

DualTime: A Dual-Adapter Multimodal Language Model for Time Series Representation

Weiqi Zhang^{1,2*} Jixia Ye^{2*} Ziyue Li^{3†} Jia Li^{1,2†} Fugee Tsung^{1,2}

¹Hong Kong University of Science and Technology, Hong Kong SAR, China

² Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

³University of Cologne, Germany

wzhangcd@connect.ust.hk, jye324@connect.hkust-gz.edu.cn

zlibn@wiso.uni-koeln.de, {jialee, season}@ust.hk

Abstract

The recent rapid development of language models (LMs) has attracted attention in the field of time series, including multimodal time series modeling. However, we note that current time series multimodal methods are biased, often assigning a primary role to one modality while the other assumes a secondary role. They overlook the mutual benefits and complementary of different modalities. For example, in seizure diagnosis, relying solely on textual clinical reports makes it difficult to pinpoint the area and type of the disease, while electroencephalograms (EEGs) alone cannot provide an accurate diagnosis without considering the symptoms. In this study, based on the complementary information mining of time series multimodal data, we propose **DualTime**, a **Dual**-adapter multimodal language model for **Time** series representation implementing temporal-primary and textual-primary modeling simultaneously. By injecting lightweight adaption tokens, the LM pipeline shared by dual adapters encourages embedding alignment and achieves efficient fine-tuning. Empirically, our method outperforms state-of-the-art models in both supervised and unsupervised settings, highlighting the complementary benefits of different modalities. In addition, we conduct few-shot label transfer experiments, which further verifies the transferability and expressiveness of our proposed DualTime.

1 Introduction

As a data modality prevalent in practical applications, modeling time series has always been an essential and ongoing challenge [28, 35, 43]. The rapid advancements in language models (LMs) have inspired many time series studies to incorporate them into the time series modeling strategies. These models, cultivated from extensive training corpora, exhibit remarkable proficiency in understanding and generating sequential data. Similar to the multimodal language models in areas like computer vision and audio processing, as exemplified by BLIP-2 [11] and VALL-E [24], recent work has begun to focus on integrating time series data with other modalities (e.g., textual data) to enhance time series analysis [14, 7, 1, 32, 12].

Overall, we can categorize the existing multimodal time series methods into the following two paradigms: temporal-primary multimodal model, and textual-primary multimodal model (as shown in Figure 1(a)), with the former being more commonly adopted. For instance, UniTime [14] offers domain instructions to help the model distinguish different sources of datasets and adjust the time

*Equal contributions.

†The corresponding authors.

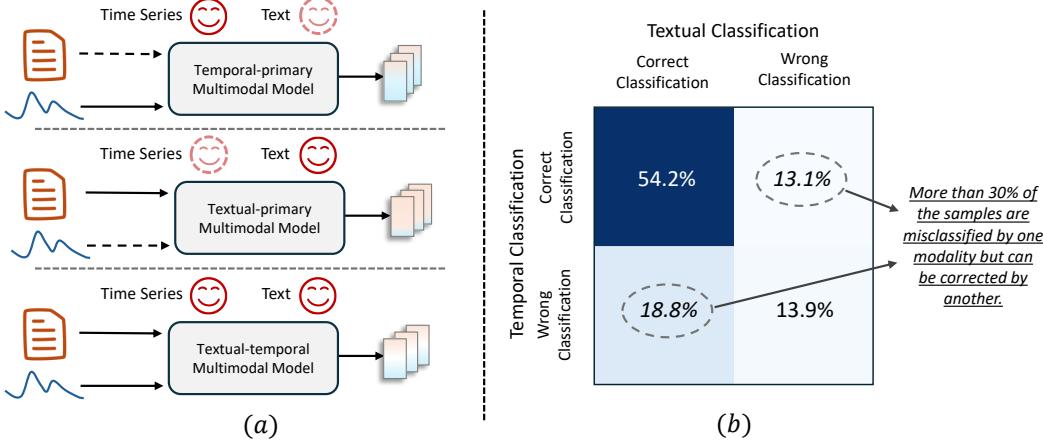


Figure 1: (a) Comparisons between different time series multimodal modeling paradigms. The solid line represents the primary modality and the dashed line is for the secondary modality. Examples of temporal-primary multimodal models: UniTime [14], TimeLLM [7]. Examples of textual-primary multimodal models: [32]. (b) Unimodal classification results on PTB-XL datasets. The circled samples are misclassified by one modality but can be corrected by another, which demonstrates the complementary information of different modalities.

series modeling strategy accordingly. Similarly, TimeLLM [7] assembles dataset descriptions, task instructions, and data statistics to enhance time series embedding. In the aforementioned work, the textual input consists of dataset-level information, such as dataset descriptions, which makes it difficult to provide information and discernibility at the sample level. Consequently, in temporal-primary multimodal models, time series plays a primary role with the textual input as a secondary modality. In contrast, in [32], time series data (i.e., stock return) has been reformulated as textual tokens for enhancing the modeling of primary textual input, thereby establishing a textual-primary multimodal modeling framework.

However, in some cases, especially when each time series sample has corresponding texts, the sample-level temporal and textual inputs are equally important and can mutually benefit each other. This is because the strongly coupled multimodal data often display different types of information. For instance, in medical applications, time series modalities such as electrocardiograms (ECGs) and electroencephalograms (EEGs) capture the physiological electrical activities of patients, while textual modalities like clinical records and laboratory reports offer insights into historical health conditions or symptoms. Analyzing symptoms and medical history alone allows doctors to infer that a patient might have epilepsy, but it is challenging to specify the type and area of the seizures. Conversely, EEG data can detect abnormal bursts of electrical activity [9]. However, considering individual differences and the lack of patient background, the diagnosis may not be entirely accurate. Thus, integrating different modalities during diagnostics can lead to more precise and rational judgments.

Empirically, when conducting vanilla classification on unimodal data, we encounter numerous (about 30%) samples that are misclassified by one modality but can be correctly classified by another (Figure 1(b)). It substantiates the non-overlapping information embedded in different modalities and reveals the necessity of textual-temporal multimodal modeling. Thus, the simultaneously textual-temporal multimodal model (as shown at the bottom of Figure 1(a)) is a more promising solution for leveraging the advantages of complementary multimodal data.

On the other hand, constructing an effective textual-temporal multimodal model is technically non-trivial. The most straightforward solution is to train a temporal-primary multimodal model and a textual-primary multimodal model separately, and then combine these two submodels. Nevertheless, there remain three-fold challenges: First, considering the LMs involved, two separately trained multimodal models suffer non-negligible computational costs. Second, the alignment of two submodels is hard to guarantee, which may impact the submodel combination performance. Additionally, most existing time series multimodal design mainly focuses on concatenating different modalities together as LM’s input. Taking full advantage of pre-trained language models and implementing multimodal fusion in the intermediate layers remains a challenge.

