Information Sciences 681 (2024) 121213

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Relation-preserving masked modeling for semi-supervised time-series classification

Sangho Lee [a,b], Chihyeon Choi [a,b], Youngdoo Son [a,b,*]

[a] Department of Industrial and Systems Engineering, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul, 04620, Republic of Korea
[b] Data Science Laboratory (DSLAB), Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul, 04620, Republic of Korea

ARTICLE INFO

Keywords:
Semi-supervised learning
Time-series classification
Masked time-series modeling
Self-supervised learning

ABSTRACT

In this study, we address the challenge of label sparsity in time-series classification using semi-supervised learning that effectively leverages numerous unlabeled instances. Our approach introduces a pioneering framework for semi-supervised time-series classification based on masked time-series modeling, a recent advancement in self-supervised learning that can effectively capture intricate temporal structures in time series. The proposed method first extracts the intrinsic semantic information from unlabeled instances by considering diverse temporal resolutions and using various masking ratios during model training. Subsequently, we combine the semantic information captured from unlabeled instances with supervisory features obtained from labeled instances that encompass hard-to-learn class information to enhance classification performance. Extensive experiments on semi-supervised time-series classification demonstrate the superiority of the proposed method by achieving state-of-the-art performance.

1. Introduction

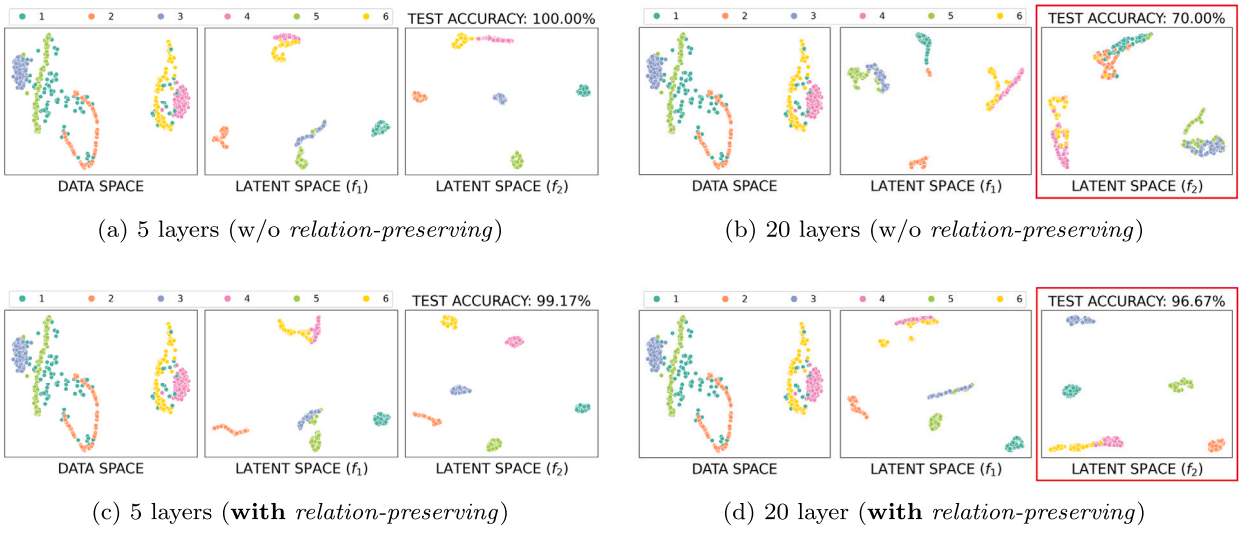
Recent advances in deep learning have shown promising performance in time-series classification, a fundamental task driven by the increasing accessibility of vast time-series data [22]. These remarkable achievements require numerous labeled training instances; however, in practice, these are frequently lacking, whereas unlabeled instances abound [3,15]. Annotating all unlabeled time series within a reasonable time and cost is often infeasible [25,35]. Therefore, semi-supervised learning, which leverages both labeled and unlabeled instances to mitigate label sparsity, has attracted considerable attention in the context of time-series classification [4].

Semi-supervised learning aims to enhance the generalization capability of models by leveraging a large set of unlabeled instances along with a few labeled ones during model training. Recent studies on semi-supervised learning have actively exploited self-supervised learning, particularly contrastive learning, to learn implicit temporal patterns within unlabeled time series under the supervision of self-generated labels [24,26]. However, these studies have two limitations. First, most captured coarse-grained context information focused on the instance level, resulting in insufficient recognition of the temporal patterns of time series [30,38,41]. Second, the model performance highly depends on the techniques used to construct self-generated labels [33]. For example, when augmented time series are generated as self-generated labels, the classification performance can differ with respect to the selected

* Corresponding author at: Department of Industrial and Systems Engineering, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul, 04620, Republic of Korea.
E-mail address: youngdoo@dongguk.edu (Y. Son).

https://doi.org/10.1016/j.ins.2024.121213
Received 6 February 2024; Received in revised form 4 July 2024; Accepted 18 July 2024

Available online 23 July 2024



(a) 5 layers (w/o *relation-preserving*) (b) 20 layers (w/o *relation-preserving*)

(c) 5 layers (**with** *relation-preserving*) (d) 20 layer (**with** *relation-preserving*)

Fig. 1. Data space and latent spaces produced by the sequential sub-encoders f_1 and f_2 for *SyntheticControl*. In (a) and (b), we present each space when f_2 has 5 and 20 layers, respectively, without *relation-preserving*. In contrast, (c) and (d) show each space when f_2 has 5 and 20 layers, respectively, with *relation-preserving*. In (b), the representations of each class from f_2 are less distinctive compared to those from f_1. However, in (d), with *relation-preserving* loss function, the representations of each class remain distinguishable.

data augmentation methods, such as jittering and time warping [39,49]. For time series in particular, it is also challenging to adopt proper perturbations that do not corrupt the nature of the time series [45].

To capture fine-grained context information and deal with the high sensitivity of constructing self-generated labels, the concept of masked modeling has emerged in natural language processing and computer vision [10,42]. Masked modeling aims to learn useful representations reflecting semantic information from data by enabling a model to reconstruct the masked content based on the unmasked part. However, unlike text and images, which possess rich semantic information within words or patches, the semantic information of a time series is generally contained in temporal variations, such as trends and periodicity [11,43]. Thus, to sufficiently consider time-series characteristics while leveraging the representation ability of masked modeling, some studies have extended this concept to time series, an approach called *masked time-series modeling (MTM)* [11,31].

The powerful representation ability of MTM can benefit semi-supervised learning by effectively capturing semantic information from unlabeled time series; however, to our knowledge, it has not yet been introduced to semi-supervised time-series classification. Even if we attempt to exploit existing MTM approaches for semi-supervised time-series classification, two potential drawbacks exist. First, although considering temporal dependencies, which span different time intervals [37,46], can enrich the semantic information of a time series, the transformer architecture typically adopted as an encoder in MTM insufficiently captures temporal patterns at different time scales [5,44,47]. Second, conventional MTM is sensitive to masking ratios [7,23]. It is often impractical to explore the optimal masking ratio for each dataset originating from various sources within a reasonable time and cost.

Therefore, in this work, we propose the first MTM-based framework for semi-supervised time-series classification, *Masked Dual-Temporal Autoencoder* (MDTA), to address these challenges. MDTA captures relevant semantic information from an unlabeled time series and incorporates it with supervisory features obtained from labeled time series to enhance classification performance. Specifically, we develop a *dual-temporal encoder* comprising two sequential sub-encoders, one of which captures high-level information reflecting temporal dependencies at different time scales while the other leverages this information to effectively learn intrinsic temporal patterns within the time series. However, as shown in Figs. 1(a) and (b), the dual-temporal encoder with a deep architecture may cause information loss, leading to performance degradation. Thus, we additionally introduce a simple yet effective *relation-preserving* loss function to ensure a lossless flow of temporal information within the encoder. In addition, during model training, we use *random masking ratios* to avoid exploring optimal masking ratios, while further enhancing the model's ability to capture the temporal relations of the time series. By sharing the dual-temporal encoder, MDTA directly classifies labeled instances and follows the masked modeling procedure for unlabeled instances; therefore, the labeled instances provide useful supervisory features for classification, and the unlabeled instances enrich the semantic information of the time series, improving classification performance.

Through extensive experiments on semi-supervised time-series classification, we demonstrate that the proposed method outperforms state-of-the-art (SOTA) methods by successfully leveraging semantic information from unlabeled time series.

In summary, this study provides the following main contributions:

- We propose the first MTM-based framework for semi-supervised time-series classification that effectively captures intricate temporal patterns across diverse temporal resolutions using a *dual-temporal encoder* comprising two consecutive sub-encoders.
- A *relation-preserving* loss function is introduced to address the problem of potential information loss between the sub-encoders in our encoder architecture.

- We use *random masking ratios* at each training epoch to avoid the high-cost tuning process required to find the optimal masking ratios, as well as to enhance classification performance.
- Our approach captures the inherent temporal information of a time series and successfully combines it with supervisory features, thus outperforming SOTA methods in semi-supervised time-series classification.

The remainder of this paper is organized as follows. In Section 2, we briefly review semi-supervised learning and masked time-series modeling. Next, we introduce the detailed algorithm of the proposed framework in Section 3. In Section 4, we present extensive experiments using various time-series datasets to demonstrate the superiority of our approach. Finally, concluding remarks are presented in Section 5.

2. Related work

Label sparsity is a practical obstacle hindering the use of deep learning in time-series classification. Thus, to alleviate reliance on labeled instances, semi-supervised learning for time-series classification has been extensively studied [2,4].

Pseudo-labeling strategy. This approach is a traditional and popular strategy for semi-supervised learning [32]. Specifically, this strategy assumes that the decision boundary is located in a low-density region of the marginal distribution; therefore, it trains a classifier to allow unlabeled instances to have low entropy. For example, Lee et al. [21] generated pseudo-labels for unlabeled instances based on current model predictions to supervise them. Laine and Aila [20] introduced consistency regularization for pseudo-labeling to utilize the relationships between labeled and unlabeled instances during model training. However, these methods cannot consider the temporal relations of time series; it is often difficult to assign accurate pseudo-labels to unlabeled instances because of the manually set threshold, and the high confidence of deep learning makes them vulnerable to noise [40].

Contrastive learning. Recent studies on semi-supervised time-series classification have actively exploited self-supervised learning, particularly contrastive learning, to extract and leverage context information from unlabeled time series under the supervision of self-generated labels [34]. For example, Jawed et al. [19] proposed a semi-supervised time-series classification method that combines self-supervised learning and multitask learning. Fan et al. [14] identified temporal relations by using the past-future segments and constructing positive and negative pairs to extract useful context from unlabeled instances. Extending Fan et al. [14], Xi et al. [40] considered temporal patterns between not only the past and future segments but also the present one. In addition, Liu et al. [24] learned the temporal structures of unlabeled instances with self-generated labels obtained by randomly applying time-sampling functions to the input time series, and Eldele et al. [12] introduced temporal and contextual contrasting for semi-supervised time-series classification. Liu et al. [26] proposed a temporal-frequency con-training method that utilizes complementary information from two distinct views for unlabeled instances for semi-supervised time-series classification, and Liu et al. [27] designed a shapelet-based diffusion learning mechanism and contrastive language-shapelet learning mechanism to effectively leverage shapelet learning in semi-supervised time-series classification. However, most of these methods have certain limitations, such as being highly sensitive to the construction of self-generated labels [33,39,45,49] and insufficiently reflecting context information [30,38,41].

