8	35.0	61.1	97.1	32.3	13.3	54.7	56.7	30.6	31.5	73.3	61.7
TS-TCC	98.7	26.7	**100.0**	97.8	97.2	40.0	61.8	97.8	33.1	13.3	55.9	58.3	33.3	32.3	74.2	67.0
T-Loss	98.5	26.7	97.5	97.6	97.2	37.5	61.1	97.5	32.7	13.3	55.3	58.3	30.6	31.9	73.8	65.8
TS2Vec	98.9	26.7	**100.0**	98.1	97.2	42.5	62.6	98.2	33.8	13.3	56.5	60.0	33.3	33.1	75.0	70.4
TimesURL	99.0	26.7	**100.0**	98.4	98.6	45.0	64.1	98.6	34.2	13.3	57.2	61.7	36.1	34.6	75.8	75.2
MERIT	**99.3**	**33.3**	**100.0**	**98.7**	**100.0**	**47.5**	**65.6**	**99.3**	**35.0**	**13.3**	**58.4**	**63.3**	**38.9**	**35.4**	**76.7**	**77.2**

Table 1: Classification accuracy (%) on 30 *UEA* datasets. The best results are highlighted in bold. More results can be found in Appendix J.

Target	TNC	T-Loss	TS2Vec	TimesURL	MERIT
Synthetic Control	78.3	82.1	85.6	87.2	**89.5**
Two Patterns	76.5	80.8	83.9	85.7	**87.8**
Wafer	75.2	79.4	82.7	84.3	**86.4**
ECG200	77.8	81.5	84.2	86.1	**88.3**
ECGFiveDays	76.9	80.6	83.5	85.2	**87.4**
TwoLeadECG	75.6	79.2	82.8	84.6	**86.7**
Average	76.7	80.6	83.8	85.5	**87.7**

Table 2: Transfer learning results (accuracy %) from source (*CBF* or *CinCECGTorso*) to target domains.

Dataset		InfoTS		TimesURL		MERIT	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.125	0.659	0.640	0.717	0.666	**0.639**	**0.630**
	0.250	0.679	0.648	0.726	0.674	**0.659**	**0.638**
	0.375	0.702	0.656	0.726	0.676	**0.682**	**0.646**
	0.500	0.712	0.693	0.783	0.695	**0.692**	**0.683**
ETTh2	0.125	2.455	1.215	2.491	1.199	**2.435**	**1.205**
	0.250	2.560	1.239	2.644	1.244	**2.540**	**1.229**
	0.375	2.673	1.269	2.757	1.266	**2.653**	**1.259**
	0.500	2.701	1.281	2.844	1.283	**2.681**	**1.271**
Avg.		1.326	0.860	1.386	0.864	**1.306**	**0.850**

Table 3: Imputation results on *ETT* dataset with 15% missing values.

conduct experiments on 30 datasets from the *UEA* archive (Bagnall et al., 2018) and *UCR* archive (Dau et al., 2019). For classification, we train MERIT on the training set and extract representations for both training and testing sets. A linear classifier is then trained on the extracted representations from the training set and evaluated on the testing set. For transfer learning, we evaluate the transferability of the learned representations using datasets from the *UCR* archive (Dau et al., 2019). Following (Yue et al., 2022), we first pre-train the model on a source dataset and then transfer the learned representations to target datasets for classification tasks. We use *CBF* and *CinCECGTorso* as source domains and evaluate different methods on target domains with different characteristics.

Results. Table 1 presents the classification accuracy of different methods. MERIT achieves the highest average accuracy of 77.2%, outperforming other methods and demonstrating the effectiveness of our multi-agent collaborative framework for time series representation learning. Table 2 shows the classification accuracy on target domains. The results demonstrate that MERIT achieves superior transfer performance across different target datasets. This superior performance can be attributed to the following reasons: ❶ The retrieval agent effectively identifies the most relevant context for each time series, providing a strong foundation for augmentation. ❷ The augmentation agent selects the most suitable augmentation strategies based on the specific characteristics of the time series and its context, ensuring the generation of high-quality augmented views. ❸ The review agent filters out low-quality augmented views, further enhancing the quality of the learned representations.

4.2 Imputation

Setups. Time series data often suffers from missing values due to sensor failures or irregular sampling. We evaluate the imputation performance of MERIT on the *ETT* dataset (Zhou et al., 2021), which contains power load data collected from electricity transformers. Following recent works (Yue et al., 2022), we randomly mask 15% of the values in the test set and use the learned representations to reconstruct the missing values.

Results. Table 3 shows the MSE and MAE results on the *ETT* dataset. We can observe that MERIT achieves the best performance across all settings, demonstrating its effectiveness in learning representations that capture the underlying patterns of time series data. The superior performance can be attributed to the collaborative mechanism among the three agents, which helps to generate high-quality augmented views that preserve the semantic information of the original time series.

4.3 Forecasting

Setups. Time series forecasting is a fundamental task in many real-world applications. We evaluate our proposed MERIT on both short-term and long-term forecasting tasks using the *ETT* dataset

Method		TimesURL		CoST		MERIT	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.125	0.659	0.640	0.690	0.658	**0.636**	**0.631**
	0.250	0.679	0.648	0.710	0.668	**0.657**	**0.637**
	0.375	0.702	0.656	0.728	0.676	**0.681**	**0.647**
	0.500	0.712	0.693	0.751	0.682	**0.694**	**0.682**
ETTh2	0.125	2.455	1.215	2.866	1.288	**2.437**	**1.203**
	0.250	2.560	1.239	2.792	1.271	**2.542**	**1.227**
	0.375	2.673	1.269	2.793	1.271	**2.655**	**1.258**
	0.500	2.701	1.281	2.769	1.267	**2.684**	**1.272**
Avg.		1.643	0.955	1.762	0.973	**1.624**	**0.946**

Table 4: Forecasting results on the *ETT* dataset.

(Zhou et al., 2021). Following recent works (Yue et al., 2022), we consider four different prediction lengths: 12.5%, 25%, 37.5%, and 50% of the input sequence length. For each setting, we use the learned representations to predict future values through a simple MLP network.

Results. Table 4 shows the MSE and MAE results for both short-term and long-term forecasting. We can observe that MERIT consistently outperforms all baseline methods across different prediction lengths, demonstrating its strong capability in capturing temporal dependencies. The performance improvement is particularly significant for longer prediction horizons, which suggests that our multi-agent collaboration framework is effective in learning robust and generalizable representations for time series forecasting.