To address the aforementioned challenges, we propose DualTime, a multimodal language model for time series representation learning, consisting of a temporal-primary multimodal adapter and a textual-primary multimodal adapter, to effectively explore the complementary information of multimodal input. Based on our dual adapter design, each modality has the chance to serve as the primary modality and gets improved through the fusion of the other modality. Furthermore, dual adapters within our proposed structure share the frozen LM backbone parameters to make different modalities benefit from language model pre-training and lead to efficient model fine-tuning. Meanwhile, by pipeline sharing, the alignment of different adapters could be accomplished. Within each adapter, we perform multimodal fusion by injecting learnable adaptation tokens into the intermediate layers, rather than simply concatenating the multimodal inputs. Overall, our main contributions can be summarized as below:

- We discuss the existing multimodal methods from the perspective of modality fusion paradigms for the first time. Instead of using one modality to serve another, we propose a novel multimodal representation learning framework, which is designed to facilitate the mutual complementarity of coupled text and time series multimodal data.
- We propose **DualTime**, a dual-adapter multimodal language model for time series representation learning by introducing learnable tokens to perform the mutual injection of text and time series multimodal data. Additionally, adaption-based multimodal fusion allows dual adapters to share the pre-trained parameters of LMs, taking advantage of the sequential modeling ability of LMs and achieving more efficient fine-tuning.
- Through extensive experiments, **DualTime** achieves superior empirical performance on real-world datasets. Its expressive multimodal representation results in significant improvements in both supervised and unsupervised learning tasks. Furthermore, label transfer experiments showcase the model’s transferability and few-shot generalization capabilities, empowered by language models.

2 Methodology

In this work, we focus on sample-level time series multimodal data. Specifically, each sample is a time-text pair (e.g., ECG signal and its coupled clinical report). The whole dataset is denoted as $\mathcal{S} = \{(\mathbf{X}_1, \mathbf{S}_1), (\mathbf{X}_2, \mathbf{S}_2), \dots, (\mathbf{X}_N, \mathbf{S}_N)\}$, where $\mathbf{X}_i \in \mathbb{R}^{T \times d}$ denotes a d -dimension multivariate time series modality with length T and \mathbf{S}_i denotes the paired textual modality. For simplicity, we omit the sample indicator subscript in the following.

In summary, to fully utilize the complementary information of different modalities, DualTime consists of two multimodal adapters, namely a textual-primary multimodal adapter, and a temporal-primary multimodal adapter. Each adapter treats one modality as the primary modality and enhances it with the other modality. Both adapters share the same frozen pre-trained language model with L layers. Each adapter implements multimodal fusion in the topmost M ($M \leq L$) transformer blocks of the language model. The shared language model backbone facilitates efficient fine-tuning and encourages the dual adapters’ embedding space alignment.

2.1 Textual-primary Multimodal Adapter

Processed by the textual tokenizer, the text input can be modeled by I^s -length word tokens with embedding $\mathbf{E}_s \in \mathbb{R}^{I^s \times D}$, where D is the hidden dimension. For the first $L - M$ transformer layers, the forward process of layer- l is:

$$\tilde{\mathbf{H}}_s^{l-1} = \text{LN}(\text{MHA}(\mathbf{W}_q^l \mathbf{H}_s^{l-1}, \mathbf{W}_k^l \mathbf{H}_s^{l-1}, \mathbf{W}_v^l \mathbf{H}_s^{l-1})) + \mathbf{H}_s^{l-1}, \quad (1)$$

$$\mathbf{H}_s^l = \text{LN}(\text{MLP}(\tilde{\mathbf{H}}_s^{l-1})) + \tilde{\mathbf{H}}_s^{l-1}, \quad (2)$$

where \mathbf{H}_s^l is the output of layer- l with $\mathbf{H}_s^0 = \mathbf{E}_s$, MHA, LN, MLP denote the multi-head attention, the layer normalization, and the multi-layer perception, respectively. To obtain the query, key, value matrices at layer- l , $\mathbf{W}_q^l, \mathbf{W}_k^l, \mathbf{W}_v^l$ are parameterized by the pre-trained language model. Meanwhile, the attention operation Attention is defined by:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\mathbf{Q}\mathbf{K}^T / \sqrt{d_k}\right) \mathbf{V}, \quad (3)$$

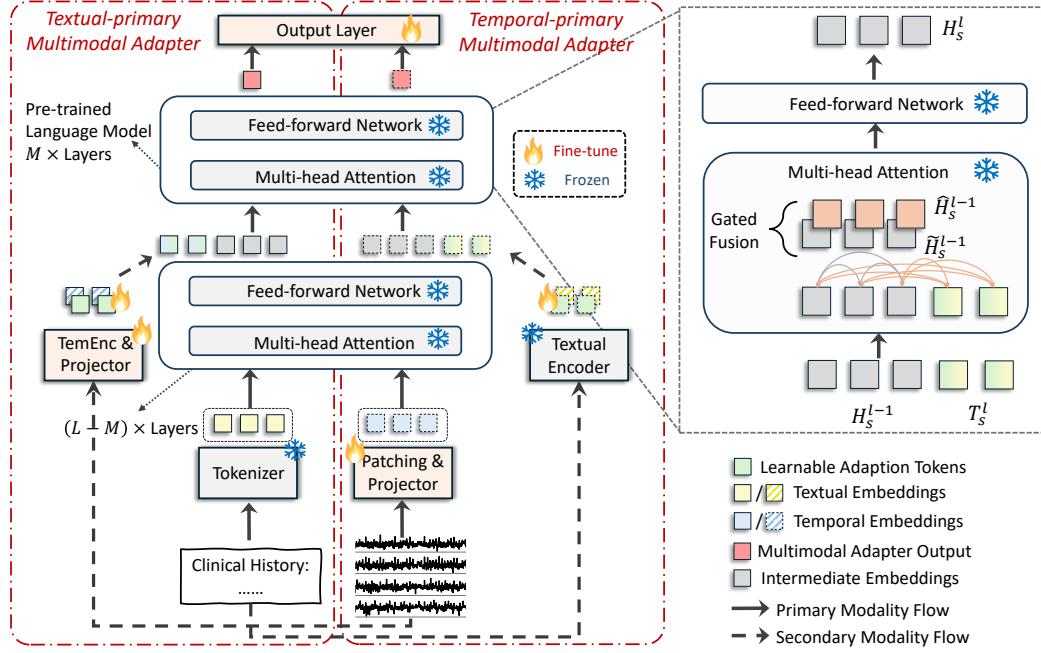


Figure 2: DualTime architecture. It consists of dual adapters to model time series and text as primary modality respectively, i.e. a textual-primary multimodal adapter and a temporal-primary multimodal adapter. Dual adapters share the same LM parameters to facilitate efficient fine-tuning and preserve LM’s pre-trained knowledge by adopting a zero-initialized gating strategy. Each adapter injects trainable adaption tokens in the intermediate layers to achieve multimodal fusion.

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are corresponding query, key, and value matrices, d_k is the dimension of key.