Masked modeling. To address the limitations of contrastive learning, the concept of masked modeling, which extracts semantic information from data by enabling a model to reconstruct the masked content based on the unmasked part, has emerged in natural language processing and computer vision [10,42]. However, unlike text and images, which can contain rich semantic information in words and patches, time series typically embed their semantic information in temporal variations, such as trends and periodicity [11,43]. Thus, some studies have extended masked modeling to time series. For example, in time-series representation learning, Zerveas et al. [46] designed a transformer encoder for MTM, whereas Dong et al. [11] used MTM to capture complementary temporal variations from multiple masked series. In addition, Nie et al. [31] improved the performance of long-term time-series forecasting using MTM. Despite these empirical successes, there have been no attempts to introduce MTM into semi-supervised time-series classification. Moreover, conventional MTM has two notable drawbacks: inability to reflect diverse temporal resolutions [5,47] and high sensitivity to masking ratios [7,23].

In contrast, our method, MDTA, is the first MTM framework for semi-supervised time-series classification. It captures valuable semantic information from an unlabeled time series by effectively reflecting diverse temporal resolutions and using random masking ratios during model training. Moreover, we achieve superior classification performance by incorporating the obtained semantic information with supervisory features learned from labeled instances.

3. Proposed method

To enhance classification performance by effectively leveraging a large set of unlabeled instances along with labeled instances, we propose MDTA, a novel MTM framework for semi-supervised time-series classification.

3.1. Problem statement

Let $\mathbb{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ be a set of n samples, where $\boldsymbol{x}_i \in \mathbb{R}^{t \times v}$ is a time-series instance with t lengths and v variables, and y_i denote the class label of $\boldsymbol{x}_i$. We suppose some of the labels to be missing; thus, $\mathbb{D}$ is split into two subsets: a labeled set $\mathbb{D}_\ell = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_\ell}$ of size n_ℓ and an unlabeled set $\mathbb{D}_u = \{(\boldsymbol{x}_i, \cdot)\}_{i=n_\ell+1}^{n}$ of size $n_u = n - n_\ell$. We define two sequential sub-encoders $f_1 : \boldsymbol{x} \to \boldsymbol{u}$ and $f_2 : \boldsymbol{u} \to \boldsymbol{z}$, and

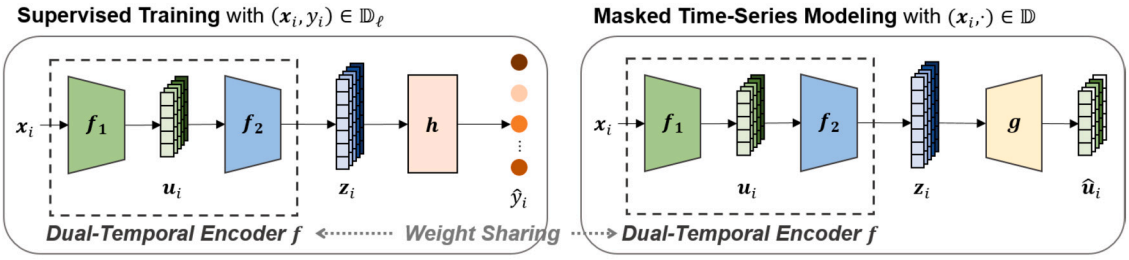


Fig. 2. Overview of the proposed method.

a decoder $g : \boldsymbol{z} \to \hat{\boldsymbol{u}}$, where $\boldsymbol{z} \in \mathbb{R}^{t \times d_z}$, $\boldsymbol{u} \in \mathbb{R}^{t \times d_u}$, and $\hat{\boldsymbol{u}} \in \mathbb{R}^{t \times d_u}$. In particular, for $\mathbb{D}_\ell$, we also define a classification head $h : \boldsymbol{z} \to \hat{y}$. The objective is to optimize f, g, and h using all accessible instances in $\mathbb{D}$ to improve classification performance.

3.2. Masked dual-temporal autoencoder

Fig. 2 shows an overview of MDTA, which incorporates supervised training with the MTM paradigm that learns useful representations reflecting the semantic information of a time series by reconstructing the masked content using the unmasked part. Our method consists of three architectural components: the *dual-temporal encoder*, *simple decoder*, and *classification head*.

3.2.1. Dual-temporal encoder

To effectively learn the inherent temporal structures of time series, we develop a *dual-temporal encoder* f sequentially configured with a multi-resolution sub-encoder f_1 and transformer-based sub-encoder f_2 ($f := f_2 \circ f_1$). Note that the weights of this encoder are shared between MTM and supervised training.

Multi-resolution sub-encoder. Despite the effectiveness and scalability of the transformer used as an encoder in MTM, it has limitations in controlling diverse temporal resolutions [37,44]. Because temporal dependencies span various time intervals, reflecting diverse resolutions can significantly improve the model performance for time-series classification [5,37,46,47]. Thus, we construct the multi-resolution sub-encoder f_1 before the transformer-based sub-encoder f_2 to capture temporal dependencies at different time scales.

Let $\boldsymbol{x}_i \in \mathbb{R}^{t \times v} = [\boldsymbol{x}_{i,1}, \cdots, \boldsymbol{x}_{i,t}]$ be a time series composed of a sequence of t observations. $\boldsymbol{x}_i$ are mapped onto a d_u-dimensional latent space along the temporal dimension using f_1 as follows:

$$\boldsymbol{u}_i = f_1(\boldsymbol{x}_i), \tag{1}$$

where $\boldsymbol{u}_i \in \mathbb{R}^{t \times d_u} = [\boldsymbol{u}_{i,1}, \cdots, \boldsymbol{u}_{i,t}]$ denotes the high-level temporal features used as the input for the subsequent sub-encoder f_2. Each time step of $\boldsymbol{u}_i$ corresponds to that of $\boldsymbol{x}_i$.

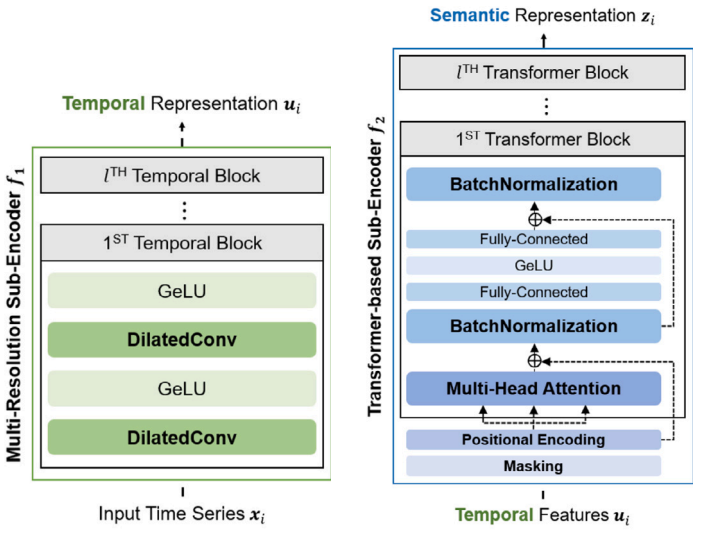
Specifically, the sub-encoder f_1 is designed using *one-dimensional convolutional layers with causal padding and dilated filters* (*DilatedConv*). This architecture ensures that the model does not perturb the temporal order of the input time series and considers diverse temporal resolutions by gradually increasing the dilation rate ρ [1]. In particular, causal padding prevents the convolution filter from observing future inputs beyond the current time step through zero padding on the left side of $\boldsymbol{x}_i$. In addition, the dilated filters, convolution filters with strides controlled by ρ, allow the model to recognize various temporal patterns at different time scales. For a single time step τ,[1] the output of *DilatedConv*, $\boldsymbol{x}'_{i,\tau}$, is calculated as

$$\boldsymbol{x}'_{i,\tau} = DilatedConv(\boldsymbol{x}_{i,\tau}) = \sum_{\kappa=0}^{k-1} \boldsymbol{x}_{i,(\tau-\kappa\rho-(k-1)\rho)} \times c_\kappa, \tag{2}$$

where c_κ is the weight (or kernel coefficient) at time step κ in the convolution filter, and k is the filter size. By passing $\boldsymbol{x}_i$ through several temporal blocks comprising *DilatedConv*s and GeLU activation functions [17] (see Fig. 3(a)), we obtain high-level temporal features $\boldsymbol{u}_i$, which reflect intricate temporal patterns using various dilation rates that control how the receptive field of the convolution filter expands across time steps. Moreover, these features enhance the efficiency of the subsequent f_2 by allowing it to access the high-level temporal information obtained from f_1 [13]. In other words, we can achieve the same performance with a relatively shallow f_2.

Transformer-based sub-encoder. The high-level features $\boldsymbol{u}_i$ extracted from the multi-resolution sub-encoder f_1 are used as inputs for the subsequent sub-encoder f_2. As shown in Fig. 3(b), we construct f_2 as the transformer introduced by Zerveas et al. [46]. Note that f_2 could be flexibly replaced by any model with a similar transformer architecture.

[1] Here, we show the operation for a single time step for clear presentation; all input values are embedded simultaneously by a single matrix multiplication.



(a) Multi-resolution sub-encoder f_1 (b) Transformer-based sub-encoder f_2

Fig. 3. Configuration of (a) multi-resolution and (b) transformer-based sub-encoders.

Following the MTM paradigm [11,46], we hide some time steps of $\boldsymbol{u}_i$ with uniformly distributed masks and make the model predict the masked values. In particular, we create a binary mask $\boldsymbol{m}_i \in \mathbb{R}^t$ with a masking ratio $r \in [0.1, 0.9]$. Then, the masked features $\bar{\boldsymbol{u}}_i \in \mathbb{R}^{t \times d_u}$ are derived through element-wise multiplication: $\bar{\boldsymbol{u}}_i = \boldsymbol{m}_i \odot \boldsymbol{u}_i$. Time series inherently contain information redundancy, enabling the recovery of missing values even with a basic understanding of the temporal patterns observed at adjacent time steps. However, when the masking ratio is high, this information redundancy can be eliminated, creating a challenging self-supervisory task that allows the model to identify sophisticated temporal relations. In addition, as with most conventional MTM approaches, exploring the optimal masking ratio for each individual dataset can be time-consuming [7,23]. Thus, during model training, we randomly pick the masking ratio $r \in [0.1, 0.9]$ for every epoch to avoid searching for the optimal masking ratio while enhancing the capability to extract semantic information from $\boldsymbol{u}_i$. In other words, MDTA can capture inherent temporal relations by simultaneously considering a wide range of masking ratios.

Subsequently, because the transformer is insensitive to the ordering of input time series, we add positional encodings $\boldsymbol{\xi}_i \in \mathbb{R}^{t \times d_u}$, obtained via deterministic sinusoidal encoding [36], to $\bar{\boldsymbol{u}}_i$ to indicate the sequential nature of the time series: $\tilde{\boldsymbol{u}}_i = \bar{\boldsymbol{u}}_i + \boldsymbol{\xi}_i$.

