SimMTM: A Simple Pre-Training Framework for Masked Time-Series Modeling

Jiaxiang Dong [*1] **Haixu Wu** [*1] **Haoran Zhang** [1] **Li Zhang** [1] **Jianmin Wang** [1] **Mingsheng Long** [1]

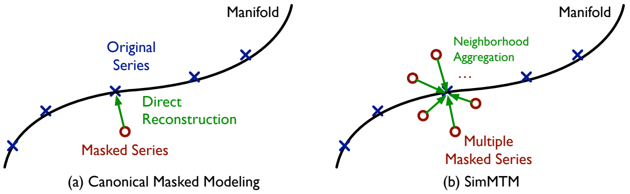
Abstract

Time series analysis is widely used in extensive areas. Recently, to reduce labeling expenses and benefit various tasks, self-supervised pre-training has attracted immense interest. One mainstream paradigm is masked modeling, which successfully pre-trains deep models by learning to reconstruct the masked content based on the unmasked part. However, since the semantic information of time series is mainly contained in temporal variations, the standard way of randomly masking a portion of time points will ruin vital temporal variations of time series seriously, making the reconstruction task too difficult to guide representation learning. We thus present SimMTM, a Simple pre-training framework for Masked Time-series Modeling. By relating masked modeling to manifold learning, SimMTM proposes to recover masked time points by the weighted aggregation of multiple neighbors outside the manifold, which eases the reconstruction task by assembling ruined but complementary temporal variations from multiple masked series. SimMTM further learns to uncover the local structure of the manifold helpful for masked modeling. Experimentally, SimMTM achieves state-of-the-art fine-tuning performance in two canonical time series analysis tasks: forecasting and classification, covering both in- and cross-domain settings.

1. Introduction

Time series analysis has attached immense importance in extensive real applications, such as financial analysis, energy planning and etc (Wu et al., 2021; Xu et al., 2021). Vast amounts of time series are incrementally collected from IoT and wearable devices. However, the semantic information of time series is mainly buried in human-indiscernible temporal variations, making it difficult to annotate. Recently, self-supervised pre-training has been widely explored (Liu et al., 2021; Jiang et al., 2022), which benefits deep models from pretext knowledge learned over large-scale unlabeled data and further promotes the performance of various downstream tasks. Especially, as a well-recognized pre-training paradigm, masked modeling has achieved great successes in many areas, such as masked language modeling (MLM) (Devlin et al., 2018; Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020; Gao et al., 2020) and masked image modeling (MIM) (He et al., 2022; Xie et al., 2022b; Li et al., 2022). This paper extends pre-training methods to time series, especially masked time-series modeling (MTM).

The canonical technique of masked modeling is to optimize the model by learning to reconstruct the masked content based on the unmasked part (Devlin et al., 2018). However, unlike images and natural languages whose patches or words contain abundant even redundant semantic information, we observe that the valuable semantic information of time series is mainly contained in the temporal variations, such as the trend, periodicity and peak valley, which can correspond to weather processes, abnormal faults or etc in the real world. Therefore, directly masking a portion of time points will ruin the temporal variations of the original time series seriously, which as a result makes the reconstruction task too difficult to guide representation learning of time series.



(a) Canonical Masked Modeling

(b) SimMTM

Figure 1. A manifold perspective for understanding SimMTM.

According to the analysis in stacked denoising autoencoders (Vincent et al., 2010), as shown in Figure 1, we can view the randomly masked series as the "neighbor" of the original time series outside the manifold and the reconstruction process is to project the masked series back to the manifold of original series. However, as we analyzed above, direct reconstruction may fail since the essential temporal variations are ruined by random masking. Inspired by the manifold per-

[*] Equal contribution [1] School of Software, BNRist, Tsinghua University. Jiaxiang Dong <djx20@mails.tsinghua.edu.cn>. Haixu Wu <whx20@mails.tsinghua.edu.cn>. Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

Preliminary work.

spective, we go beyond the direct reconstruction convention of masked modeling and propose a natural idea as reconstructing the original data from its *multiple* "neighbors", i.e. multiple masked series. Although the temporal variations of the original time series have been partially dropped in each randomly masked series, the multiple randomly masked series will complement each other, making the reconstruction process much easier than directly reconstructing the original series from a single masked series. This process will also pre-train the model to uncover the local structure of the time series manifold implicitly, thereby benefiting masked modeling and representation learning (Schroff et al., 2015; Wang & Isola, 2020).