Furthermore, we utilize a lightweight adapter mechanism to achieve multimodal modeling at the topmost M transformer blocks. Specifically, we adopt learnable length- P adaption tokens \mathbf{T}_s^l at each multimodal fusion layer l ($L - M + 1 \leq l \leq L$), where the adaption tokens $\mathbf{T}_s^l \in \mathbb{R}^{P \times D}$ have the same dimension as language model. As to the secondary temporal modality, a trainable temporal encoder and a cross-modal projector are utilized to transform the time series input into the language model embedding space:

$$\mathbf{Z}_s = \text{Proj}(\text{TemEnc}(\mathbf{X})). \quad (4)$$

For decreasing the computational cost, different multimodal fusion layers will share the same temporal embedding. Thus, the adaption tokens of textual-primary multimodal adapter will be calculated by:

$$\tilde{\mathbf{T}}_s^l = \mathbf{T}_s^l + \mathbf{Z}_s. \quad (5)$$

For the topmost M transformer layers, the multimodal forward process is formalized as:

$$\tilde{\mathbf{H}}_s^{l-1} = \text{LN}(\text{MHA}(\mathbf{W}_q^l \mathbf{H}_s^{l-1}, \mathbf{W}_k^l \mathbf{H}_s^{l-1}, \mathbf{W}_v^l \mathbf{H}_s^{l-1})) + \mathbf{H}_s^{l-1}, \quad (6)$$

$$\hat{\mathbf{H}}_s^{l-1} = \text{LN}(\text{MHA}(\mathbf{W}_q^l \mathbf{H}_s^{l-1}, \mathbf{W}_k^l \tilde{\mathbf{T}}_s^l, \mathbf{W}_v^l \tilde{\mathbf{T}}_s^l)) + \mathbf{H}_s^{l-1}, \quad (7)$$

$$\mathbf{H}_s^l = \text{LN}(\text{MLP}(\text{Gate}^l \hat{\mathbf{H}}_s^{l-1} + \tilde{\mathbf{H}}_s^{l-1})) + (\text{Gate}^l \hat{\mathbf{H}}_s^{l-1} + \tilde{\mathbf{H}}_s^{l-1}). \quad (8)$$

In particular, combined with the pre-trained projection matrices $\mathbf{W}_k^l, \mathbf{W}_v^l$, the learnable adaption tokens will serve as key, value matrices of the multi-head attention layer. In Equation (8), we perform a zero-initialized gating strategy to achieve multimodal adaption token fusion [36]. Gating parameter Gate^l will be initialized as zero at the beginning of training, the multimodal adaption tokens will be injected gradually, which can preserve the pre-trained knowledge and capacities of LMs.

2.2 Temporal-primary Multimodal Adapter

Considering the sequential property of time series, the temporal-primary multimodal adapter takes the time series data as the language model input. Several adjacent timestamps will be assembled as a token, which can provide local semantic information within a time series. For a pre-defined patch size p and stride s , the time series input $\mathbf{X} \in \mathbb{R}^{T \times d}$ can be reorganized as $\tilde{\mathbf{X}} \in \mathbb{R}^{T_s \times (p \times d)}$, where $T_s = \left\lceil \frac{T-p}{s} \right\rceil + 1$ is the number of temporal tokens. Subsequently, we utilize a projector to adjust the dimension of temporal tokens. The adjusted temporal token can be denoted as \mathbf{E}_t ($\mathbf{E}_t \in \mathbb{R}^{T_s \times D}$).

With $\mathbf{H}_t^0 = \mathbf{E}_t$ as the input of the first transformer layer, the model forward process will be similar to the ones introduced in Section 2.1, e.g., Equation (1-2) and Equation (5 - 8).

Differently, for the secondary text input, we use a pre-trained BERT [3] model as a text encoder (similar to the temporal encoder in Equation (4)) to extract textual information:

$$\mathbf{Z}_t = \text{Proj}(\text{BERT}(\mathbf{S})). \quad (9)$$

2.3 Pre-trained Language Model Parameters Sharing

Aided by our dual adapter model design, most of the pre-trained language model parameters (e.g., the attention weight matrices $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$, and the MLP layer of each transformer block) could be shared by both textual-primary multimodal adapter and temporal-primary multimodal adapter. On the one hand, the frozen parameters could preserve the knowledge and sequential modeling capacities of the language model. On the other hand, since most of the parameters in our proposed adapters are shared, there is only a minimal increase in the training parameters compared to a single adapter. This ensures complementary modeling between the two modalities while still allowing for efficient fine-tuning. Additionally, by sharing the same LM pipeline, the embedding spaces of different adapters are easily aligned, further facilitating the integration of dual adapters.

2.4 Training Loss

Supervised Learning. For supervised classification, we add the last transformer layer output of each adapter together to obtain the final multimodal representation. Then, an extra linear classifier and the cross-entropy loss are used for supervised training.

Unsupervised Representation Learning. For unsupervised representation learning, we adopt the contrastive learning paradigm. In particular, by random dropout, we first create the augmentation as positive pairs within each adapter. For example, \mathbf{H}'_s^L denotes the augmentation of \mathbf{H}_s^L , and \mathbf{H}'_t^L denotes the augmentation of \mathbf{H}_t^L . The contrastive loss could be divided into two parts, within-adapter contrastive loss and cross-adapter contrastive loss.

Formally, by maximizing the agreement between positive pairs and minimizing the similarity between negative pairs (i.e., different input instances), in a mini-batch with size B , the within-adapter contrastive losses are

$$\mathcal{L}_s = - \sum_{i=1}^B \log \frac{\exp(\text{sim}(\mathbf{H}_{s,i}^L, \mathbf{H}'_{s,i}^L)/\tau)}{\sum_{k=1}^B \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{H}_{s,i}^L, \mathbf{H}_{s,k}^L)/\tau)}, \quad \mathcal{L}_t = - \sum_{i=1}^B \log \frac{\exp(\text{sim}(\mathbf{H}_{t,i}^L, \mathbf{H}'_{t,i}^L)/\tau)}{\sum_{k=1}^B \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{H}_{t,i}^L, \mathbf{H}_{t,k}^L)/\tau)}, \quad (10)$$

where $\mathbb{1}_{[k \neq i]}$ is the indicator function and τ is the temperature parameter, $\text{sim}(\cdot, \cdot)$ is the dot product between two ℓ_2 -normalized vectors.

The cross-adapter contrastive learning assumes that the embeddings from two adapters for one temporal-textual input pair should be similar. Concurrently, embedding from different instances should be considered negative pairs. In this vein, the cross-adapter contrastive loss is given by:

$$\mathcal{L}_{cross} = - \sum_{i=1}^B \left(\log \frac{\exp(\text{sim}(\mathbf{H}_{s,i}^L, \mathbf{H}_{t,i}^L)/\tau)}{\sum_{k=1}^B \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{H}_{s,i}^L, \mathbf{H}_{t,k}^L)/\tau)} + \log \frac{\exp(\text{sim}(\mathbf{H}_{t,i}^L, \mathbf{H}_{s,i}^L)/\tau)}{\sum_{k=1}^B \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{H}_{t,i}^L, \mathbf{H}_{s,k}^L)/\tau)} \right). \quad (11)$$

The overall loss function of unsupervised representation learning is given by:

$$\mathcal{L}_{unsup} = \mathcal{L}_s + \mathcal{L}_t + \mathcal{L}_{cross}. \quad (12)$$

3 Experiments

Table 1: **Supervised Learning of disease detection and classification.** DualTime achieves an average improvement of **7%** in Acc. and **15%** in F1 across all experiments in PTB-XL and TUSZ datasets. The best results are in **bold** while the second and third best are in underlined. "Acc.", "Pre.", and "Rec." represent accuracy, precision and recall respectively. All LM-based Models are highlighted in grey.