Finally, by passing $\boldsymbol{u}_i$ through the transformer-based sub-encoder f_2, a semantic representation $\boldsymbol{z}_i \in \mathbb{R}^{t \times d_z}$ is generated as follows:

$$\boldsymbol{z}_i = f_2(\boldsymbol{m}_i \odot \boldsymbol{u}_i + \boldsymbol{\xi}_i) = f_2(\bar{\boldsymbol{u}}_i + \boldsymbol{\xi}_i) = f_2(\tilde{\boldsymbol{u}}_i). \tag{3}$$

Relation-preserving loss function. The proposed dual-temporal encoder $f := f_2 \circ f_1$ has a potential risk in that temporal information obtained from f_1 can be distorted after passing through f_2. In other words, as shown in Figs. 1(a) and (b), information loss can occur within the encoder f because of its deep sequential architecture, degrading performance by making the representations of some classes indistinguishable [8]. Thus, we introduce a simple yet effective *relation-preserving* loss function to prevent the loss of temporal structural information obtained from the former sub-encoder f_1, even after passing the subsequent sub-encoder f_2.

To identify structural relations between latent features of $\boldsymbol{u}_i$ along the temporal dimension, we create an adjacency matrix $\mathcal{A}_i \in \mathbb{R}^{d_u \times d_u}$ based on similarities between latent features across time steps of $\boldsymbol{u}_i \in \mathbb{R}^{t \times d_u}$ as follows:

$$\mathcal{A}_i = \lfloor \sigma(\boldsymbol{u}_i^\top \boldsymbol{u}_i) \rceil, \tag{4}$$

where σ is the sigmoid function that maps input values to the range from zero to one, and $\lfloor \cdot \rceil$ denotes the rounding operation. The adjacency matrix $\mathcal{A}_i$ derived from $\boldsymbol{u}_i$ is regarded as the ground truth of the temporal structures that should be maintained. Then, we obtain another adjacency matrix $\hat{\mathcal{A}}_i \in \mathbb{R}^{d_z \times d_z}$ with $\boldsymbol{z}_i \in \mathbb{R}^{t \times d_z}$ as follows:

$$\hat{\mathcal{A}}_i = \sigma(\boldsymbol{z}_i^\top \boldsymbol{z}_i). \tag{5}$$

Note that d_u and d_z should have the same dimensions. Finally, we minimize the difference between each element of $\mathcal{A}_i$ and $\hat{\mathcal{A}}_i$ to preserve the structural relations of the temporal information captured by f_1 as follows:

$$\mathcal{L}_{RP,i} = - \sum_{a_{pq} \in \mathcal{A}_i, \hat{a}_{pq} \in \hat{\mathcal{A}}_i} a_{pq} \log \hat{a}_{pq} + (1 - a_{pq}) \log(1 - \hat{a}_{pq}), \tag{6}$$

where a_{pq} and $\hat{a}_{pq}$ are the (p,q) elements of $\mathcal{A}_i$ and $\hat{\mathcal{A}}_i$, respectively. As shown in Figs. 1(c) and (d), this loss function helps ensure a lossless flow of temporal information between f_1 and f_2, effectively capturing intricate temporal patterns in the time series and enhancing the model performance. The effect of this loss function is further discussed in Section 4.3.

3.2.2. Simple decoder

Following the MTM paradigm, a decoder only predicts masked values; therefore, its architecture can be flexibly designed regardless of the encoder architecture [16]. Thus, we design a lightweight decoder g as a fully connected layer to reduce the number of calculations in the training phase. Note that this decoder reconstructs $\boldsymbol{u}_i$ obtained by f_1 based on $\boldsymbol{z}_i$ generated by f_2 (see Fig. 2). Here, we calculate the mean squared error only for the masked content; hence, the *reconstruction* loss function is defined as follows:

$$\mathcal{L}_{RE,i} = \frac{1}{|\mathbb{M}|} \sum_{\tau \in \mathbb{M}} (\hat{\boldsymbol{u}}_{i,\tau} - \boldsymbol{u}_{i,\tau})^2, \tag{7}$$

where $\mathbb{M}$ is a set of indices of masked values, and $\hat{\boldsymbol{u}}_{i,\tau}$ is the value reconstructed by the decoder g.

3.2.3. Classification head

We employ a classification head h along with f and g to obtain supervisory features, including hard-to-learn class information, from labeled instances. Here, we design h using two fully connected layers with batch normalization and a GeLU activation function.

Given a semantic representation $\boldsymbol{z}_i$ corresponding to $(\boldsymbol{x}_i, y_i) \in \mathbb{D}_\ell$, we first pass $\boldsymbol{z}_i$ through the average pooling layer (AP) and then use it as input for the classification head h. Formally, we obtain the predicted class label as $\hat{y}_i = h(\mathrm{AP}(\boldsymbol{z}_i)) = h(\mathrm{AP}(f(\boldsymbol{x}_i))$, where $f := f_2 \circ f_1$; thereby, the *classification* loss function is defined with the cross-entropy as follows:

$$\mathcal{L}_{CL,i} = -y_i \log \hat{y}_i. \tag{8}$$

3.2.4. Overall learning procedure

Following previous works [24,40], we first train f, g, and h with supervised learning using labeled time-series instances, and then update f and g with the MTM paradigm using all accessible instances.

Specifically, given $\mathbb{D}$ and its subsets $\mathbb{D}_\ell$ and $\mathbb{D}_u$, f, g, and h are trained using $\mathbb{D}_\ell$ with the *classification* loss function as follows:

$$\mathcal{L}_\ell = \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} \mathcal{L}_{CL,i}. \tag{9}$$

Subsequently, we further train f and g using all accessible instances, including unlabeled instances, in $\mathbb{D} = \mathbb{D}_\ell \cup \mathbb{D}_u$ with the *relation-preserving* and *reconstruction* loss functions as follows:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \alpha \mathcal{L}_{RP,i} + \beta \mathcal{L}_{RE,i}, \tag{10}$$

where α and β are the weights for each loss function. Note that we enrich the inherent semantic information of the time series by leveraging all accessible instances, including labeled and unlabeled time series, when updating f and g.

By iterating this learning procedure, the labeled instances enable the model to capture useful supervisory features suitable for classification, whereas all accessible instances enhance the implicit semantic information of the time series. Algorithm 1 summarizes the proposed method.

Algorithm 1 Learning procedure of MDTA.

Input: Set of n samples $\mathbb{D}$, its labeled subset $\mathbb{D}_\ell = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_\ell}$ and unlabeled subset $\mathbb{D}_u = \{(\boldsymbol{x}_i)\}_{i=n_\ell+1}^{n}$, dual-temporal encoder $f := f_2 \circ f_1$, simple decoder g, and classification head h
Output: Trained f, g, and h

```
Initialize f, g, and h.
for each epoch do
    for (x_i, y_i) ∈ D_ℓ do
        Obtain u_i and z_i by equations (1) and (3), respectively.
        Calculate L_CL,i by equation (8).
    end for
    Update f, g, and h by equation (9).
    for (x_i, ·) ∈ D do
        Obtain u_i and z_i by equations (1) and (3), respectively.
        Generate adjacency matrices A_i and Â_i by equations (4) and (5), respectively.
        Calculate L_RP,i and L_RE,i by equations (6) and (7), respectively.
    end for
    Update f and g by equation (10).
end for
```

Table 1
Detailed description of 15 datasets. The data type, numbers of data and classes, and sequence lengths are provided for each dataset.

Dataset	Abbreviation	Data Type	# of Data	# of Classes	Sequence Length
CBF	CB	Simulated	930	3	128
CricketX	CX	Motion	780	12	300
ECGFiveDays	EF	ECG	884	2	136
Lightning2	L2	Sensor	121	2	637
MoteStrain	MS	Sensor	1272	2	84
Plane	PL	Sensor	210	7	144
PowerCons	PC	Power	360	2	144
RefrigerationDevices	RD	Device	750	2	720
SonyAIBORobotSurface1	SR	Sensor	621	2	70
SwedishLeaf	SL	Image	1125	15	128
SyntheticControl	SC	Simulated	600	6	60
ToeSegmentation1	T1	Motion	268	2	277
Trace	TR	Sensor	200	4	275
TwoPatterns	TP	Simulated	5000	4	128
Yoga	YO	Image	3300	2	426

4. Experiments

We conducted extensive experiments to demonstrate the superiority of the proposed method, MDTA, in semi-supervised time-series classification.

4.1. Experimental settings

Here, we describe the experimental settings, including baseline methods, datasets, and evaluation metrics; the implementation details are provided in Appendix A.

Baselines. To demonstrate the efficacy of the proposed method, we compared MDTA with nine baseline methods, including SOTAs in semi-supervised time-series classification and a fully supervised model as follows:

- *CE* is trained in a supervised manner using the cross-entropy loss for labeled instances only.
- *Pseudo* [21] generates pseudo-labels of unlabeled instances based on the current model prediction to supervise them.
- Π-*model* [20] is a semi-supervised learning approach that incorporates consistency regularization into pseudo-labeling, exploiting relationships between labeled and unlabeled instances during model training.
- *FixMatch* [34] is a semi-supervised method that creates high-confidence pseudo-labels for weakly augmented unlabeled instances and uses these labels to supervise strongly augmented instances.
- *MTL* [19] is a semi-supervised time-series classification method that combines self-supervised and multitask learning.
- *SSTSC* [40] is a semi-supervised time-series classification approach that considers the inherent temporal information of a time series by exploring the temporal relations between the past, present, and future.
- *iTimes* [24] is a semi-supervised time-series classification method that captures the temporal structure of unlabeled instances by training the model to recognize time-sampling functions that are randomly applied to the input time series.
- *CA-TCC* [12] is a SOTA in semi-supervised time-series classification that jointly leverages temporal and contextual contrasting.
- *TS-TFC* [26] is a SOTA in semi-supervised time-series classification that introduces temporal-frequency co-training with complementary information.

Datasets. We used 15 univariate time-series classification datasets from the UCR time-series classification archive [9]. Due to limitations on computing resources and time, it was difficult to utilize all the datasets in the UCR archive. Hence, as in several previous studies [12,14,19,24,40], we selected datasets with various data types, quantities, and sequence lengths. A detailed description of the datasets is provided in Table 1. We set the proportions of the training, validation, and test sets to 60%, 20%, and 20%, respectively, for each dataset. In addition, we split the training dataset into labeled and unlabeled instances based on the label ratios. All instances were normalized using z-scores. The proposed method can also cover multivariate time series, although we confined the evaluation to univariate time series ($v = 1$) to ensure fair comparisons with the baselines [24,40].

Evaluation metric. We evaluated the classification performance by measuring the accuracy score.

4.2. Comparison with baselines

The model performance of semi-supervised time-series classification can be evaluated using inductive and transductive inference. Inductive inference involves measuring the model performance using a test dataset that is separate from the training dataset, whereas transductive inference evaluates the model performance for unlabeled instances in the training dataset. Referring to Xi et al. [40]

Table 2
Average classification performance across label ratios ranging from 0.1 to 0.9 for MDTA and baselines under inductive inference. For each dataset, the best score is highlighted in boldface.