4.4 Anomaly Detection

Setups. Detecting anomalies from monitoring data is essential for industrial maintenance and system reliability. We evaluate MERIT on two benchmark datasets: KPI (Ren et al., 2019), a competition dataset containing multiple minutely sampled KPI curves, and Yahoo (Nikolay Laptev, 2015), which includes 367 hourly sampled time series. Following recent research (Ren et al., 2019), we adopt a streaming evaluation protocol that determines whether the last point in a time series slice is anomalous. During training, each time series is split into two halves according to time order, with the first half for training and the second for evaluation.

Results. Table 5 demonstrates the compared performance across different methods on two datasets. From the results, we can find that MERIT achieves superior performance across both datasets from Table 5, particularly showing significant improvements in F1-score. This demonstrates that our multi-agent collaboration framework can effectively capture anomalous patterns while maintaining high precision and recall rates.

Datasets		KPI			Yahoo	
Metrics	F1	Precision	Recall	F1	Precision	Recall
SPOT	0.751	0.783	0.722	0.847	0.856	0.839
DSPOT	0.768	0.795	0.743	0.861	0.872	0.850
DONUT	0.779	0.803	0.756	0.873	0.885	0.862
SR	0.785	0.812	0.760	0.879	0.890	0.868
TS2Vec	0.791	0.818	0.766	0.885	0.896	0.874
TimesURL	0.803	0.825	0.782	0.892	0.901	0.883
MERIT	**0.815**	**0.836**	**0.795**	**0.905**	**0.913**	**0.897**

Table 5: Anomaly detection results on the KPI and Yahoo datasets.

Task	Classification		Forecasting	
Dataset	***UCR***	***UEA***	***ETTh1***	***ETTh2***
MERIT w/o RET	84.2	75.1	0.672	2.489
MERIT w/o AUG	83.1	74.3	0.695	2.523
MERIT w/o REV	84.9	75.8	0.668	2.476
MERIT (Full Model)	**86.8**	**77.2**	**0.639**	**2.435**

Table 6: Ablation study of our MERIT in both classficiation and forecasting tasks.

4.5 Further Analysis

Ablation Study. To analyze the effectiveness of different components in our MERIT, we introduce three different variants: (i) MERIT w/o RET, which replaces the context retrieval with random sampling. (ii) MERIT w/o AUG, which uses fixed augmentation strategies instead of our augmentation agent. (iii) MERIT w/o REV, which removes the procedure of our review agent. Table 6 shows the performance comparison on classification and forecasting tasks. From the results, we have the following observations: ❶ Removing the retrieval agent results in a slight decrease in performance, suggesting that the contextual information provided by the retrieval agent helps the model learn a better representation. ❷ MERIT w/o AUG performs much worse than the full model, indicating the importance of dynamic augmentation strategy selection. ❸ MERIT outperforms MERIT w/o REV, indicating that the review agent is effective in filtering low-quality augmented views.

Sensitivity Analysis. Figure 3 illustrates the sensitivity of MERIT to the number of retrieved candidates and the number of semantically relevant sequences. For retrieved candidates, performance tends to improve as retrieved candidates increase from 1 to 7, suggesting that considering more candidates is beneficial for identifying relevant contexts. However, performance tends to decrease when the number is too large, which could result from the potential noise of too many samples. For the number

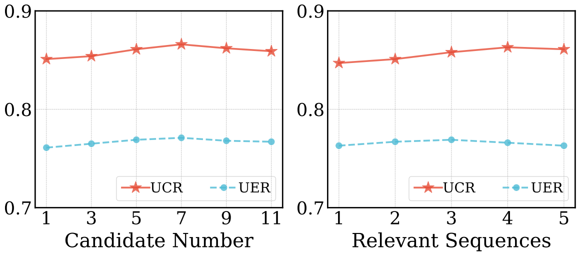


Figure 3: Sensitivity analysis of our proposed MERIT with respect to the number of candidate numbers and relevant sequences.

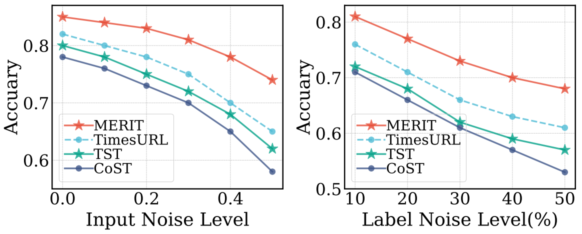


Figure 4: Robustness of MERIT classification accuracy to input noise (left) and label noise (right).

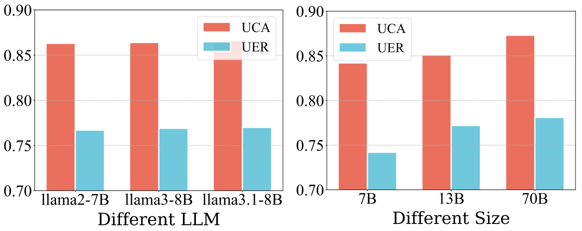
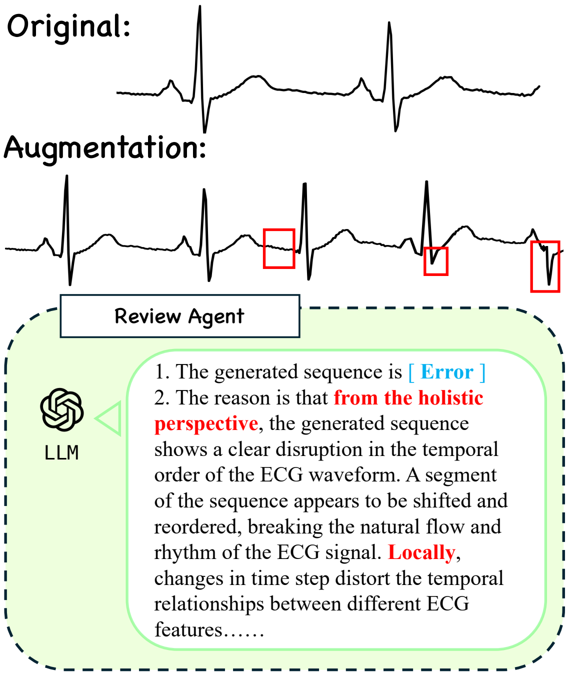


Figure 5: Performance of our MERIT with different LLMs and sizes.

of semantically relevant sequences, the model performance improves as relevant sequences increase from 1 to 4 before saturation. The performance of our MERIT is quite stable when the number of relevant sequences is between 3 to 4.