Modality	Model	PTB-XL								TUSZ								Average		
		Detection				Classification				Detection				Classification				Acc.	F1	
		Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1			
LM-free Model	Time	LSTM	0.68	0.60	0.48	0.48	0.67	0.63	0.50	0.52	0.76	0.53	0.54	0.54	0.58	0.44	0.27	0.26	0.67	0.45
		TimesNet	0.68	0.46	0.46	0.45	0.67	0.59	0.48	0.50	0.74	0.59	0.63	0.59	0.76	0.75	0.72	0.71	0.71	0.56
		LightTS	0.68	0.59	0.53	0.54	0.59	0.46	0.44	0.45	0.74	0.53	0.53	0.54	0.71	0.72	0.58	0.58	0.68	0.53
		DLinear	0.68	0.58	0.50	0.49	0.61	0.46	0.41	0.41	0.78	0.52	0.52	0.52	0.71	0.62	0.60	0.59	0.70	0.50
		Pyraformer	0.76	0.66	0.59	0.58	0.66	0.56	0.49	0.51	0.84	0.47	0.50	0.47	0.75	0.77	0.67	0.72	0.75	0.57
		ETSformer	0.72	0.63	0.57	0.55	0.54	0.45	0.38	0.40	0.79	0.53	0.53	0.53	0.73	0.70	0.66	0.66	0.70	0.54
		Autoformer	0.72	0.56	0.56	0.54	0.62	0.47	0.44	0.44	0.79	0.52	0.51	0.51	0.70	0.64	0.64	0.61	0.71	0.53
		Crossformer	0.68	0.58	0.51	0.53	0.65	0.55	0.48	0.50	0.79	0.50	0.51	0.50	0.72	0.71	0.58	0.58	0.71	0.53
		FEDformer	0.67	0.57	0.50	0.51	0.65	0.53	0.47	0.49	0.76	0.57	0.58	0.57	0.68	0.48	0.54	0.48	0.69	0.51
		Informer	0.67	0.59	0.51	0.52	0.67	0.59	0.51	0.52	0.82	0.57	0.55	0.55	0.77	0.74	0.69	0.71	0.73	0.58
		Reformer	0.69	0.56	0.53	0.54	0.65	0.53	0.48	0.49	0.84	0.52	0.50	0.48	0.74	0.75	0.61	0.66	0.73	0.54
		iTransformer	0.56	0.42	0.36	0.37	0.54	0.39	0.31	0.29	0.80	0.50	0.50	0.49	0.73	0.75	0.59	0.61	0.66	0.44
		PatchTST	0.78	0.76	0.62	0.62	0.74	0.69	0.59	0.62	0.73	0.54	0.55	0.54	0.70	0.65	0.59	0.57	0.74	0.59
LM-based Model	Time	GPT4TS	0.71	0.58	0.52	0.53	0.59	0.46	0.45	0.45	0.78	0.48	0.48	0.48	0.71	0.73	0.60	0.64	0.70	0.53
	Text	GPT2	0.72	0.65	0.56	0.58	0.73	0.65	0.61	0.62	0.72	0.49	0.49	0.50	0.64	0.69	0.53	0.58	0.70	0.57
	Text	BERT	0.70	0.64	0.51	0.53	0.73	0.65	0.59	0.62	0.72	0.49	0.49	0.49	0.59	0.45	0.39	0.40	0.69	0.51
	Time + Text	TimeLLM	0.69	0.60	0.48	0.47	0.67	0.59	0.46	0.48	0.75	0.51	0.51	0.51	0.69	0.70	0.50	0.47	0.65	0.41
	Text	DualTime (Time)	0.72	0.61	0.55	0.54	0.68	0.58	0.53	0.53	0.83	0.61	0.57	0.58	0.72	0.74	0.60	0.59	0.74	0.56
	Text	DualTime (Text)	0.82	0.75	0.74	0.74	0.76	0.69	0.63	0.65	0.82	0.65	0.66	0.65	0.78	0.74	0.72	0.73	0.79	0.69
	Text	DualTime	0.83	0.77	0.75	0.76	0.80	0.74	0.73	0.73	0.84	0.69	0.69	0.69	0.79	0.77	0.80	0.78	0.82	0.74

3.1 Experimental Setup

Datasets All the experiments are conducted on two real-world multimodal time series datasets: PTB-XL [23], TUSZ v1.5.2 [19]. PTB-XL contains 12-lead electrocardiograms (ECGs) with paired clinical reports describing signal characteristics without diagnosis labels.

Following [10], all the non-English ECG reports are translated into English. TUSZ is a large-scale EEG seizure database containing 19-channel EEG signals and clinical history for each session of patients. Following [22], we process TUSZ to obtain 60-second EEGs for experiments. To avoid data imbalance, we randomly sample at most 8 normal EEGs per patient for training. Both datasets offer two sets of labels: a coarse-grained label set for disease detection and a fine-grained label set for disease classification. More details of dataset are in Appendix A.

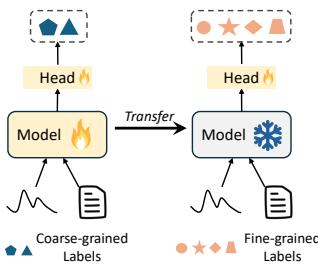


Figure 3: Illustration for **In-dataset Label Transfer**: we first pre-train a model on dataset with coarse-grained but redundant labels (e.g., disease detection), then fine-tune it with fine-grained but limited labels (e.g., disease classification).

Baselines Multiple representative unimodal/multimodal baselines are selected to ensure sufficient experiments. BERT [3] and GPT-2 [18] represent unimodal textual approaches. Unimodal time series baselines include CNN-based models (TimesNet [28], TS2Vec [33], TS-CoT [38]), MLP-based models (LightTS [37], DLinear [34]), RNN-based models (LSTM [6]), Transformer-based models (Pyraformer [13], ETSformer [26], Autoformer [27], Crossformer [39], FEDformer [42], Informer [41], Reformer [8], iTransformer [15], PatchTST [17], TS-TCC [4]), language model based models (GPT4TS [43]). For multimodal baselines, TimeLLM [7] is adopted for supervised learning and METS [10] is for unsupervised learning. We also ablate DualTime into two variants, namely *DualTime (Time)* for temporal-primary multimodal adapter and *DualTime (Text)* for textual-primary multimodal adapter.

Implementations DualTime adopts a frozen GPT-2 [18] as backbone. In textual-primary multimodal adapter, the temporal encoder is trainable, consisting of three conv-blocks and each with three CNN layers. The tokenizer is from GPT-2. In temporal-primary multimodal adapter, a frozen BERT serves as a textual encoder providing embeddings for textual modality. To align with the dimension of GPT-2, all hidden dimensions are set to 768.