Dataset	CE	Pseudo	Π-model	FixMatch	MTL	SSTSC	iTimes	CA-TCC	TS-TFC	MDTA (ours)
CB	99.20	99.30	99.26	99.40	98.38	99.38	99.44	**99.84**	99.22	99.71
CX	52.26	52.62	61.92	59.66	40.38	41.89	67.04	**71.03**	41.58	64.07
EF	98.49	98.51	83.44	83.33	98.39	98.24	95.33	97.27	99.44	**99.81**
L2	67.56	68.89	68.98	69.87	67.56	67.64	71.56	**75.67**	69.21	75.23
MS	93.29	93.46	94.15	94.19	89.17	92.31	93.56	93.24	92.66	**95.03**
PL	96.98	96.98	85.98	88.73	84.87	94.50	85.66	96.35	**98.72**	96.88
PC	89.41	88.89	85.37	86.23	87.84	89.07	87.78	87.20	87.50	**93.58**
RD	58.60	57.61	57.04	57.32	57.51	58.19	60.64	**62.26**	61.72	59.56
SR	97.35	97.21	93.44	94.22	93.28	96.80	94.20	77.90	97.64	**99.61**
SL	84.16	84.83	70.34	69.66	55.45	76.93	56.18	77.67	**89.71**	86.18
SC	96.19	97.70	97.98	97.52	96.98	93.78	97.31	91.85	96.35	**98.11**
T1	84.16	84.44	84.73	84.36	82.30	82.39	86.71	67.87	90.04	**94.03**
TR	91.94	93.22	91.72	92.72	91.50	91.44	95.78	81.94	97.72	**98.44**
TP	99.81	99.85	99.30	99.48	98.91	99.73	96.86	**100.00**	98.58	99.97
YO	83.99	83.98	64.72	63.85	74.76	80.58	75.92	80.76	83.93	**85.17**
Avg. Rank	5.60**	4.60**	7.33**	6.00**	7.60**	6.40**	5.67**	5.47**	4.47*	**1.87**
(*p-value*)	(1.22e-4)	(1.22e-4)	(6.10e-5)	(6.10e-5)	(6.10e-5)	(6.10e-5)	(2.01e-3)	(4.13e-2)	(6.37e-2)	-

and Liu et al. [24], we focused on providing results under inductive inference; however, those under transductive inference are also given in Appendix E.

Table 2 shows the classification performance of the baseline methods and MDTA for 15 time-series datasets under the inductive setting. Here, we show the classification performance by averaging the accuracy scores across label ratios from 0.1 to 0.9 on each dataset. The complete results for different label ratios on each dataset are provided in Appendix E. Moreover, we performed statistical tests on the classification performance to ensure the significance of the performance improvement achieved by MDTA. Specifically, we employed a two-sample Wilcoxon signed rank test between MDTA and each baseline. The superscripts * and ** for the average rank in Table 2 imply that the p-value of the rank test was smaller than 0.1 and 0.05, respectively.

MDTA achieved the best average rank of 1.87, remarkably outperforming the baseline methods. The statistical test showed that MDTA performed significantly better than the baselines. In particular, when the labels were highly limited, the proposed method exhibited outstanding performance on most datasets (see Fig. E.2 in Appendix E), supporting the effectiveness of MDTA in leveraging unlabeled instances.

In addition, the proposed method showed superior performance in most datasets, regardless of label ratios, number of classes, and sequence length, by successfully incorporating the intrinsic semantic information of the time series with supervisory features. In contrast, the baseline methods exhibited performance differences according to the dataset specifications. For example, *Pseudo* showed relatively low performance on datasets with long sequences, such as the *RefrigerationDevices* and *Lightning2* datasets, because it does not consider the temporal dependency of the time series; *iTimes* performed poorly on *SwedishLeaf*, which has the largest number of classes, and *SonyAIBORobotSurface1*, which has a short sequence length.

Compared to the most recent work, CA-TCC, the proposed method outperformed CA-TCC on average, achieving better performance on 10 of 15 datasets. In particular, for several datasets, such as *PowerCons*, *SonyAIBORobotSurface1*, *SwedishLeaf*, *SyntheticControl*, *ToeSegmentation1*, *Trace*, and *Yoga*, our method achieved overwhelmingly superior performance compared to CA-TCC. In contrast, the performance gaps in the five datasets in which CA-TCC outperformed our method were relatively small. In addition, the classification performance of CA-TCC tended to vary greatly depending on the label ratio, unlike that of MDTA, indicating that its ability to integrate semantic information and supervisory features is relatively low (see Fig. E.2 in Appendix E). Moreover, when we compared MDTA to TS-TFC, another recent work with the second-best average rank, the proposed method also outperformed TS-TFC on average by showing better performance on 12 of 15 datasets.

4.3. Ablation studies

The proposed method, MDTA, has three key components:

- The **dual-temporal encoder architecture** for effectively capturing the intricate temporal structures of time series with diverse temporal resolutions using two sequential sub-encoders
- The **relation-preserving loss function** to prevent temporal information loss within the encoder
- **Random masking ratios** to avoid exploring optimal masking ratios while enhancing the model performance

To demonstrate the effectiveness of these components, we compared MDTA to three ablation models: MDTA replacing the multi-resolution sub-encoder with one fully connected layer (*MDTA w/o* f_1), MDTA without relation-preserving (*MDTA w/o* $\mathcal{L}_{RP}$), and MDTA without random masking ratios (*MDTA w/o RM*). For *MDTA w/o RM*, the masking ratio was fixed at 0.5.

Table 3
Average classification performance across label ratios ranging from 0.1 to 0.9 for MDTA and ablation models under the inductive setting. For each dataset, the best score is highlighted in boldface.

Dataset	MDTA (ours)	*w/o* f_1	*w/o RM*	*w/o* $\mathcal{L}_{RP}$	*w/o* $\mathcal{L}_{RE}$	*w/o* $\mathcal{L}_{RE}$ & $\mathcal{L}_{RP}$
CB	**99.71**	99.60	99.22	99.08	99.04	99.61
CX	**64.07**	34.24	61.32	58.78	37.78	31.44
EF	99.81	82.49	99.56	**99.87**	99.74	76.22
L2	**75.23**	67.85	67.41	72.59	70.22	60.89
MS	**95.03**	90.38	94.52	93.03	89.96	87.50
PL	96.88	93.83	96.12	**97.27**	95.77	89.10
PC	**93.58**	86.78	92.23	92.70	93.38	86.73
RD	59.56	53.85	58.22	**60.07**	50.95	61.69
SR	**99.61**	93.75	99.35	99.38	99.16	91.57
SL	**86.18**	68.89	83.36	83.60	55.71	60.38
SC	**98.11**	83.86	97.99	96.85	97.85	91.00
T1	**94.03**	90.74	91.91	91.98	92.88	78.56
TR	98.44	94.54	95.74	98.52	96.67	**98.78**
TP	**99.97**	98.11	99.93	99.88	79.47	51.64
YO	**85.17**	77.33	84.16	78.29	58.45	64.13
Average Decline Rate (%)		9.60	1.81	1.75	9.57	16.07

Moreover, we confirmed the effect of the *reconstruction loss*, which is essential for the MTM paradigm, in addition to our relation-preserving loss, by comparing MDTA to two ablation models: MDTA without reconstruction loss (*MDTA w/o* $\mathcal{L}_{RE}$) and MDTA without reconstruction and relation-preserving losses (*MDTA w/o* $\mathcal{L}_{RE}$ & $\mathcal{L}_{RP}$).

The classification performance of all ablation models and MDTA is listed in Table 3. Here, we present the results by averaging the accuracy scores across label ratios from 0.1 to 0.9 on each dataset under inductive inference. The complete results for all label ratios are given in Appendix E.

4.3.1. Dual-temporal encoder

In general, temporal dependencies, one of the unique characteristics of time series, span various time intervals within a time series [37,46]. Hence, reflecting diverse resolutions enables the model to easily recognize temporal dependencies, improving classification performance [5,44,47]. Thus, we designed the *dual-temporal encoder* f with the multi-resolution sub-encoder f_1 to allow the model to capture intrinsic temporal patterns at different time scales. To examine its effect, we compared the classification performance of MDTA and *MDTA w/o* f_1. As shown in Table 3, the average performance of *MDTA w/o* f_1 severely decreased by approximately 9.60% compared to that of MDTA. Thus, we demonstrated that the proposed encoder architecture can improve classification performance by successfully reflecting diverse temporal resolutions.

4.3.2. Relation-preserving loss function

Our encoder architecture, comprising two sequential sub-encoders, has one potential risk: temporal information obtained from the multi-resolution sub-encoder, f_1, can be corrupted by passing through the transformer-based sub-encoder, f_2. To address this risk, we introduced *relation-preserving*, $\mathcal{L}_{RP}$, which minimizes the temporal structural difference between representations generated by f_1 and f_2. Here, we investigate the ability of $\mathcal{L}_{RP}$ to mitigate temporal information loss and improve classification performance.

Information loss may be more severe when the number of layers in f_2 is greater. Thus, we constructed four models, each with either 5 or 20 layers in f_2, with or without the relation-preserving loss term. We trained the models with 100 epochs when the label ratio was 0.1. As shown in Figs. 1(a) and (b), the models without the relation-preserving loss function caused information loss between the two sub-encoders by mixing the representations of some classes when the number of layers was large. In this case, the model with 20 layers in f_2 achieved an accuracy score of 70%. By contrast, in Figs. 1(c) and (d), the models with the relation-preserving loss function learned the representations to successfully discriminate for all classes with an accuracy score of 96.67%, even after passing through f_2 with 20 layers.

In addition, we conducted a graphical analysis to demonstrate the necessity and effectiveness of the relation-preserving loss function for preventing the information loss that can be caused by the deep encoder architecture. Fig. 4 visualizes the data space and latent spaces produced by the sequential sub-encoders, f_1 and f_2, using UMAP [29] for the six largest datasets: *CBF*, *ECGFiveDays*, *MoteStrain*, *SwedishLeaf*, *TwoPatterns*, and *Yoga*. Here, we present each space in which the transformer-based sub-encoder f_2 has 20 layers without the relation-preserving loss function. As shown in Fig. 4, in most datasets, the representations of each class from the transformer-based sub-encoder f_2 showed diminished distinctiveness compared to those from the multi-resolution sub-encoder f_1, while achieving relatively low classification performance. In contrast, in Fig. 5, which illustrates each space for the six datasets when using the relation-preserving loss function, the representations of each class gradually formed more distinct groups for each class as they passed through each sub-encoder and achieved superior classification performance. Through this analysis, we validated the effectiveness of the proposed relation-preserving loss function.

Furthermore, as shown in Table 3, *MDTA w/o* $\mathcal{L}_{RP}$ exhibited 1.75% lower classification performance than MDTA on average. Therefore, we demonstrated that the relation-preserving loss function ensures a lossless flow of temporal information between two sub-encoders, enhancing classification performance.