Robustness Analysis to Noise. To evaluate the robustness of the MERIT framework, we add different levels of noise to the input data in a time series classification task and set the noise level from 0 to 0.5. From the results, we can observe that as the noise level increases, the performance of all models decreases, while MERIT consistently outperforms the other models. We also show the effect of the percentage of random label flip noise on the model. The results also show that our MERIT achieves higher performance than others. These results highlight the robustness of our multi-agent framework in the presence of noise and its advantage over methods that do not explicitly explore semantic information during augmentation.

Different LLMs and Sizes. In this part, we evaluate the performance of our proposed MERIT with respect to different LLMs and sizes. Firstly, we compare the performance of different LLMs including LLaMa2-7B, LLaMa3-8B and LLaMa3.1-8B with similar parameters. The results are shown in Figure 5. From the results, we can observe that more advanced large modeling frameworks lead to better results. Then we vary different sizes of LLaMa2 in {7B, 13B and 70B}. From the results, we can still find that bigger parameters give better results. However, both different LLMs and sizes have restricted improvement in results.



1. The generated sequence is [**Error**]
2. The reason is that **from the holistic perspective**, the generated sequence shows a clear disruption in the temporal order of the ECG waveform. A segment of the sequence appears to be shifted and reordered, breaking the natural flow and rhythm of the ECG signal. **Locally**, changes in time step distort the temporal relationships between different ECG features……

Figure 6: An example of the review agent on the ECG dataset. The review agent reasons from two perspectives: **holistic** and **locally**.

Case Study. Here, we study how the review agent can reason from multiple perspectives to make an “accept” or “reject” decision. In particular, Figure 6 shows a case where the proposed augmentation is rejected by the review agent. This case study reveals that the review agent is able to effectively recognize and reject the introduction of semantically distorted or low-quality augmented views, thus ensuring that the MERIT framework produces high-quality, semantically reliable time series representations. This highlights the critical role of the review agent as a quality gatekeeper in the MERIT framework and the reasoning capability of the LLM in complex quality assessment tasks.

5 Conclusion

In this paper, we study the problem of unsupervised time series representation learning and we present a novel multi-agent collaboration framework MERIT for this problem. By leveraging LLMs as intelligent decision-makers, our framework introduces three agents with different functionalities that work collaboratively to generate high-quality positive views for contrastive learning. Extensive experiments demonstrate that MERIT consistently outperforms existing methods across various downstream tasks. Future research directions include optimizing LLM inference efficiency, and extending the framework to handle multi-modal time series data.

Limitations

MERIT is currently designed and evaluated for univariate and multivariate time series data. However, the real world presents diverse data forms, including event sequences, time series with accompanying text or images, and heterogeneous data. The framework's adaptability to such data modalities remains an area for future exploration.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 72074108) and Fundamental Research Funds for the Central Universities at Nanjing University (Grant No. 010814370338), Jiangsu Young Talents in Social Sciences and Tang Scholar of Nanjing University.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Oron Anschel, Nir Baram, and Nahum Shimkin. 2018. Learning representations in a multi-agent environment. In *International Conference on Learning Representations*.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*.

Marco Baroni, Roberto Dessi, and Angeliki Lazaridou. 2022. Emergent language-based coordination in deep multi-agent systems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 11–16, Abu Dubai, UAE. Association for Computational Linguistics.

Karla L Caballero Barajas and Ram Akella. 2015. Dynamically modeling patient's health state from electronic medical records: A time series approach. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 69–78.

Zhengping Chen, Jianmin Wu, Yingjie Xu, and Xiaoyong Wang. 2023. Timegpt: Large language models for temporal pattern understanding. *arXiv preprint arXiv:2310.03589*.

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, and Xiaoli Li. 2024. Label-efficient time series representation learning: A review. *IEEE Transactions on Artificial Intelligence*.

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*.

Archibald Fraikin, Adrien Bennetot, and Stéphanie Allassonnière. 2023. T-rep: Representation learning for time series using time-embeddings. *arXiv preprint arXiv:2310.04486*.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. 2019a. Unsupervised scalable representation learning for multivariate time series. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

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

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Apit Hemakom, Danita Atiwiwat, and Pasin Israsena. 2023. Ecg and eeg based detection and multilevel classification of stress using machine learning for specified genders: A preliminary study. *Plos one*, 18(9):e0291070.

Haigen Hu, Xiaoyuan Wang, Yan Zhang, Qi Chen, and Qiu Guan. 2024. A comprehensive survey on contrastive learning. *Neurocomputing*, page 128645.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Wei-Chia Huang, Chiao-Ting Chen, Chi Lee, Fan-Hsuan Kuo, and Szu-Hao Huang. 2023b. Attentive gated graph sequence neural network-based time-series information fusion for financial trading. *Information Fusion*, 91:261–276.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mark A Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243.

Baptiste Lafabregue, Jonathan Weber, Pierre Gançarski, and Germain Forestier. 2022. End-to-end deep representation learning for time series clustering: a comparative study. *Data Mining and Knowledge Discovery*, 36(1):29–81.

Huao Li, Yu Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. 2023. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 180–192, Singapore. Association for Computational Linguistics.

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

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B. Aditya Prakash. 2024. LST-Prompt: Large language models as zero-shot time series forecasters by long-short-term prompting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7832–7840, Bangkok, Thailand. Association for Computational Linguistics.

Jiexi Liu and Songcan Chen. 2024a. Timesurl: Self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13918–13926.

Jiexi Liu and Songcan Chen. 2024b. Timesurl: Self-supervised contrastive learning for universal time series representation learning. 38(12):13918–13926.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Siteng Liu, Zhao Lin, Yue Wu, and Tianyi Zhou. 2023b. Tst: Time series transformer for forecasting and anomaly detection. *arXiv preprint arXiv:2301.04028*.

Yang Liu, Shuang Li, Zhenghua Zhang, and Haoyi Wang. 2023c. Tstext: Time series as text for language model-based analysis. *arXiv preprint arXiv:2308.11228*.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pages 6379–6390.