The number of multimodal fusion layers M and adaption token length P are 11 and 5. The time series patching size and stride are all 25. We adopt Adam as the optimizer. All experiments are implemented by PyTorch Framework with a NVIDIA A6000 (48G) GPU.

3.2 Evaluation Strategy

In this work, we evaluate our proposed model from three aspects. The first is **Supervised Learning**, where a linear classifier is added as the output layer of our model to verify its high-quality representation learning ability with supervision signals. Second, to discuss the transferability of learned representations, we establish a **Few-shot In-dataset Label Transfer** framework representing the in-dataset transfer between different label sets (as shown in Figure 3). This experimental setting is quite common in real-world applications. Considering that the difficulty of obtaining data labels varies with granularity, coarse-grained labels tend to be easier and less costly to obtain, while fine-grained labels may be more expensive to acquire. We hope to discuss the few-shot transfer capabilities by training models with coarse-grained labels (such as whether there is an illness) and transferring the pre-trained model to more fine-grained labels (such as specific types of diseases). The last setting is **Unsupervised Representation Learning**, which can evaluate our model’s capacity to produce general representations without ground truth supervision. For DualTime (Time) and DualTime (Text), we only adopt the within-adapter contrastive loss for training.

3.3 Supervised Learning

As shown in Table 1, (1) advanced time-only models perform better than text-only models, achieving second best in most experiments. PatchTST significantly outperforms other baselines in PTB-XL. This indicates that time series modality contains more information for decision than text modality. (2) In addition, compared with text-only BERT and GPT-2, DualTime (Text) utilizes text augmented by time and shows noticeable improvement, underscoring the importance of including time series in the textual-primary model. (3) When both modalities are available, it is noteworthy that DualTime (Text) generally outperforms DualTime (Time), likely due to GPT-2’s stronger capability in processing text compared to time series. (4) Additionally, DualTime significantly outperforms TimeLLM, probably because DualTime models sample-level and complementary multimodal while TimeLLM works in a temporal-primary manner. Overall, DualTime is the best by improving accuracy by 7% and F1 by 15% on average.

3.4 Few-shot Learning for In-dataset Label Transfer

For few-shot label transfer, we freeze pre-trained model parameters and fine-tune an additional classifier using limited fine-grained labeled data. Specifically, we pre-train model on disease detection and fine-tune it on disease classification for both PTB-XL and TUSZ. (1) The 5-shot experiment results in Table 2 show that time-only models generally outperform text-only models. The limited 5-shot time series samples might exhibit patterns captured by time-only models while GPT-2 and BERT struggle with few textual samples. (2) Also, DualTime (Time) outperforms DualTime (Text) on PTB-XL and is close to it on TUSZ, indicating that time series modality is more important than text modality when samples are limited. (3) Even with just 5-shot training samples, DualTime performs better than baselines. (4) As shot K increases, DualTime’s advantage in accuracy gradually grows (shown in Figure 4).

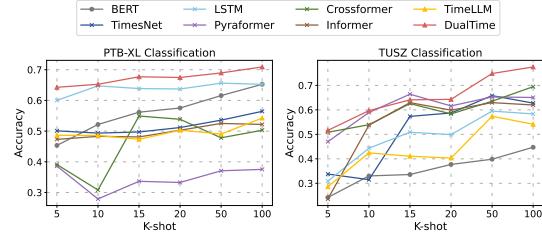


Figure 4: Performance comparison for label transfer with different shots. DualTime shows the best performance on nearly all the shots, For small shots, its advantage is not significant while as the shot increases, the performance gap becomes obvious.

Table 2: **5-shot in-dataset label transfer**. DualTime achieves almost the best fine-tuning performance, demonstrating its superior few-shot transfer capacity due to its adaptive complementary multimodal modeling.

Modality	Model	PTB-XL				TUSZ			
		Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
Time	LSTM	0.60	0.37	0.38	0.37	0.31	0.55	0.48	0.37
	TimesNet	0.50	0.33	0.32	0.29	0.34	0.26	0.21	0.20
	LightTTS	0.22	0.24	0.25	0.20	0.33	0.39	0.44	0.33
	Dlinear	0.30	0.24	0.24	0.23	0.42	0.37	0.48	0.37
	Pyraformer	0.39	0.24	0.23	0.22	0.47	0.33	0.43	0.33
	ETSTformer	0.46	0.33	0.24	0.21	0.44	0.53	0.33	0.32
	Autoformer	0.25	0.26	0.26	0.22	0.24	0.62	0.29	0.17
	Crossformer	0.39	0.32	0.35	0.31	0.51	0.34	0.36	0.35
	FEDformer	0.21	0.23	0.22	0.18	0.34	0.26	0.21	0.20
	Informer	0.47	0.35	0.35	0.34	0.24	0.33	0.21	0.17
Text	Reformer	0.32	0.38	0.27	0.25	0.34	0.30	0.31	0.24
	iTransformer	0.25	0.20	0.20	0.29	0.51	0.41	0.47	0.41
	PatchTST	0.45	0.38	0.40	0.38	0.34	0.21	0.31	0.19
	GPT4TS	0.20	0.20	0.20	0.18	0.45	0.42	0.49	0.38
Text	GPT2	0.24	0.22	0.22	0.18	0.20	0.31	0.44	0.19
	BERT	0.45	0.34	0.33	0.32	0.24	0.35	0.32	0.24
Time + Text	TimeLLM	0.49	0.28	0.33	0.30	0.29	0.33	0.26	0.25
	DualTime (Time)	0.58	0.41	0.39	0.38	0.46	0.41	0.51	0.42
	DualTime (Text)	0.49	0.37	0.38	0.36	0.47	0.45	0.51	0.43
DualTime	DualTime	0.64	0.52	0.50	0.52	0.48	0.56	0.48	

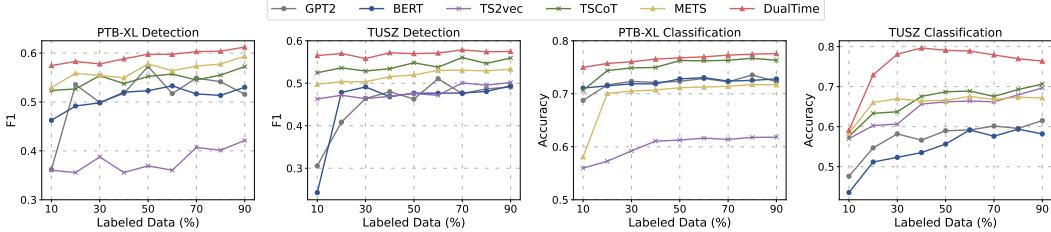


Figure 5: Performance comparison for unsupervised representation learning with different proportions of labeled data. DualTime consistently performs best, especially in TUSZ classification perhaps due to the beneficial seizures history of patients.

indicating the robustness of its transferability and its superiority at handling increasingly complex information. The results of all baselines can be found in the Appendix B.2.