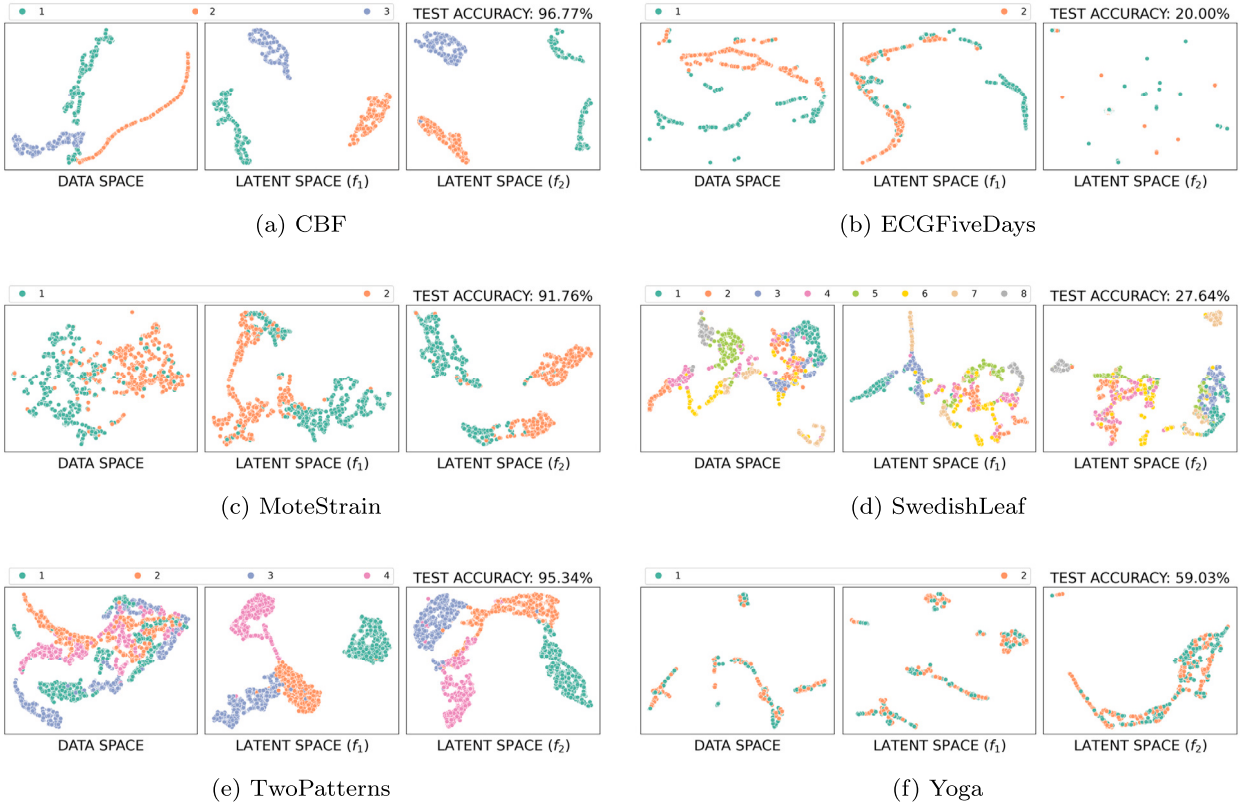


Fig. 4. Data space and latent spaces from the sequential sub-encoders, f_1 and f_2, without *relation-preserving* for the six largest datasets: (a) *CBF*, (b) *ECGFiveDays*, (c) *MoteStrain*, (d) *SwedishLeaf*, (e) *TwoPatterns*, and (f) *Yoga*.

4.3.3. Random masking ratios

A practical drawback of conventional MTM is its sensitivity to masking ratios [7,23]. In general, exploring the optimal masking ratios for each individual dataset within a reasonable time and cost is impractical. In addition, if we consider various masking ratios during model training, the information redundancy originating from the correlation between time steps can be eliminated from diverse perspectives, thereby creating a variety of challenging self-supervisory tasks that allow the model to identify sophisticated temporal relations. Therefore, we used random masking ratios to enhance the model generalization performance by identifying intricate temporal relations without the inefficiency of searching for appropriate masking ratios.

To demonstrate the effect of random masking ratios, in Fig. 6, we compared the random masking ratio with three fixed masking ratios of 0.2 (low), 0.5 (medium), and 0.8 (high). Although these fixed masking ratios may not be optimal for every dataset, we can examine the overall tendency of each dataset for low, medium, and high masking ratios. Consequently, when we fixed the masking ratio to a certain value, the performance varied greatly among the datasets. In contrast, the random masking ratio achieved the best performance in 12 of the 15 datasets and also showed decent performance in the remaining datasets.

In addition, Table 3 compares the average classification performance of the random masking ratio with that of the fixed ratio of 0.5 in more detail. As shown, the average classification performance of the fixed ratio of 0.5 showed a decrease of 1.81% compared to the random masking ratio. Specifically, as shown in Fig. E.4, which provides the complete results, the performance of the fixed masking ratio of 0.5 remarkably decreased by over 10% in several cases, especially for low label ratios on some datasets. Therefore, we demonstrated that the random masking ratio enhances the generalization performance of the model without requiring the high-cost tuning process for determining the optimal masking ratios.

Furthermore, the random masking ratios help the model to identify intricate temporal relations; therefore, the model is robust to missing values occurring in the inference phase because it can easily recover the missing parts. We validated the robustness of the proposed method against missing values compared with that of models with fixed masking ratios (see Appendix C).

4.3.4. Reconstruction loss function

In our approach, as described in Section 3.2.4, we update the model, which is trained by the classification loss function, with the reconstruction and relation-preserving loss functions to effectively leverage all accessible instances, including unlabeled instances. Thus, we examined the influence of the reconstruction loss, which is crucial for the MTM paradigm, along with the relation-preserving loss. As shown in Table 3, *MDTA w/o* $\mathcal{L}_{RE}$ and *MDTA w/o* $\mathcal{L}_{RE}$ & $\mathcal{L}_{RP}$ exhibited remarkably low performance compared to MDTA, with

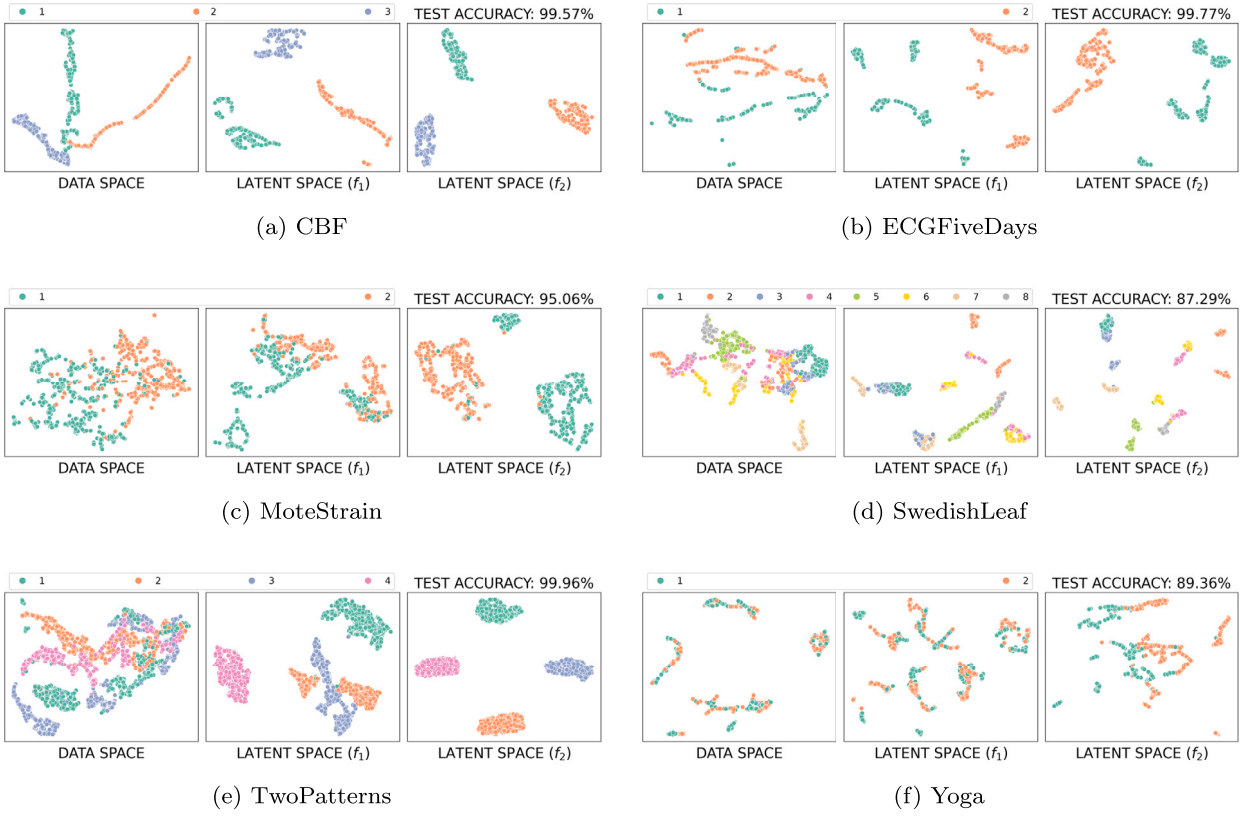


Fig. 5. Data space and latent spaces from f_1 and f_2 for the six largest datasets: (a) *CBF*, (b) *ECGFiveDays*, (c) *MoteStrain*, (d) *SwedishLeaf*, (e) *TwoPatterns*, and (f) *Yoga*.

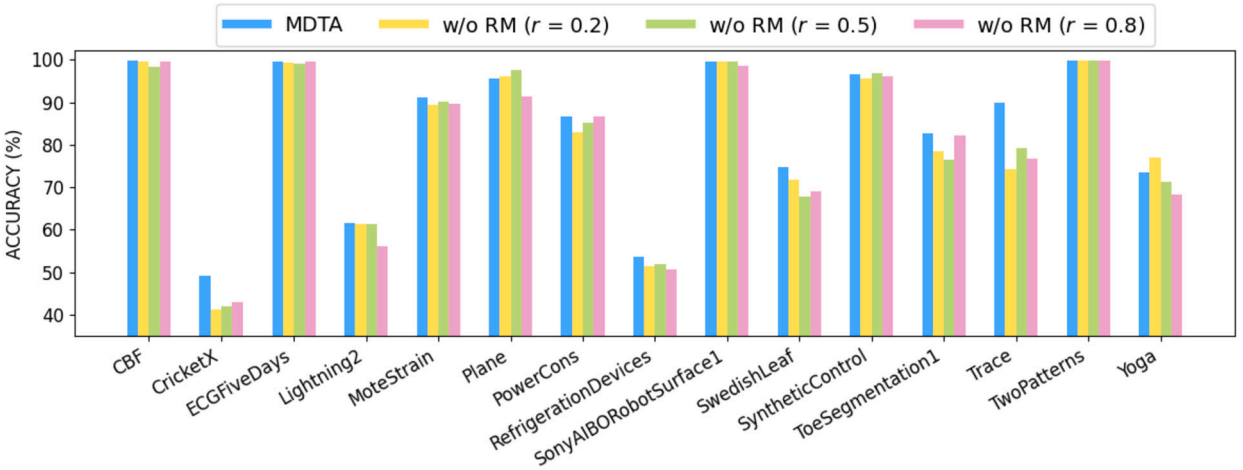


Fig. 6. Accuracy scores of MDTA and the ablation models using fixed masking ratios, r, of 0.2, 0.5, and 0.8, respectively. Here, we provide the scores when the label ratio is 0.1 because it best demonstrates the effect of masking ratios in capturing semantic information of time series.