Dongsheng Luo, Wei Cheng, Yingheng Wang, Dongkuan Xu, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Yanchi Liu, Yuncong Chen, Haifeng Chen, et al. 2023. Time series contrastive learning with information-aware augmentations. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Mike A Merrill, Mingtian Tan, Vinayak Gupta, Thomas Hartvigsen, and Tim Althoff. 2024. Language models still struggle to zero-shot reason about time series. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3512–3533, Miami, Florida, USA. Association for Computational Linguistics.

Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84.

Yuri Billawala Nikolay Laptev, Saeed Amizadeh. 2015. A benchmark dataset for time series anomaly detection.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*.

Someen Park, Jaehoon Kim, Seungwan Jin, Sohyun Park, and Kyungsik Han. 2024. PREDICT: Multi-agent-based debate simulation for generalized hate speech detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20963–20987, Miami, Florida, USA. Association for Computational Linguistics.

Karl Pearson. 1901. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Peng Qiu, Erin F Simonds, Sean C Bendall, Kenneth D Gibbs Jr, Robert V Bruggner, Michael D Linderman, Karen Sachs, Garry P Nolan, and Sylvia K Plevritis. 2011. Extracting a cellular hierarchy from high-dimensional cytometry data with spade. *Nature biotechnology*, 29(10):886–891.

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time series anomaly detection with multiresolution ensemble decoding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4027–4034.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181.

Le Sun, Jiancong Liang, Chunjiong Zhang, Di Wu, and Yanchun Zhang. 2023. Meta-transfer metric learning for time series classification in 6g-supported intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech M Czarnecki, Vinicius Zambaldi, et al. 2018. Value-decomposition networks for cooperative multi-agent learning. In *International Conference on Learning Representations*.

Yuqi Tang, Yuanhang Chen, Xiyuan Zhang, Yifang Wang, Zhenghua Zhang, et al. 2023. Foundation models for time series: A survey. *arXiv preprint arXiv:2310.07820*.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer.

Siavash Tonekaboni, Danny Eytan, and Anna Goldenberg. 2021. Unsupervised representation learning for time series with temporal neighborhood coding. In *International Conference on Learning Representations*.

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

Patara Trirat, Yooju Shin, Junhyeok Kang, Youngeun Nam, Jihye Na, Minyoung Bae, Joeun Kim, Byunghyun Kim, and Jae-Gil Lee. 2024. Universal time-series representation learning: A survey. *arXiv preprint arXiv:2401.03717*.

Jingwei Wang, Qianyue Hao, Wenzhen Huang, Xiaochen Fan, Zhentao Tang, Bin Wang, Jianye Hao, and Yong Li. 2024. Dyps: Dynamic parameter sharing in multi-agent reinforcement learning for spatio-temporal resource allocation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3128–3139.

Zhixin Wang, Xiyuan Zhang, Hao Wang, and Jun Zhou. 2023. Timellm: Understanding time series data through large language models. *arXiv preprint arXiv:2310.01927*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2020. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. 2022. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*.

Zhongqing Wu, Junhan Lin, Wei Chen, Jia Zhou, and Hui Yang. 2023. Temporal knowledge acquisition via large language models. *arXiv preprint arXiv:2305.14019*.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, pages 187–196.

Zhihan Xu, Yanbo Shang, Guangyi Wang, Jingyuan Li, et al. 2023. Timemae: Self-supervised representations of time series with decoupled masked autoencoders. *arXiv preprint arXiv:2303.00320*.

Chaochao Yang, Lei Chen, Qiannan Guo, Xiaoyun Wang, and Shuang Li. 2021. Robust time series representation learning with multi-view attention consistency. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1493–1498.

Ling Yang and Shenda Hong. 2022. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International conference on machine learning*, pages 25038–25054. PMLR.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124.

Lei Zhang, Binglu Wang, Yongqiang Zhao, Yuan Yuan, Tianfei Zhou, and Zhijun Li. 2024. Collaborative multimodal fusion network for multiagent perception. *IEEE Transactions on Cybernetics*.

Wei Zhang, Jinyang Wang, Wei Chen, and Jiashu Huang. 2023. Seriesllm: Learning language models on time series data. *arXiv preprint arXiv:2309.08482*.

Weijia Zhang, Hao Liu, Jindong Han, Yong Ge, and Hui Xiong. 2022a. Multi-agent graph convolutional reinforcement learning for dynamic electric vehicle charging pricing. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2471–2481.

Xiyuan Zhang, Zhao Yu, Qitian Zhou, and Yong Li. 2022b. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. *arXiv preprint arXiv:2202.01575*.

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.

Shu Zhou, Xin Wang, Jingwen Qiu, Xiaomin Li, Bin Shi, and Hao Wang. 2025a. Losdf: A logical optimization and semantic decoupling framework for question answering in multi-party conversations. *Information Processing & Management*, 62(5):104200.

Shu Zhou, Rui Zhao, Zhengda Zhou, Haohan Yi, Xuhui Zheng, and Hao Wang. 2025b. Enhancing extractive question answering in multiparty dialogues with logical inference memory network. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8725–8738, Abu Dhabi, UAE. Association for Computational Linguistics.

Changxi Zhu, Mehdi Dastani, and Shihan Wang. 2022. A survey of multi-agent reinforcement learning with communication. *arXiv preprint arXiv:2203.08975*, 1.

A More Related Work

A.1 Large Language Models for Time Series Analysis

Large Language Models (LLMs) are demonstrating remarkable capabilities beyond natural language processing, leading to growing interest in their application to time series analysis. Initial explorations focused on leveraging LLMs for time series forecasting by generating natural language descriptions of temporal patterns (Tang et al., 2023). Subsequently, research has expanded to encompass various approaches for integrating LLMs into time series tasks, including prompt engineering to enhance LLMs' understanding of temporal data (Wu et al., 2023), combining LLMs with traditional time series models (Wang et al., 2023; Zhang et al., 2023), and leveraging LLMs for semantic understanding of time series data by converting them into natural language descriptions or generating human-readable explanations for temporal patterns (Liu et al., 2023c; Chen et al., 2023). These diverse research directions highlight the growing potential of LLMs to enhance various aspects of time series analysis.

B Datasets Details

Classification Tasks: The *UEA* archive (Bagnall et al., 2018) is a widely used benchmark for evaluating time series classification algorithms. It comprises a diverse collection of time series datasets from various domains, including motion capture, sensor data, and medical recordings. These datasets vary in length, number of classes, and complexity, providing a comprehensive and challenging evaluation environment for time series representation learning methods. For the *UEA* datasets, we use the original train/test splits.