3.5 Unsupervised Representation Learning

Following the training strategy introduced in Section 2.4, we first train the multimodal encoder in an unsupervised manner. After obtaining unsupervised embeddings for all samples, different proportions of them along with labels are utilized to train a linear classifier. Figure 5 shows the performance comparison among competitive unsupervised approaches with data proportions varying from 10% to 90%. Table 3 shows the results of 100% data proportion. More results can be found in Appendix B.2.

- (1) Similar to supervised learning and few-shot learning, time-only models usually have a better performance than text-only models across all unsupervised experiments, underscoring the significance of time series data. (2) Figure 5 shows that multimodal METS performs closely to the best time-only model TSCoT in PTB-XL detection and TUSZ classification, but slightly lags behind in the other two tasks, indicating that multimodality may not always be better than single modality. (3) However, in Table 3, DualTime’s outperformance over TSCoT demonstrates the effectiveness of multimodal modeling. DualTime and its variants outperform METS in most experiments highlighting the superiority of our complementary textual-temporal multimodal design. (4) Overall, DualTime achieves 5% average accuracy gains in Table 3 and consistently exceeds baselines across different proportions of data with stable performance in Figure 5. This indicates that representations learned by DualTime have stronger expressive and transferability, enabling effective training even with limited samples. The full results with all baseline methods can be found in Appendix B.2.

3.6 Ablation Study

We ablate DualTime into DualTime (Time) and DualTime (Text). Specifically, DualTime (Time) leverages the textual modality to enhance the temporal modality modeling, while DualTime (Text) treats the textual modality as primary and the temporal modality as secondary. We evaluate their performances under all three settings, as shown in Table 1, 2, 3. (1) Generally speaking, DualTime (Text) has a better performance than DualTime (Time) in supervised learning and unsupervised learning. This suggests that the backbone language model (i.e., GPT-2) demonstrates a better understanding of text compared with time series. (2) While DualTime (Time) outperforms DualTime (Text) in PTB-XL 5-shot experiments (as shown in Table 2), possibly because the model lacks sufficient understanding of limited textual data and temporal modality can provide more valuable clues for decision-making. (3) Overall, DualTime consistently outperforms single adapter variants, indicating the contributions of both adapters and highlighting the advantages of complementary multimodal modeling over treating one modality as primary and the other as supplementary.

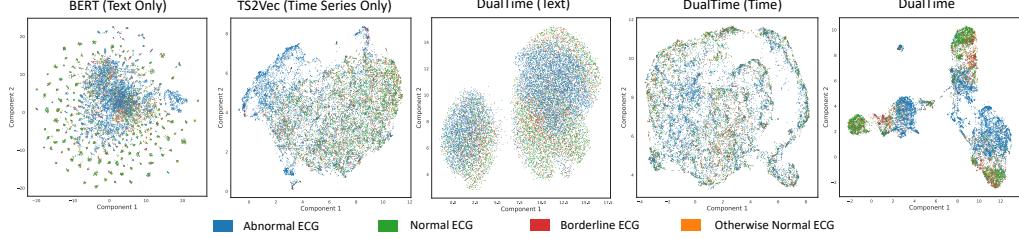


Figure 6: Embedding visualization of different encoders on PTB-XL. Labels are distinguished by colors. Overall, DualTime can distinguish different classes more clearly than others, validating the effectiveness of complementary textual-temporal multimodal modeling.

3.7 Analysis

Visualization To better visualize the learned representations, we adopt UMAP [16] to project the results of unsupervised representation learning into 2-D plots. (1) Figure 6 shows the embeddings of different encoders with corresponding labels on PTB-XL datasets. TS2Vec (time-only) can identify Abnormal ECG while BERT (text-only) shows the worst performance owing to mixing all categories, illustrating the advantage of time series modality. (2) Compared with BERT, DualTime (Text) can better distinguish abnormal ECG and normal ECG, indicating the effectiveness of two modalities over one modality. (3) Compared with DualTime (Time), DualTime (Text) has obviously better discriminative capacity, supporting the advantage of textual-primary multimodal modeling over temporal-primary multimodal modeling. (4) Furthermore, DualTime can provide the most discriminative representations, attributed to the benefit of complementary multimodal modeling.

Efficiency Analysis We analyze the efficiency of representative models by comparing model performance, trainable parameters, and training time per epoch, as shown in Figure 7. (1) Overall, DualTime has a moderate trainable parameter size with best performance. In TUSZ detection, it has about 1.0 million trainable parameters, larger than PatchTST while much smaller than TimesNet. (2) The complexity of TimesNet may be attributed to its 2D convolution operation. While the light-weight trainable parameters of DualTime are beneficial from the frozen setting of backbone model GPT-2. (3) Compared with another multimodal model TimeLLM, DualTime exhibits better training efficiency and model performance.

4 Related Work

Recent LM-based approaches for time series analysis have two classes, i.e. unimodal or multimodal methods [31]. **For unimodal approaches**, like GPT4TS [43] and LLM4TS [2], concentrate on parameter efficient fine-tuning to unlock the LM’s capacities without updating extensive parameters. TEST [21] makes the large language model (LLM) understand time series by aligning its embeddings to LLM textual space through contrastive learning. LLMTIME [5] encodes numerical time series as text and proposes procedures to effectively process time series. Some works extract time series properties and patterns for accuracy improvement [30, 29, 40]. TEMPO [1] decomposes time series and designs a shared prompt pool to alleviate data distribution shift. LLM-Mob [25] also decomposes human mobility sequences to help LLM understand the underlying passenger patterns. Nevertheless, all of them are not applicable to scenarios with additional data modality available (e.g., textual data).

For multimodal approaches, both UniTime [14] and Time-LLM [7] utilize text modality as an auxiliary to time series. In particular, their textual signals are mainly coarse and designed for the whole dataset instead of individual samples, like data description [7], and domain description [14]. Some works utilize more fine-grained multimodal signals at the sample level. For example, paired ECG signals and clinical reports for ECG are used for classification in METS [10]. However, existing multimodal approaches tend to adopt textual input to enhance time series modeling. With time series as the primary modality, there is little focus on mining complementary information between different modalities. Furthermore, current LM-based time series methods mainly focus on the model transferability of forecasting tasks, neglecting discussions on classification tasks.

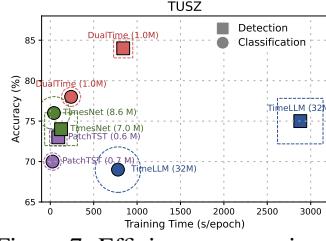


Figure 7: Efficiency comparison on TUSZ. The dotted size represents the model trainable parameter size. DualTime ranges middle in size but with best performance.