Based on the above insights, in this paper, we propose the SimMTM as a simple but effective pre-training framework for time series. Instead of directly reconstructing the masked time points from unmasked parts, SimMTM recovers the original time series from multiple randomly masked time series. Technically, SimMTM presents a neighborhood aggregation design for reconstruction, which is to aggregate the point-wise representations of time series based on the similarities learned in the series-wise representation space. In addition to the reconstruction loss, a constraint loss is presented to guide the series-wise representation learning based on the neighborhood assumption of the time series manifold. Benefiting from the above designs, SimMTM achieves consistent state-of-the-art in various time series analysis tasks when fine-tuning the pre-trained model into downstream tasks, covering both the low-level forecasting and high-level classification tasks, even if there is a clear domain shift between the pre-training and fine-tuning datasets. Overall, our contributions can be summarized as follows:

- Inspired by the manifold perspective of masking, we propose a new task for masked time-series modeling, which reconstructs the original series on the manifold based on multiple masked series outside the manifold.
- Technically, we present SimMTM as a simple but effective pre-training framework, which aggregates point-wise representations for reconstruction based on the similarities learned in series-wise representation space.
- SimMTM consistently achieves state-of-the-art fine-tuning performance in typical time series analysis tasks, including low-level forecasting and high-level classification, covering both in- and cross-domain settings.

2. Related Work

2.1. Self-supervised Pre-training

Self-supervised pre-training is an important research topic for learning generalizable and shared knowledge from large-scale data and benefiting downstream tasks. Firstly, this topic has been widely explored in computer vision and natural language processing. Elaborative manually-designed self-supervised tasks are presented, which can be roughly categorized into contrastive learning (He et al., 2020; Chen et al., 2020) and masked modeling (Devlin et al., 2018; He et al., 2022). Recently, following previous contrastive learning and masked modeling paradigms, some self-supervised pre-training methods for time series have been proposed (Franceschi et al., 2019; Sarkar & Etemad, 2020; Rebjock et al., 2021; Sun et al., 2021; Yang & Hong, 2022).

Contrastive learning. The key insight of contrastive learning is to optimize the representation space based on the manually designed positive and negative pairs, where representations of positive pairs are optimized to be close to each other while negative ones tend to be far apart (Wu et al., 2018; Jaiswal et al., 2020). The canonical design presented in SimCLR (Tang et al., 2020) views the different augmentations of the same sample as positive pairs and the augmentations among different samples as negative pairs.

Recently, in time series pre-training, many designs of positive and negative pairs have been proposed by utilizing the invariant properties of time series. Concretely, to make the representation learning seamlessly related to temporal variations, TimCLR (Yang et al., 2022) adopts the DTW (Mueen & Keogh, 2016) to generate phase-shift and amplitude-change augmentations, which is more suitable for time series. TS2Vec (Yue et al., 2022) splits multiple time series into several patches and further defines the contrastive loss in both instance-wise and patch-wise aspects. TS-TCC (Eldele et al., 2021) presents a new temporal contrastive learning task as making the augmentations predict each other's future. TF-C (Zhang et al., 2022) proposes a novel time-frequency consistency architecture and optimizes time-based and frequency-based representations of the same example to be close to each other. Mixing-up (Wickstrøm et al., 2022) exploits a data augmentation scheme in which new samples are generated by mixing two data samples and the model is optimized to predict the mixing weights. Note that contrastive learning mainly focuses on the high-level information (Xie et al., 2022a) and the series-wise or patch-wise representations inherently mismatch the low-level tasks, such as time series forecasting. Thus, in this paper, we focus on the masked modeling paradigm.