Forecasting Tasks: We utilize *ETT*, Electricity and Weather datasets for both short-term (24 and 48 horizons) and long-term (96 to 720 horizons) forecasting evaluations. These datasets are commonly used benchmarks for time series forecasting.

Anomaly Detection: We employ two benchmark datasets: KPI (Ren et al., 2019), a competition dataset with minutely sampled KPI curves, and Yahoo (Nikolay Laptev, 2015), containing 367 hourly sampled time series. These datasets represent real-world scenarios where anomaly detection is crucial.

Transfer Learning: We use datasets from the *UCR* archive (Dau et al., 2019), including *CBF* and *CinCECGTorso* as source domains, and evaluate on multiple target domains. This allows us to assess the ability of MERIT to generalize to new, unseen datasets.

C Baseline Descriptions

DTW (Müller, 2007): Dynamic Time Warping (DTW) is a classical algorithm for measuring similarity between two temporal sequences that may vary in speed.

TimesURL (Liu and Chen, 2024b): A self-supervised contrastive learning framework for universal time series representation learning.

TS-TCC (Eldele et al., 2021): A self-supervised framework that learns representations by contrasting temporal contexts.

TS2Vec (Yue et al., 2022): A universal framework for learning representations of time series in an arbitrary semantic level.

T-Loss (Franceschi et al., 2019b): A self-supervised method that learns representations by predicting future values.

InfoTS (Luo et al., 2023): A self-supervised method that learns representations by maximizing the mutual information between different augmented views of a time series.

TST (Zerveas et al., 2021): A Transformer-based model for self-supervised learning of time series representations.

DTW (Müller, 2007): A classical algorithm for measuring similarity between two temporal sequences that may vary in speed.

TS-TCC (Eldele et al., 2021): A self-supervised framework that learns representations by contrasting temporal contexts.

CoST (Woo et al., 2022): A self-supervised method that learns disentangled seasonal and trend representations.

Informer (Zhou et al., 2021): A transformer-based model for long sequence time series forecasting.

LogTrans (Li et al., 2019): A transformer with log-sparse attention for time series forecasting.

N-BEATS (Oreshkin et al., 2019): A deep neural architecture based on backward and forward residual links.

DONUT (Xu et al., 2018): A VAE-based model for unsupervised anomaly detection.

SR (Ren et al., 2019): A spectral residual-based anomaly detection method.

TNC (Tonekaboni et al., 2021): A temporal neighborhood coding approach for representation learning.

D Evaluation Metrics

Classification: For classification tasks, we use accuracy (ACC) and macro-averaged F1-score (F1) as the evaluation metrics. ACC measures the overall prediction correctness, while F1 considers both precision and recall, which is particularly important for imbalanced datasets.

Forecasting: For forecasting tasks, we adopt two widely-used metrics:

- Mean Squared Error (MSE): MSE $= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$,
- Mean Absolute Error (MAE): MAE $= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$,

where y_i and $\hat{y}_i$ denote the ground truth and predicted values, respectively.

Anomaly Detection: For anomaly detection tasks, we use the following metrics:

- Area Under the Receiver Operating Characteristic curve (AUROC)
- Area Under the Precision-Recall curve (AUPRC)
- F1-score at the best threshold (Best-F1)

Transfer Learning: For transfer learning tasks, we evaluate the performance using accuracy (ACC) on the target domains. We also report the relative performance degradation compared to the source domain performance to assess the transfer efficiency.

E Implementation Details

Retrieval Agent: This agent consists of two components: a trainable encoder (Fraikin et al., 2023) and a pre-trained LLM (LLaMA3.1-8B). This agent uses an encoder to extract initial representations for coarse candidate selection. The agent first selects the top $K = 5$ most similar sequences as candidates based on cosine similarity between their TCN representations. Then, it employs the LLM to select $M = 3$ semantically relevant sequences from the candidates as the retrieved contexts.

Augmentation Agent: This agent utilizes a pre-trained LLM (LLaMA3.1-8B) to dynamically select augmentation strategies based on the retrieved contexts and the target sequence. The strategy library $\mathcal{S}$ includes: {Sailing, Resizing, Jittering, Flipping, Permutation, Time Masking, Frequency Masking, Time Neighboring}.

Review Agent: This agent employs a pre-trained LLM (LLaMA3.1-8B) to evaluate the quality of augmented samples. It assesses whether the augmented samples maintain semantic consistency with the original sequence. The agent approves high-quality augmented samples and stores them in a memory bank.

Other Details: We use a trainable encoder (Fraikin et al., 2023) to map the original time series and its augmented versions to a representation space. The encoder has the same architecture as the one used in the retrieval agent. This encoder is trained using the contrastive loss function. For training the encoder, we use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.001 and weight decay of 0.0005. The batch size is set to 128 for all experiments. We train the model for 100 epochs on all datasets. All experiments are conducted using 5 different random seeds, and we report the mean of the results. For all downstream tasks, we use the same train/validation/test splits as in (Yue et al., 2022) to ensure a fair comparison. For all metrics except MSE and MAE, higher values indicate better performance. For MSE and MAE, lower values indicate better performance. Following common practice, we report the mean over multiple runs for all experiments to ensure statistical significance. We use Llama3.1-8b as our base LLM model. The LLM inference is performed on a dedicated GPU server with the following specifications: an NVIDIA A100 GPU (40GB), a 64-core AMD EPYC CPU, 512GB of RAM, and a 2TB NVMe SSD. The average response times are as follows: 27ms per query for the retrieval agent, 78ms per augmentation for the augmentation agent, and 68ms per review for the review agent.

F Analysis of LLM Prompt Design

We experiment with different prompt templates for each agent to analyze their impact on performance. Figure 7 shows the performance comparison of three template variants:

- **Basic**: Simple instruction-based prompts;
- **Structured** (default): Detailed prompts with specific format requirements;
- **CoT**: Chain-of-thought prompts encouraging step-by-step reasoning.