9.57% and 16.07% decreases, respectively, on average. These results imply that the MTM paradigm, which uses the reconstruction loss in addition to the relation-preserving loss, helps enhance the model performance for semi-supervised time-series classification by effectively incorporating semantic information from unlabeled time series.

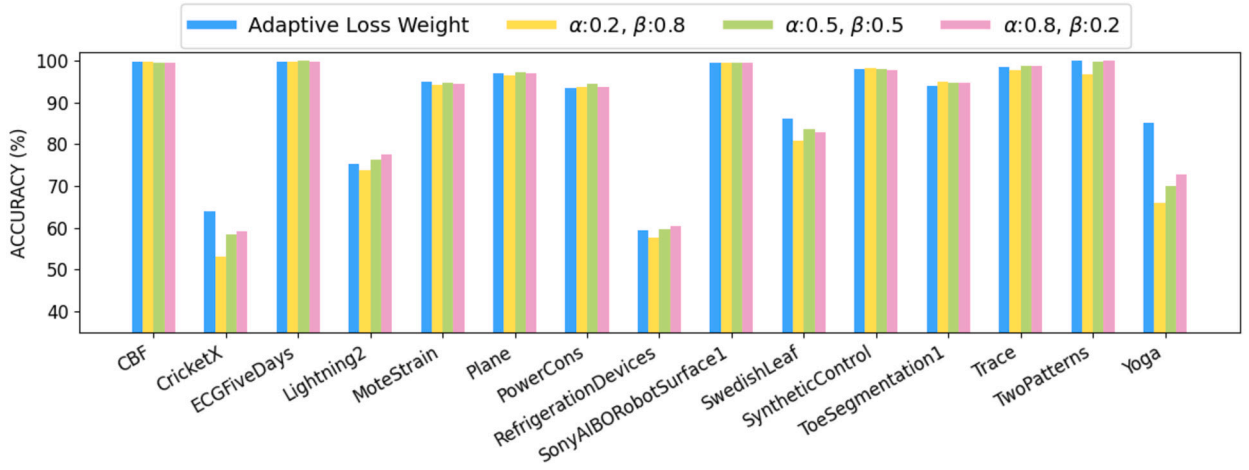


Fig. 7. Accuracy scores for different α and β values. Here, we compared the adaptive loss weighting strategy and three fixed loss weights.

4.4. Sensitivity analysis

To examine the influence of the loss weights α and β used in equation (10) on the MTM paradigm, we performed sensitivity analysis against them. We compared the adaptive loss weighting strategy [18] with three fixed loss weights. Here, the pairs of α and β for the fixed loss weights were set to (0.2, 0.8), (0.5, 0.5), and (0.8, 0.2). Consequently, as shown in Fig. 7, the adaptive loss weighting exhibited average performance comparable to that of the three fixed loss weights. However, in some datasets, such as *CricketX*, *SwedishLeaf*, and *Yoga*, adaptive loss weighting performed notably better than the others. In addition, it allows us to reduce the effort required to find the optimal values for α and β on each dataset. Therefore, we used the adaptive loss weighting strategy in the experiments performed in this study.

5. Conclusion

We propose a novel MTM-based framework, MDTA, for semi-supervised time-series classification. MDTA effectively captures the semantic information of a time series by leveraging the dual-temporal encoder, which effectively reflects diverse temporal resolutions, while introducing the relation-preserving loss function to prevent information loss within the encoder. In addition, the proposed method addresses the challenge of sensitivity to masking ratios by using random masking ratios. By incorporating the semantic information extracted from unlabeled instances with supervisory features obtained from labeled instances, our method enhances classification performance on time-series data. Through extensive experiments on semi-supervised time-series classification, we demonstrated that MDTA is effective for capturing semantic information in time series and performs better than SOTAs.

Nevertheless, as discussed in Appendix D, our method is relatively inefficient compared to the other baselines because it employs two consecutive sub-encoders, especially the transformer-based sub-encoder. In particular, for the transformer-based sub-encoder, we used the transformer architecture introduced by Zerveas et al. [46]; however, its computational cost is known to be high [6]. Therefore, we can improve the efficiency of our approach by replacing the transformer architecture with more recent alternatives, thereby improving computational efficiency.

Moreover, the multi-resolution sub-encoder can further increase computational complexity, especially as the dilation rate increases [48]. This incurs high memory and computational costs for both training and inference, particularly for temporal blocks configured with multiple *DilatedConv*s. Thus, another possible solution for future research to improve the computational efficiency of MDTA is to devise a lightweight transformer encoder architecture that can reflect diverse temporal resolutions.

CRediT authorship contribution statement

Sangho Lee: Writing – original draft, Visualization, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Chihyeon Choi:** Writing – original draft, Validation, Investigation, Data curation. **Youngdoo Son:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table A.1
Average classification performance across label ratios ranging from 0.1 to 0.9 for the baselines with different encoder architectures. The highest average score is highlighted in boldface for each baseline ($\mathbb{D}$: Dataset, *Trans*: transformer architecture).

$\mathbb{D}$	CE		Pseudo		Pi		FixMatch		MTL		SSTSC		iTimes	
	Trans	SimConv	*Trans*	SimConv	*Trans*	SimConv	*Trans*	SimConv	*Trans*	SimConv	*Trans*	SimConv	*Trans*	SimConv
CB	99.61	99.20	99.61	99.30	99.45	99.26	93.20	99.40	98.63	98.38	99.71	99.38	99.67	99.44
CX	31.44	52.26	31.38	52.62	62.17	61.92	26.60	59.66	24.29	40.38	31.67	41.89	36.38	67.04
EF	76.22	98.49	77.07	98.51	84.77	83.44	74.12	83.33	74.50	98.39	76.20	98.24	77.94	95.33
L2	60.89	67.56	63.38	68.89	68.18	68.98	59.11	69.87	60.71	67.56	67.02	67.64	75.20	71.56
MS	87.50	93.29	88.83	93.46	94.20	94.15	85.53	94.19	87.28	89.17	89.89	92.31	91.90	93.56
PL	89.10	96.98	87.88	96.98	85.45	85.98	48.57	88.73	69.52	84.87	75.03	94.50	52.22	85.66
PC	86.73	89.41	86.36	88.89	86.11	85.37	84.44	86.23	86.30	87.84	85.59	89.07	85.31	87.78
RD	61.69	58.60	62.46	57.61	57.13	57.04	58.99	57.32	61.87	57.51	62.50	58.19	60.21	60.64
SR	91.57	97.35	90.99	97.21	94.22	93.44	89.21	94.22	82.20	93.28	91.18	96.80	92.50	94.20
SL	60.38	84.16	62.51	84.83	71.92	70.34	22.82	69.66	31.86	55.45	57.91	76.93	52.55	56.18
SC	91.00	96.19	91.59	97.70	97.24	97.98	69.67	97.52	80.15	96.98	85.93	93.78	89.07	97.31
T1	78.56	84.16	78.48	84.44	85.10	84.73	76.34	84.36	72.80	82.30	76.50	82.39	78.19	86.71
TR	98.78	91.94	94.00	93.22	94.39	91.72	98.22	92.72	91.78	91.50	97.39	91.44	99.61	95.78
TP	51.64	99.81	61.02	99.85	99.34	99.30	28.41	99.48	83.52	98.91	89.18	99.73	72.68	96.86
YO	64.13	83.99	65.23	83.98	66.56	64.72	58.00	63.85	66.30	74.76	77.90	80.58	64.05	75.92
Average	75.28	**86.23**	76.05	**86.50**	**83.08**	82.56	64.88	**82.70**	71.45	**81.15**	77.57	**84.19**	75.17	**84.26**

Data availability

All datasets used in this study are publicly available.

Acknowledgements

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (MSIT) of Korea (No. RS-2023-00208412) and by the Ministry of Education (No. RS-2023-00271054).

Appendix A. Implementation details

Time-series datasets have been collected from various sources and their label ratios vary; hence, selecting optimal hyperparameters for each dataset is impractical. Thus, we used fixed hyperparameters regardless of the datasets and their label ratios to avoid impractical tuning [45]. In MDTA, the multi-resolution sub-encoder f_1 contains four temporal blocks, each comprising two *DilatedConv*s with GeLU activation functions [17] (see Fig. 3(a)), and skip connections were used between neighboring blocks. For the l-th block, the dilation rate ρ was set as 2^l. The kernel size was set to 3, each *DilatedConv* had a dimension of 16, and a residual block mapped the hidden features to d_u-dimensional temporal features $\boldsymbol{u}_i$. Subsequently, the sub-encoder f_2 consisted of a transformer block with batch normalization, as proposed by Zerveas et al. [46]. In particular, we configured f_2 with eight heads for multihead attention and three transformer blocks (see Fig. 3(b)). The dimensions of the two fully connected layers in each transformer block were set to 256 and d_z. Then, f_2 mapped the temporal features $\boldsymbol{u}_i$ to a d_z-dimensional representation $\boldsymbol{z}_i$. The decoder g was designed as a fully connected layer, reconstructing the semantic representation $\boldsymbol{z}_i$ generated by f_2 into d_u-dimensional temporal features $\hat{\boldsymbol{u}}_i$. The classification head h was constructed with two fully connected layers with batch normalization and the GeLU activation function with a hidden dimension of 256. This head produced predicted class labels $\hat{y}_i$ using $\boldsymbol{z}_i$ of $\boldsymbol{x}_i$ obtained by $f := f_2 \circ f_1$ as input. The dimensions d_u and d_z were both set to 64. The loss weights α and β used in equation (10) for the MTM paradigm were set differently at every epoch by the adaptive loss weighting strategy introduced in Heydari et al. [18]. Specifically, at each epoch j, the weights α^j and β^j for the loss functions $\mathcal{L}_{RP}^j$ and $\mathcal{L}_{RE}^j$ were calculated by

$$\alpha^j = \frac{e^{\eta(\mathcal{L}_{RP}^j - \mathcal{L}_{RP}^{j-1})}}{e^{\eta(\mathcal{L}_{RP}^j - \mathcal{L}_{RP}^{j-1})} + e^{\eta(\mathcal{L}_{RE}^j - \mathcal{L}_{RE}^{j-1})}}, \ \beta^j = \frac{e^{\eta(\mathcal{L}_{RE}^j - \mathcal{L}_{RE}^{j-1})}}{e^{\eta(\mathcal{L}_{RP}^j - \mathcal{L}_{RP}^{j-1})} + e^{\eta(\mathcal{L}_{RE}^j - \mathcal{L}_{RE}^{j-1})}},$$

where η is a hyperparameter that assigns greater weight to the worst-performing loss when $\eta > 0$ and to the best when $\eta < 0$. Here, we set η to 0.1.

Following Xi et al. [40] and Liu et al. [24], for the baselines, we employed a simple four-layer convolutional neural network (SimConv) with a ReLU activation function and batch normalization as the backbone encoder. In particular, the dimensions of the four layers were set to 8, 16, 32, and 64, respectively; the kernel size was set to 4, and the stride was set to 2 for every layer. The encoder architecture of the baselines differed from that of MDTA because of the better performance of SimConv compared to the transformer architecture in most baselines. Table A.1 shows the classification performance of the baseline methods by averaging accuracy scores across label ratios from 0.1 to 0.9 on each dataset. The classifier for the baselines was configured with the same architecture as the classification head h of the proposed method. We used time-warping and magnitude-warping augmentations for all baselines during model training [24,40]. The baseline methods, except iTimes, CA-TCC, and TS-TFC, were implemented based

Table B.2
Detailed description of five large-scale datasets. The data type, numbers of data and classes, and sequence lengths are provided for each dataset.