5 Conclusions

In this paper, we discuss the existing multimodal time series models from the perspective of multimodal fusion paradigms. Instead of utilizing one modality to serve the other modality, we propose DualTime, a novel multimodal framework delving into the complementary modeling of different modalities. The dual adapter design further facilitates the textual-temporal multimodal fine-tuning efficiency and achieves better embedding space alignment via a shared pre-trained language model pipeline. Considering the significant performance gain, the extensive experiments demonstrate that DualTime serves as an effective representation learner in both supervised and unsupervised settings. Regarding the transferability of the model, we demonstrate the superiority of DualTime through a few-shot label transfer experiment. In future work, we will investigate how various language models (e.g., LLaMA, OPT) influence multimodal learning results. We remain confident that our model design is adaptable to different language model backbones.

References

- [1] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2023.
- [2] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. LLM4TS: Two-Stage Fine-Tuning for Time-Series Forecasting with Pre-Trained LLMs. In *arXiv:2308.08469*, 2023.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [4] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2352–2359, 2021.
- [5] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.
- [6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [7] 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. Time-lm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2023.
- [8] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.
- [9] Jia Li, Yu Rong, Helen Meng, Zhihui Lu, Timothy Kwok, and Hong Cheng. Tatk: predicting alzheimer’s disease with actigraphy data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 509–518, 2018.
- [10] Jun Li, Che Liu, Sibo Cheng, Rossella Arcucci, and Shenda Hong. Frozen language model helps ecg zero-shot learning. In *Medical Imaging with Deep Learning*, pages 402–415. PMLR, 2024.
- [11] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.
- [12] Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*, 2024.

- [13] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.
- [14] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, 2024.
- [15] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.
- [16] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [17] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.
- [18] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [19] Vinit Shah, Eva Von Weltin, Silvia Lopez, James Riley McHugh, Lillian Veloso, Meysam Golmohammadi, Iyad Obeid, and Joseph Picone. The temple university hospital seizure detection corpus. *Frontiers in neuroinformatics*, 12:83, 2018.
- [20] Nils Strodthoff, Temesgen Mehari, Claudia Nagel, Philip J Aston, Ashish Sundar, Claus Graff, Jørgen K Kanters, Wilhelm Haverkamp, Olaf Dössel, Axel Loewe, et al. PtB-XL+, a comprehensive electrocardiographic feature dataset. *Scientific data*, 10(1):279, 2023.
- [21] Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. In *The Twelfth International Conference on Learning Representations*, 2023.
- [22] Siyi Tang, Jared A Dunnmon, Khaled Saab, Xuan Zhang, Qianying Huang, Florian Dubost, Daniel L Rubin, and Christopher Lee-Messer. Self-supervised graph neural networks for improved electroencephalographic seizure analysis. *arXiv preprint arXiv:2104.08336*, 2021.
- [23] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. PtB-XL, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):1–15, 2020.
- [24] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.
- [25] Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. Where would i go next? large language models as human mobility predictors. *arXiv preprint arXiv:2308.15197*, 2023.
- [26] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.
- [27] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.
- [28] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The eleventh international conference on learning representations*, 2022.
- [29] Hao Xue and Flora D Salim. Utilizing language models for energy load forecasting. In *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 224–227, 2023.

- [30] Hao Xue, Bhanu Prakash Voutharoja, and Flora D. Salim. Leveraging language foundation models for human mobility forecasting. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 1–9, 2022.
- [31] Jiexia Ye, Weiqi Zhang, Ke Yi, Yongzi Yu, Ziyue Li, Jia Li, and Fugee Tsung. A survey of time series foundation models: Generalizing time series representation with large language mode. *arXiv preprint arXiv:2405.02358*, 2024.
- [32] Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal data meets llm-explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*, 2023.
- [33] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987, 2022.
- [34] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.
- [35] Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3273–3285, 2023.
- [36] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.
- [37] Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022.
- [38] Weiqi Zhang, Jianfeng Zhang, Jia Li, and Fugee Tsung. A co-training approach for noisy time series learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3308–3318, 2023.
- [39] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2022.
- [40] Zheng Zhang, Hossein Amiri, Zhenke Liu, Andreas Züfle, and Liang Zhao. Large Language Models for Spatial Trajectory Patterns Mining. In *arXiv:2310.04942*, 2023.
- [41] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.
- [42] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.
- [43] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36, 2024.

A Datasets

A.1 Dataset Details

We show the summary of datasets in Table A.1 with dataset statistics and data splitting displayed.

For PTB-XL, the coarse-grained labels divide the samples into four classes: *Normal ECG*, *Borderline ECG*, *Abnormal ECG*, *Otherwise normal ECG* [20], and the fine-grained labels refer to *Normal ECG*, *Conduction Disturbance*, *Myocardial Infarction*, *Hypertrophy*, and *ST/T change*. Similarly, the coarse-grained labels of TUSZ distinguish seizure and non-seizure EEG signals and the fine-grained labels provide further seizure classification: *CF*, *GN*, *AB*, *CT*.

Table A.1: Dataset statistics and data split for PTB-XL and TUSZ datasets.

	PTB-XL		TUSZ	
	Detection	Classification	Detection	Classification
Size of Training Set	17084	17084	7766	1924
Size of Validation Set	2146	2146	5426	446
Size of Test Set	2158	2158	8848	521
Number of Classes	4	5	2	4
Sequence Length	1000	1000	6000	6000
Number of Channels	12	12	19	19
Average Text Length	13.7	13.7	24.3	23.0

A.2 Examples of Experimental Datasets

PTB-XL dataset contains clinical 12-lead electrocardiograms (ECGs) and their corresponding reports. The clinical reports are automatically generated by the machine and have no diagnosis revealed. TUSZ dataset is the largest EEG seizure database containing 19-channel EEG signals and clinical notes of each patient, for example, clinical history, medications, etc. In this work, we take the clinical history as the experimental textual input.

Furthermore, we show two examples for PTB-XL and TUSZ dataset in Figure A.1, respectively. Both time series data and textual data are displayed.

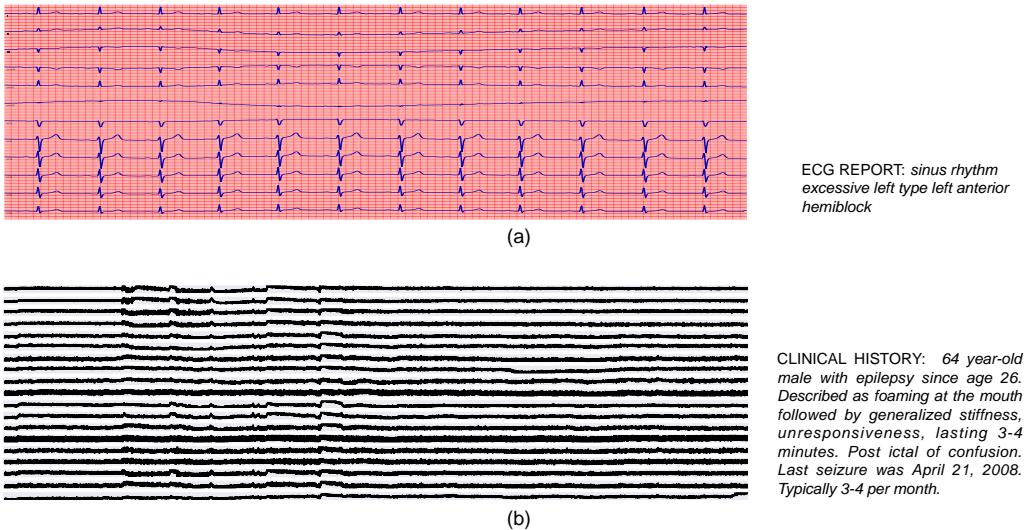


Figure A.1: Examples of experimental datasets. (a): PTB-XL dataset collected for electrocardiogram (ECG) analysis. (b): TUSZ dataset collected for electroencephalogram (EEG) analysis.