From Figure 7, it can be concluded that (1) Structured prompts (Structured) perform the best in terms of accuracy and manual consistency; (2) CoT prompts, while providing a more detailed reasoning process, have a significantly higher response

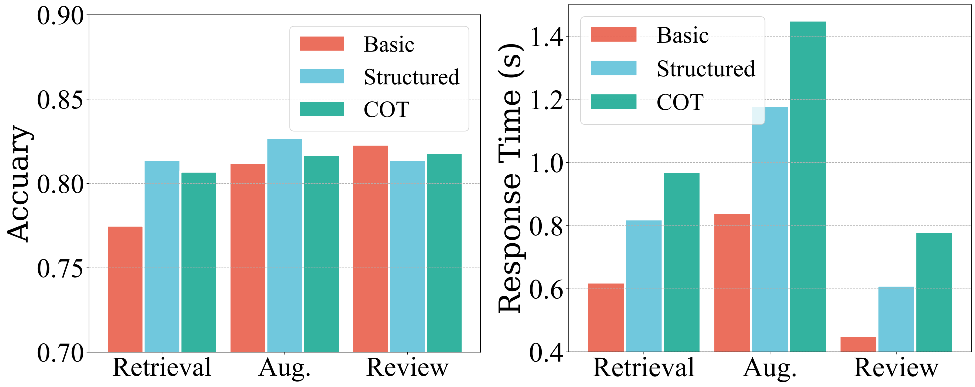


Figure 7: Performance comparison of different prompt templates across agents. Left: Accuracy of different prompt templates. Right: Response time of different prompt templates.

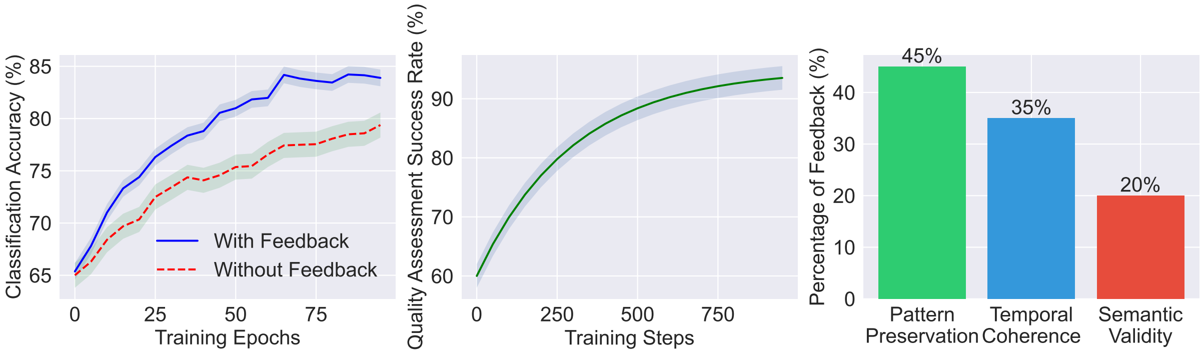


Figure 8: Analysis of feedback loop mechanism. (Left) Performance comparison between with and without feedback loop on *UCR* classification task over training epochs. (Mid) Quality assessment success rate of augmented samples. (Right) Distribution of feedback types from the Review Agent. The shaded areas in (left) and (mid) represent the standard deviation over 5 runs.

time; and (3) Basic prompts (Basic) are the fastest to respond, but are less accurate.

G Impact of Feedback Loop

We analyze the effectiveness of the feedback loop mechanism by examining its impact on the retrieval and augmentation agents, focusing on the quality assessment success rate and the feedback distribution. The iterative nature of the feedback loop improves the quality of generated positive views. Figure 8 (Left) shows that compared to the model without the feedback loop, the model with the feedback mechanism achieves 5.2% higher classification accuracy, which indirectly shows the positive effects of our feedback mechanism, but it should be noted that the feedback loop itself does not directly influence the model's convergence, but rather the quality of positive samples. Furthermore, the quality assessment success rate improves from 60% to over 90% during training (Figure 8 (Mid)), demonstrating the review agent's increasing effectiveness in identifying high-quality augmentations, which further proves the positive role of feedback mechanism. The feedback distribution (Figure 8 (Right)) reveals that pattern preservation (45%) and temporal coherence (35%) are the primary concerns, followed by semantic validity (20%), aligning with our design principle of maintaining both structural and semantic integrity. These findings demonstrate the crucial role of the feedback loop in maintaining high-quality augmentations and guiding the continuous improvement of the retrieval and augmentation agents.

H Algorithm

The details of our algorithm are shown in Algorithm 1.

I Prompt Design

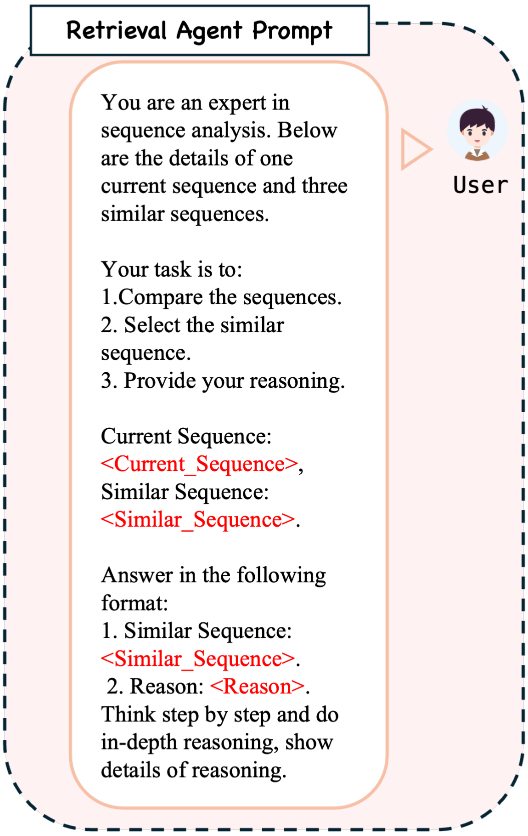
The prompts for our three agents are shown in Figure 9, Figure 10 and Figure 11, respectively.