Dataset	Data Type	# of Data	# of Classes	Sequence Length
ECG5000	ECG	5000	5	140
Haptics	Motion	463	5	1092
HouseTwenty	Device	159	2	2000
Wafer	Sensor	7164	2	152
WormsTwoClass	Motion	258	2	900

Table B.3
Average classification performance across label ratios ranging from 0.1 to 0.9 on five large-scale datasets under inductive inference. For each dataset, the best score is highlighted in boldface, and the second-best value is underlined.

Dataset	CE	Pseudo	Π-model	FixMatch	MTL	SSTSC	iTimes	CA-TCC	TS-TFC	MDTA (ours)
ECG5000	89.13	93.25	92.32	91.74	92.84	93.59	92.90	**95.49**	94.05	94.34
Haptics	36.24	37.24	32.66	35.40	31.62	31.06	26.28	35.17	34.52	**38.07**
HouseTwenty	90.39	91.90	89.13	89.24	90.05	91.90	91.08	75.92	64.47	**92.01**
Wafer	89.39	99.19	89.85	89.41	98.18	98.58	96.69	99.42	**99.77**	99.50
WormsTwoClass	72.93	73.72	68.73	73.15	71.58	73.72	73.93	47.55	70.10	**75.00**

on the official code of SSTSC[2]; iTimes was implemented using the code provided by its authors,[3] and CA-TCC[4] and TS-TFC[5] were implemented based on their official code.

We set the batch size and maximum training epochs to 10 and 1000, respectively, for all methods, including MDTA. We used the Adam optimizer with weight decay (AdamW) [28] and a learning rate of 0.001 for model training. In addition, we adapted an early stopping strategy with a patience of 50 epochs based on the validation accuracy for efficient model training. We repeated the experiments five times and reported the average.

All experiments were executed using the PyTorch platform on a system with an Intel Core i9-10900X CPU at 3.70 GHz, 256 GB RAM, and a GeForce RTX 3090 24GB GPU.

Appendix B. Applicability on large-scale datasets

To validate the applicability of the proposed method to large-scale datasets, we applied MDTA to five additional datasets with larger numbers of instances or longer sequences: *ECG5000*, *Haptics*, *HouseTwenty*, *Wafer*, and *WormsTwoClass*. A detailed description of these datasets is preovided in Table B.2. In Table B.3, we provide the average classification performance across label ratios ranging from 0.1 to 0.9 for MDTA and the baselines on the five datasets. As shown, the proposed method outperformed the baselines on average, even on large-scale datasets, and achieved the best performance on three of the five datasets.

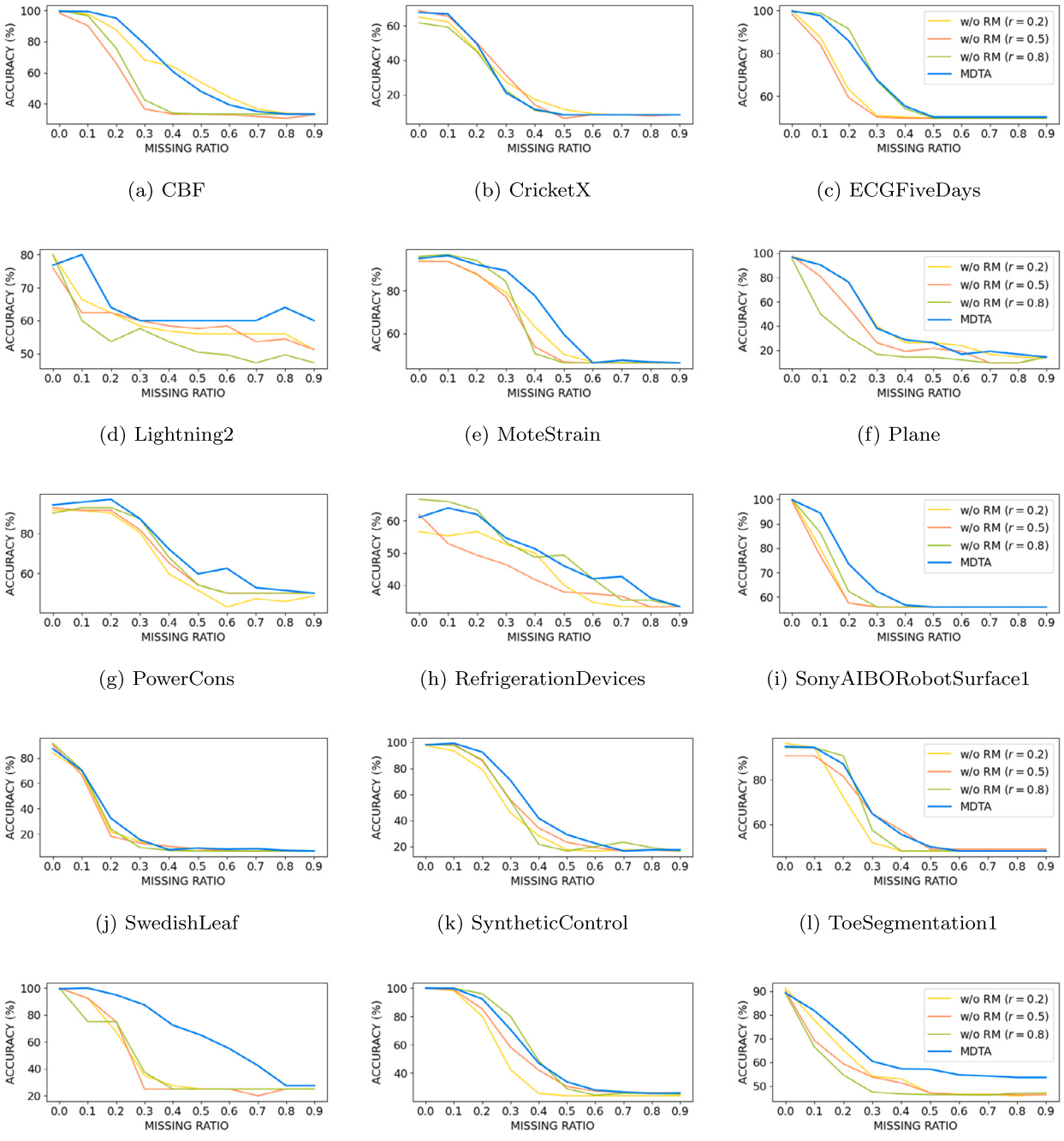
Appendix C. Robustness to missing values

Inherent temporal relations that are not easily identified can be captured by masking several time steps and predicting the masked parts. The model can then become robust to missing values because it can infer the missing parts relatively easily. Thus, regarding robustness against missing values, the MTM paradigm can prevent a drastic decrease in performance through captured temporal structures, even if some values are missing in the inference phase.

MDTA enhances the capability of capturing complex temporal relations within a time series using random masking ratios during model training. Thus, our method is more robust against missing values than MDTA with fixed masking ratios. To confirm this, we analyzed MDTA's robustness against missing values. We compared the classification performance of MDTA with those of the models with fixed masking ratios r of 0.2, 0.5, and 0.8 when the missing ratio varied from 0.1 to 0.9 for the input time series in the inference phase. The label ratio was set to 0.5.

Consequently, as shown in Fig. C.1, the proposed method was made more robust to missing values than the models with fixed masking ratios in most datasets by considering a wide range of masking ratios in the training phase. Particularly in *Trace*, MDTA maintained decent performance, even with high missing ratios. However, when it is difficult to capture temporal structures (e.g., missing ratios greater than 0.5), the model performance is likely to decrease, even when using random masking ratios.

[2] https://github.com/mrxiliang/sstsc.
[3] 1909030127@stu.hrbust.edu.cn.
[4] https://github.com/emadeldeen24/CA-TCC.
[5] https://github.com/ZLiu21/TS-TFC.



(a) CBF (b) CricketX (c) ECGFiveDays

(d) Lightning2 (e) MoteStrain (f) Plane

(g) PowerCons (h) RefrigerationDevices (i) SonyAIBORobotSurface1

(j) SwedishLeaf (k) SyntheticControl (l) ToeSegmentation1

(m) Trace (n) TwoPatterns (o) Yoga

Fig. C.1. Accuracy scores of MDTA and the ablation models using fixed masking ratios of 0.2, 0.5, and 0.8, respectively, when varying missing ratios $\in [0.1, 0.9]$.

Table D.4
Average training time (sec) per epoch of the baselines and MDTA across label ratios from 0.1 to 0.9 on each dataset.

Dataset	Pseudo	Π-model	FixMatch	MTL	SSTSC	iTimes	CA-TCC	TS-TFC	MDTA
CB	0.57	0.97	0.98	0.90	1.39	0.93	0.59	2.26	5.74
CX	0.48	0.83	0.85	0.79	1.19	0.80	0.52	2.09	4.83
EF	0.54	0.92	0.95	0.89	1.32	0.90	0.57	2.13	5.45
L2	0.09	0.14	0.15	0.14	0.21	0.14	0.57	1.02	0.79
MS	0.76	1.29	1.34	1.23	1.90	1.30	0.65	2.69	7.88
PL	0.13	0.22	0.23	0.21	0.31	0.21	0.75	1.14	1.25
PC	0.23	0.39	0.40	0.36	0.55	0.37	0.40	1.36	2.20
RD	0.48	0.85	0.88	0.80	1.19	0.83	0.52	2.04	5.01
SR	0.38	0.66	0.68	0.60	0.93	0.63	0.46	1.75	3.80
SL	0.71	1.21	1.24	1.13	1.73	1.16	0.66	2.56	6.95
SC	0.37	0.63	0.65	0.62	0.92	0.61	0.46	1.72	3.74
T1	0.17	0.29	0.30	0.28	0.42	0.28	0.37	1.24	1.65
TR	0.13	0.22	0.22	0.21	0.31	0.21	0.77	1.12	1.23
TP	3.05	5.06	5.68	4.81	7.77	4.86	1.99	8.39	31.94
YO	1.91	3.47	3.58	3.26	4.85	3.35	1.34	6.03	21.55

Appendix D. Computational efficiency

Table D.4 shows the average training time per epoch of the proposed method with that of baseline methods across label ratios from 0.1 to 0.9 on the 15 datasets listed in Table 1. MDTA operates more slowly than the other baselines because it sequentially employs two consecutive sub-encoders.

Specifically, in our work, we used the transformer architecture introduced in Zerveas et al. [46], the first transformer for extracting useful time-series representations, as the transformer-based sub-encoder f_2 in the dual-temporal encoder $f := f_2 \circ f_1$; its computational inefficiency was demonstrated in Cheng et al. [6]. Moreover, the multi-resolution sub-encoder f_1, which is configured with temporal blocks with several *DilatedConv*s, can further increase the computational complexity, particularly as the dilation rate increases [48]. This can result in high memory and computational costs for both training and inference, particularly for temporal blocks configured with multiple *DilatedConv*s. Therefore, the efficiency of the proposed approach should be improved in future work.