Table B.1: Full experiment results in unsupervised representation learning. Based on the frozen encoder, 100% labeled data are used for linear classifier training.

Modality	Model	PTB-XL						TUSZ						Average	
		Detection			Classification			Detection			Classification				
		Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	F1
Text	GPT2	0.72	0.65	0.56	0.58	0.73	0.65	0.61	0.62	0.72	0.49	0.49	0.50	0.64	0.53
Time	BERT	0.70	0.64	0.51	0.53	0.73	0.65	0.59	0.62	0.72	0.49	0.49	0.50	0.58	0.49
	TSTCC	0.68	0.57	0.53	0.54	0.65	0.56	0.48	0.50	0.74	0.51	0.50	0.48	0.67	0.44
	TS2vec	0.61	0.46	0.43	0.43	0.61	0.54	0.48	0.49	0.70	0.49	0.49	0.49	0.70	0.57
	TSCoT	0.73	0.71	0.58	0.60	0.75	0.68	0.61	0.63	0.67	0.54	0.57	0.53	0.69	0.76
Text + Time	PatchTST	0.60	0.53	0.38	0.35	0.55	0.45	0.32	0.30	0.73	0.50	0.50	0.50	0.67	0.63
	METS	0.74	0.66	0.57	0.58	0.71	0.64	0.57	0.60	0.65	0.55	0.59	0.53	0.57	0.46
	DualTime (Time)	0.68	0.52	0.46	0.44	0.60	0.48	0.39	0.40	0.68	0.52	0.52	0.51	0.66	0.66
	DualTime (Text)	0.72	0.66	0.55	0.57	0.73	0.66	0.63	0.64	0.70	0.50	0.50	0.50	0.70	0.58
	DualTime	0.75	0.68	0.59	0.62	0.77	0.71	0.65	0.67	0.75	0.60	0.57	0.58	0.75	0.60

B Experimental Details

B.1 Evaluation Metrics

The evaluation metrics we consider in this paper include accuracy, precision, recall, f1-score. The calculation of these metrics is as follows. For multi-class classification, we report the macro average results.

- **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1 Score:**

$$F1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Here, TP , TN , FP , and FN represent *True Positives*, *True Negatives*, *False Positives*, and *False Negatives*, respectively.

B.2 Detailed Results

We show the unsupervised representation learning results (simplified as Table 3 in the main text) in Table B.1.

The full results with all the baseline methods compared will be shown in Figure B.1 and Figure B.2, whose corresponding simplified figures in the main text are Figure 4 and Figure 5.

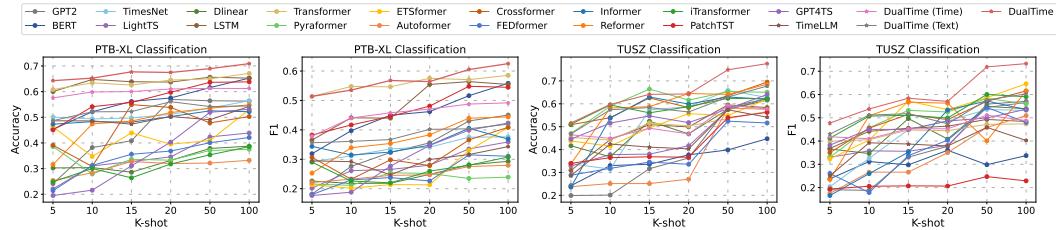


Figure B.1: Full results for label transfer with different few-shot settings.

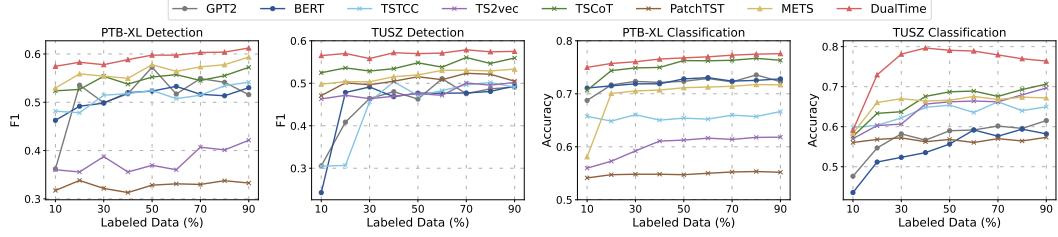


Figure B.2: Full results for unsupervised representation learning with different proportions of labeled data.

B.3 Multimodal Fusion Gating Analysis

To gain deeper insights into how multimodal information is fused within each adapter, we display the multimodal adaptation token fusion gating parameter across different transformer layers in Figure B.3. Considering the zero-initialized strategy, there is no multimodal fusion at the start of training. As training progresses, the values (absolute values) of the gating parameters continuously increase, indicating an intensification of multimodal fusion. Concurrently, we observe that the values of the gate parameters are larger in the initial (Layer 1&2) and final (Layer 10&11) few layers of the transformer compared to the middle layers (Layer 5&6). We speculate that this is because the model relies on and integrates multimodal data more when it is initially input into the model and closer to the downstream task head.

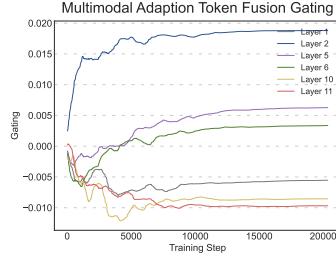


Figure B.3: Multimodal gating parameters of different transformer layers.

B.4 Sensitivity Analysis

Generally speaking, the performance of our model tends to improve with an increase in the number of network layers and the length of adaption tokens (as shown in Figure B.4). Compared to adaption token length P , the influence of multimodal fusion layers M is more evident.

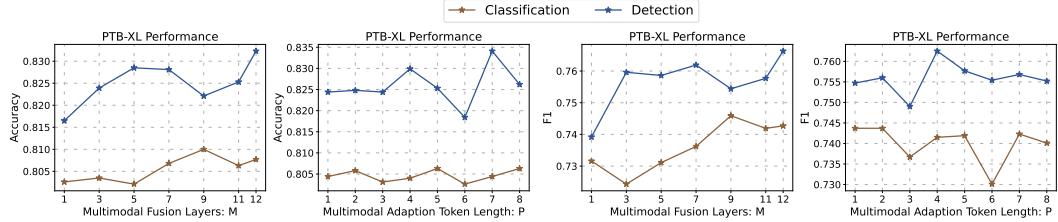


Figure B.4: Hyperparameter study of multimodal fusion layers M and length of adaption tokens P .