Algorithm 1 Multi-Agent Collaboration for Unsupervised Time series Representation Learning (MERIT)

```
Input: Dataset $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \in \mathbb{R}^{N \times T \times C}$, Similarity function $\text{Sim}(\cdot,\cdot)$, Parameters $K$, $M$, Augmentation strategy library $\mathcal{S}$, Temperature $\tau$, Maximum iterations $N_{iter}$, Maximum memory bank size $B$
Output: Learned representation function $f_\theta$
  Initialize: Encoder $f_\theta$, Memory bank $\mathcal{M} \leftarrow \emptyset$
for each $\boldsymbol{x}_i \in X$ do
    // Retrieval Agent (executed once)
    $\mathcal{C}_i \leftarrow \text{Top-K}(\text{Sim}(\boldsymbol{x}_i, \boldsymbol{x}_j))$ for all $\boldsymbol{x}_j \in X \setminus \{\boldsymbol{x}_i\}$   ▷ Candidate Selection
    $\mathcal{R}_i \leftarrow \text{Top-M}(\text{LLM-Prompt}(\boldsymbol{x}_i, \boldsymbol{x}_j, \text{Reasoning}))$ for all $\boldsymbol{x}_j \in \mathcal{C}_i$   ▷ LLM-based Refinement
    $n_{iter} \leftarrow 0$
    $\mathcal{M}_i \leftarrow \emptyset$   ▷ Initialize local memory bank for current $\boldsymbol{x}_i$
    while $n_{iter} < N_{iter}$ and $|\mathcal{M}_i| < B$ do
        // Augmentation Agent
        $\mathcal{S}_i \leftarrow \text{LLM-Prompt}(\boldsymbol{x}_i, \mathcal{R}_i, \text{Reasoning})$   ▷ Select Augmentation Strategies
        $X_i^{\text{aug}} \leftarrow \{s(\boldsymbol{x}_i) | s \in \mathcal{S}_i\} \cup \{s(\boldsymbol{x}_j) | s \in \mathcal{S}_i, \boldsymbol{x}_j \in \mathcal{R}_i\}$   ▷ Generate Augmented Samples
        // Review Agent
        $X^{\text{temp}} \leftarrow \emptyset$
        for each $\boldsymbol{x}^{\text{aug}} \in X_i^{\text{aug}}$ do
            $\text{Quality}(\boldsymbol{x}^{\text{aug}}) \leftarrow \text{LLM-Prompt}(\boldsymbol{x}_i, \boldsymbol{x}^{\text{aug}}, \text{Reasoning})$   ▷ Assess Quality
            if $\text{Quality}(\boldsymbol{x}^{\text{aug}}) = \text{'Correct'}$ then
                $X^{\text{temp}} \leftarrow X^{\text{temp}} \cup \{\boldsymbol{x}^{\text{aug}}\}$
            end if
        end for
        if $X^{\text{temp}} \neq \emptyset$ then
            $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup X^{\text{temp}}$   ▷ Store approved augmentations in local memory bank
        end if
        if all $\boldsymbol{x}^{\text{aug}} \in X_i^{\text{aug}}$ are rejected then
            $n_{iter} \leftarrow n_{iter} + 1$
        else
            $n_{iter} \leftarrow N_{iter}$   ▷ Terminate if no augmentation is rejected
        end if
    end while
    $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{M}_i$   ▷ Add approved augmentations to global memory bank
end for
// Time Series Representation Learning
for each epoch do
    for each $\boldsymbol{x}_i \in X$ do
        $\hat{X}_i^P \leftarrow \{\boldsymbol{x}_j | \boldsymbol{x}_j \in \mathcal{R}_i\} \cup \{\boldsymbol{x}^{\text{aug}} | \boldsymbol{x}^{\text{aug}} \in \mathcal{M}_i\}$   ▷ Construct positive samples
        Sample a mini-batch $B$ from $\hat{X}_i^P$
        for each $\boldsymbol{x}^{\text{aug}} \in B$ do
            $\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i)$
            $\boldsymbol{z}^{\text{aug}} \leftarrow f_\theta(\boldsymbol{x}^{\text{aug}})$
            Update $f_\theta$ by minimizing the contrastive loss $\mathcal{L}$ in Eq. (12)
        end for
    end for
end for
```

J Detailed Classification Results

The detailed classification results are shown in Table 7.



Retrieval Agent Prompt

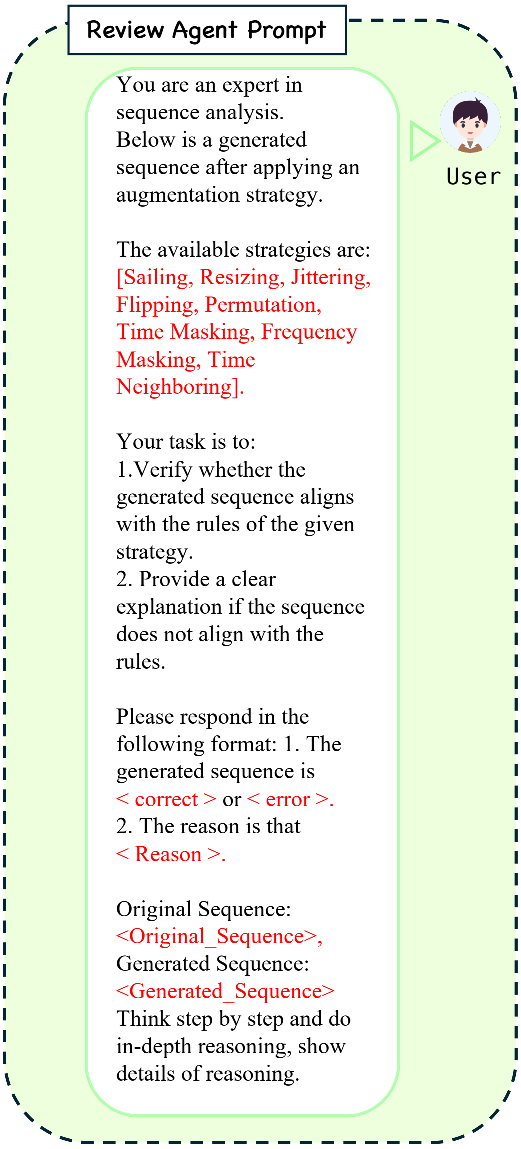
You are an expert in sequence analysis. Below are the details of one current sequence and three similar sequences.

Your task is to:
1.Compare the sequences.
2. Select the similar sequence.
3. Provide your reasoning.

Current Sequence: <Current_Sequence>,
Similar Sequence: <Similar_Sequence>.

Answer in the following format:
1. Similar Sequence: <Similar_Sequence>.
2. Reason: <Reason>.
Think step by step and do in-depth reasoning, show details of reasoning.

Figure 9: The prompt for our retrieval agent.


Review Agent Prompt
User

You are an expert in sequence analysis. Below is a generated sequence after applying an augmentation strategy.