Appendix E. Complete experimental results

Here, we provide complete experimental results for the semi-supervised time-series classification presented in Section 4.2 and the ablation studies in Section 4.3.

E.1. Comparison with baselines

We evaluated the model performance for semi-supervised time-series classification under both inductive and transductive inference.

Fig. E.2 depicts the classification performance of our method compared to baselines for various label ratios $\in [0.1, 0.9]$. MDTA achieved outstanding performance compared to the baselines on most datasets, especially with low label ratios. In addition, the proposed method showed consistently high performance regardless of label ratios. These results demonstrate the effectiveness of MDTA in leveraging unlabeled instances in semi-supervised time-series classification.

Fig. E.3 presents the classification performance under *transductive* inference, which evaluates the model performance for unlabeled instances in the training dataset, for MDTA and the baselines. Although MDTA focuses on enhancing the model's generalization performance on the test datasets under the inductive setting, as shown in Fig. E.3, MDTA, along with the recent SOTA methods, CA-TCC and TS-TFC, performed better than the others under the transductive setting. Although CA-TCC and TS-TFC showed high performance on average, we observed that they tended to overfit the training dataset, thus achieving relatively lower performance under the inductive setting, unlike the transductive case. In contrast, MDTA exhibited superior performance in both settings, while achieving performance comparable to that of CA-TCC and TS-TFC under transductive inference.

E.2. Ablation studies

Fig. E.4 presents complete results on the ablation studies shown in Table 3 in Section 4.3. The performance of ablation models decreased compared to that of MDTA in most datasets. In particular, for *MDTA w/o* f_1, which replaces the multi-resolution sub-encoder with one fully connected layer, showed remarkable performance degradation compared with the proposed method. Therefore, we have demonstrated that each component of MDTA is essential to capture semantic information of time series effectively, enhancing model performance.

(a) CBF (b) CricketX (c) ECGFiveDays

(d) Lightning2 (e) MoteStrain (f) Plane

(g) PowerCons (h) RefrigerationDevices (i) SonyAIBORobotSurface1

(j) SwedishLeaf (k) SyntheticControl (l) ToeSegmentation1

(m) Trace (n) TwoPatterns (o) Yoga

Fig. E.2. Classification performance of baselines and MDTA for various label ratios $\in [0.1, 0.9]$ under *inductive* inference. The x-axis and y-axis denote label ratios and accuracy scores, respectively.

(a) CBF (b) CricketX (c) ECGFiveDays

(d) Lightning2 (e) MoteStrain (f) Plane

(g) PowerCons (h) RefrigerationDevices (i) SonyAIBORobotSurface1

(j) SwedishLeaf (k) SyntheticControl (l) ToeSegmentation1

(m) Trace (n) TwoPatterns (o) Yoga

Fig. E.3. Classification performance of baselines and MDTA for various label ratios $\in [0.1, 0.9]$ under *transductive* inference. The x-axis and y-axis denote label ratios and accuracy scores, respectively.

(a) CBF (b) CricketX (c) ECGFiveDays

(d) Lightning2 (e) MoteStrain (f) Plane

(g) PowerCons (h) RefrigerationDevices (i) SonyAIBORobotSurface1

(j) SwedishLeaf (k) SyntheticControl (l) ToeSegmentation1

(m) Trace (n) TwoPatterns (o) Yoga

Fig. E.4. Classification performance of ablation models and MDTA for various label ratios $\in [0.1, 0.9]$. The x-axis and y-axis denote label ratios and accuracy scores, respectively.

References

[1] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, preprint, arXiv:1803.01271, 2018.
[2] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C.A. Raffel, Mixmatch: a holistic approach to semi-supervised learning, Adv. Neural Inf. Process. Syst. 32 (2019).
[3] J. Cai, J. Hao, H. Yang, X. Zhao, Y. Yang, A review on semi-supervised clustering, Inf. Sci. 632 (2023) 164–200.
[4] L. de Carvalho Pagliosa, R.F. de Mello, Semi-supervised time series classification on positive and unlabeled problems using cross-recurrence quantification analysis, Pattern Recognit. 80 (2018) 53–63.
[5] P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, C. Guo, Multi-scale transformers with adaptive pathways for time series forecasting, in: The Twelfth International Conference on Learning Representations, 2023.
[6] M. Cheng, Q. Liu, Z. Liu, Z. Li, Y. Luo, E. Chen, Formertime: hierarchical multi-scale representations for multivariate time series classification, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 1437–1445.
[7] H. Choi, H. Park, K.M. Yi, S. Cha, D. Min, Salience-based adaptive masking: revisiting token dynamics for enhanced pre-training, preprint, arXiv:2404.08327, 2024.
[8] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.
[9] H.A. Dau, E. Keogh, K. Kamgar, C.C.M. Yeh, Y. Zhu, S. Gharghabi, C.A. Ratanamahatana, Y. Chen, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, Hexagon-ML, The ucr time series classification archive, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/, 2018.
[10] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, preprint, arXiv:1810.04805, 2018.
[11] J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, M. Long, Simmtm: a simple pre-training framework for masked time-series modeling, Adv. Neural Inf. Process. Syst. 36 (2024).
[12] E. Eldele, M. Ragab, Z. Chen, M. Wu, C.K. Kwoh, X. Li, C. Guan, Self-supervised contrastive representation learning for semi-supervised time-series classification, IEEE Trans. Pattern Anal. Mach. Intell. (2023) 1–15.
[13] A. Fan, T. Lavril, E. Grave, A. Joulin, S. Sukhbaatar, Addressing some limitations of transformers with feedback memory, preprint, arXiv:2002.09402, 2020.
[14] H. Fan, F. Zhang, R. Wang, X. Huang, Z. Li, Semi-supervised time series classification by temporal relation prediction, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 3545–3549.
[15] T. Han, W. Xie, Z. Pei, Semi-supervised adversarial discriminative learning approach for intelligent fault diagnosis of wind turbine, Inf. Sci. 648 (2023) 119496.
[16] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.
[17] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), preprint, arXiv:1606.08415, 2016.
[18] A.A. Heydari, C.A. Thompson, A. Mehmood, Softadapt: techniques for adaptive loss weighting of neural networks with multi-part loss functions, preprint, arXiv:1912.12355, 2019.
[19] S. Jawed, J. Grabocka, L. Schmidt-Thieme, Self-supervised learning for semi-supervised time series classification, in: Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24, Springer, 2020, pp. 499–511.
[20] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, in: International Conference on Learning Representations, 2016.
[21] D.H. Lee, et al., Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on Challenges in Representation Learning, ICML, Atlanta, 2013, p. 896.
[22] S. Lee, J. Choi, Y. Son, Efficient visibility algorithm for high-frequency time-series: application to fault diagnosis with graph convolutional network, Ann. Oper. Res. (2023) 1–21.
[23] Z. Li, Z. Rao, L. Pan, P. Wang, Z. Xu, Ti-mae: self-supervised masked time series autoencoders, preprint, arXiv:2301.08871, 2023.
[24] X. Liu, F. Zhang, H. Liu, H. Fan, itimes: investigating semisupervised time series classification via irregular time sampling, IEEE Trans. Ind. Inform. 19 (2022) 6930–6938.
[25] Z. Liu, Z. Lai, W. Ou, K. Zhang, H. Huo, Discriminative sparse least square regression for semi-supervised learning, Inf. Sci. 636 (2023) 118903.
[26] Z. Liu, Q. Ma, P. Ma, L. Wang, Temporal-frequency co-training for time series semi-supervised learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 8923–8931.
[27] Z. Liu, W. Pei, D. Lan, Q. Ma, Diffusion language-shapelets for semi-supervised time-series classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2024, pp. 14079–14087.
[28] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, 2018.
[29] L. McInnes, J. Healy, N. Saul, L. Großberger, Umap: uniform manifold approximation and projection, J. Open Sour. Softw. 3 (2018) 861.
[30] Q. Meng, H. Qian, Y. Liu, L. Cui, Y. Xu, Z. Shen, Mhccl: masked hierarchical cluster-wise contrastive learning for multivariate time series, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 9153–9161.
[31] Y. Nie, N.H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: long-term forecasting with transformers, in: The Eleventh International Conference on Learning Representations, 2022.
[32] M.N. Rizve, K. Duarte, Y.S. Rawat, M. Shah, In defense of pseudo-labeling: an uncertainty-aware pseudo-label selection framework for semi-supervised learning, in: International Conference on Learning Representations, 2021.
[33] A.A. Semenoglou, E. Spiliotis, V. Assimakopoulos, Data augmentation for univariate time series forecasting with neural networks, Pattern Recognit. 134 (2023) 109132.
[34] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C.A. Raffel, E.D. Cubuk, A. Kurakin, C.L. Li, Fixmatch: simplifying semi-supervised learning with consistency and confidence, Adv. Neural Inf. Process. Syst. 33 (2020) 596–608.
[35] Y. Son, S. Kang, Regression with re-labeling for noisy data, Expert Syst. Appl. 114 (2018) 578–587.
[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[37] Q. Wang, J. Wang, H. Deng, X. Wu, Y. Wang, G. Hao, Aa-trans: core attention aggregating transformer with information entropy selector for fine-grained visual classification, Pattern Recognit. 140 (2023) 109547.
[38] T. Wang, P. Isola, Understanding contrastive representation learning through alignment and uniformity on the hypersphere, in: International Conference on Machine Learning, PMLR, 2020, pp. 9929–9939.
[39] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, H. Xu, Time series data augmentation for deep learning: a survey, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021.
[40] L. Xi, Z. Yun, H. Liu, R. Wang, X. Huang, H. Fan, Semi-supervised time series classification model with self-supervised learning, Eng. Appl. Artif. Intell. 116 (2022) 105331.
[41] T. Xiao, X. Wang, A.A. Efros, T. Darrell, What should not be contrastive in contrastive learning, in: International Conference on Learning Representations, 2021.
[42] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, H. Hu, Simmim: a simple framework for masked image modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9653–9663.
[43] Z. Xu, Y. Bian, J. Zhong, X. Wen, Q. Xu, Beyond trend and periodicity: guiding time series forecasting with textual cues, preprint, arXiv:2405.13522, 2024.

[44] Y. Yang, J. Lu, Foreformer: an enhanced transformer-based framework for multivariate time series forecasting, Appl. Intell. 53 (2023) 12521–12540.
[45] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, B. Xu, Ts2vec: towards universal representation of time series, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 8980–8987.
[46] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, C. Eickhoff, A transformer-based framework for multivariate time series representation learning, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2114–2124.
[47] Y. Zhang, L. Ma, S. Pal, Y. Zhang, M. Coates, Multi-resolution time-series transformer for long-term forecasting, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2024, pp. 4222–4230.
[48] Z. Zhang, P. Zhang, Z. Xu, Q. Wang, Reduce computational complexity for convolutional layers by skipping zeros, in: 2023 IEEE 30th International Conference on High Performance Computing, Data, and Analytics (HiPC), IEEE, 2023, pp. 347–356.
[49] X. Zheng, T. Wang, W. Cheng, A. Ma, H. Chen, M. Sha, D. Luo, Parametric augmentation for time series contrastive learning, in: The Twelfth International Conference on Learning Representations, 2024.