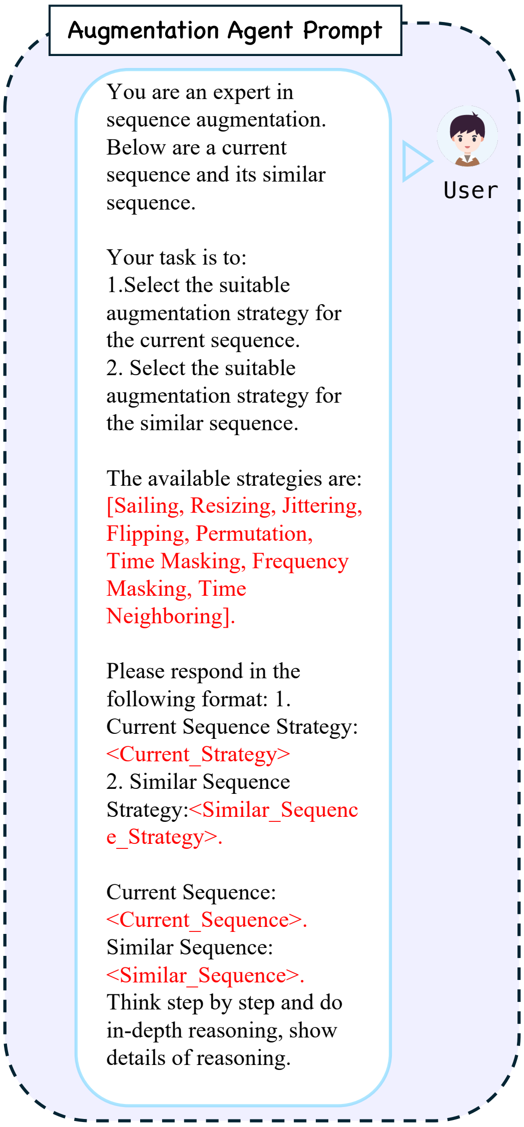
The available strategies are: [Sailing, Resizing, Jittering, Flipping, Permutation, Time Masking, Frequency Masking, Time Neighboring].

Your task is to:
1.Verify whether the generated sequence aligns with the rules of the given strategy.
2. Provide a clear explanation if the sequence does not align with the rules.

Please respond in the following format: 1. The generated sequence is < correct > or < error >.
2. The reason is that < Reason >.

Original Sequence: <Original_Sequence>,
Generated Sequence: <Generated_Sequence>
Think step by step and do in-depth reasoning, show details of reasoning.

Figure 10: The prompt for our review agent.


Augmentation Agent Prompt
User

You are an expert in sequence augmentation. Below are a current sequence and its similar sequence.

Your task is to:
1.Select the suitable augmentation strategy for the current sequence.
2. Select the suitable augmentation strategy for the similar sequence.

The available strategies are: [Sailing, Resizing, Jittering, Flipping, Permutation, Time Masking, Frequency Masking, Time Neighboring].

Please respond in the following format: 1. Current Sequence Strategy: <Current_Strategy>
2. Similar Sequence Strategy:<Similar_Sequence_Strategy>.

Current Sequence: <Current_Sequence>.
Similar Sequence: <Similar_Sequence>.
Think step by step and do in-depth reasoning, show details of reasoning.

Figure 11: The prompt for our augmentation agent.

Dataset	**DTW**	**TST**	**TS-TCC**	**T-Loss**	**TS2Vec**	**TimesURL**	**MERIT**
ArticularyWordRecognition	98.0	98.3	98.7	98.5	98.9	99.0	**99.3**
AtrialFibrillation	22.0	26.7	26.7	26.7	26.7	26.7	**33.3**
BasicMotions	97.5	97.5	**100.0**	97.5	**100.0**	**100.0**	**100.0**
CharacterTrajectories	96.9	97.3	97.8	97.6	98.1	98.4	**98.7**
Cricket	94.4	95.8	97.2	97.2	97.2	98.6	**100.0**
DuckDuckGeese	27.5	35.0	40.0	37.5	42.5	45.0	**47.5**
EigenWorms	55.7	61.1	61.8	61.1	62.6	64.1	**65.6**
Epilepsy	96.4	97.1	97.8	97.5	98.2	98.6	**99.3**
EthanolConcentration	29.3	32.3	33.1	32.7	33.8	34.2	**35.0**
ERing	13.3	13.3	13.3	13.3	13.3	13.3	**13.3**
FaceDetection	52.8	54.7	55.9	55.3	56.5	57.2	**58.4**
FingerMovements	55.0	56.7	58.3	58.3	60.0	61.7	**63.3**
HandMovementDirection	27.8	30.6	33.3	30.6	33.3	36.1	**38.9**
Handwriting	28.5	31.5	32.3	31.9	33.1	34.6	**35.4**
Heartbeat	71.7	73.3	74.2	73.8	75.0	75.8	**76.7**
InsectWingbeat	11.5	12.8	13.6	13.2	14.0	14.8	**15.6**
JapaneseVowels	95.7	96.2	97.0	96.8	97.3	97.8	**98.4**
Libras	83.3	85.0	86.7	86.1	87.2	88.3	**89.4**
LSST	45.3	47.8	49.1	48.4	50.0	51.6	**52.8**
MotorImagery	39.0	42.0	44.0	43.0	45.0	46.0	**48.0**
NATOPS	88.3	90.0	91.7	90.8	92.5	93.3	**94.2**
PenDigits	97.5	97.9	98.3	98.1	98.5	98.8	**99.2**
PEMS-SF	71.0	73.4	74.8	74.1	75.5	76.9	**78.3**
PhonemeSpectra	10.1	11.8	13.4	12.6	14.3	15.1	**16.8**
RacketSports	86.8	88.2	89.5	88.8	90.1	90.8	**91.4**
SelfRegulationSCP1	77.4	79.2	80.2	79.7	81.1	82.1	**83.0**
SelfRegulationSCP2	48.3	50.6	52.2	51.7	53.3	54.4	**55.6**
SpokenArabicDigits	96.7	97.3	97.8	97.5	98.0	98.3	**98.7**
StandWalkJump	33.3	36.7	40.0	36.7	40.0	43.3	**46.7**
UWaveGestureLibrary	85.6	87.2	88.4	87.8	89.1	90.3	**91.5**
Average	62.9	61.7	67.0	65.8	70.4	75.2	**77.2**

Table 7: Classification accuracy (%) on 30 *UEA* datasets. The best results are highlighted in bold.