Masked modeling. The masked modeling paradigm optimizes the model by learning to reconstruct the masked content from unmasked part. This paradigm has been widely explored in computer vision and natural language processing, which is to predict the masked words of a sentence (Devlin et al., 2018) and masked patches of an image (He et al., 2022; Xie et al., 2022b) respectively.

As for the time series analysis, TST (Zerveas et al., 2021) directly adopts the canonical masked modeling paradigm, which is learning to predict the removed time points based on the remaining time points. Afterward, PatchTST (Nie et al., 2022) learns to predict the masked subseries-level patches to capture the local semantic information and reduce memory usage. However, as we stated before, directly masking time series will ruin the essential temporal variations, making the reconstruction too difficult to guide the representation learning. Unlike the direct reconstruction in previous works, SimMTM presents a new masked modeling task, which is reconstructing the original time series from multiple randomly masked series.

2.2. Understanding Masked Modeling

Masked modeling has been explored in stacked denoising autoencoders (Vincent et al., 2010), where the masking is viewed as adding noise to the original data and the masked modeling is to project the masked data from the neighborhood back to the original manifold, namely denoising. Recently, it has been widely used in pre-training, which can learn valuable low-level information from data unsupervisedly (Xie et al., 2022a). Inspired by the manifold perspective, we go beyond the classical denoising process and project the masked data back to the manifold by aggregating multiple masked time series within the neighborhood.

3. SimMTM

As aforementioned, to tackle the problem that temporal variations are ruined by random masking, SimMTM proposes to reconstruct the original time series from multiple masked time series. To implement this, SimMTM first learns similarities among multiple time series in the series-wise representation space and then aggregates the point-wise representations of these time series based on the pre-learned series-wise similarities. Next, we will detail the techniques in both model architecture and pre-training protocol aspects.

3.1. Overall Architecture

The reconstruction process of SimMTM involves the following four modules: masking, representation learning, series-wise similarity learning and point-wise reconstruction.

Masking. Given $\{\mathbf{x}_i\}_{i=1}^N$ as a mini-batch of N time series samples, where $\mathbf{x}_i \in \mathbb{R}^{L\times C}$ contains L time points and C observed variates, we can easily generate a set of masked series for each sample $\mathbf{x}_i$ by randomly masking a portion of time points along the temporal dimension, which can be formalized as follows:

$$\{\overline{\mathbf{x}}_i^j\}_{j=1}^M = \mathrm{Mask}_r(\mathbf{x}_i), \tag{1}$$

where $r \in [0,1]$ denotes the masked portion. M is a hyperparameter for the number of masked time series. $\overline{\mathbf{x}}_i^j \in \mathbb{R}^{L\times C}$ represents the j-th masked time series of $\mathbf{x}_i$, where the values of masked time points are replaced by zeros. Then we can obtain a batch of augmented time series. For clarity, we present all the $(N\times(M+1))$ input series in a set as follows:

$$\mathcal{X} = \bigcup_{i=1}^N \left(\{\mathbf{x}_i\} \cup \{\overline{\mathbf{x}}_i^j\}_{j=1}^M\right). \tag{2}$$

Representation learning. After the encoder and projector layer, we can obtain the point-wise representations $\mathcal{Z}$ and series-wise representations $\mathcal{S}$, which is formalized by:

$$\begin{aligned} \mathcal{Z} &= \bigcup_{i=1}^N \left(\{\mathbf{z}_i\} \cup \{\overline{\mathbf{z}}_i^j\}_{j=1}^M\right) = \mathrm{Enocder}(\mathcal{X}) \\ \mathcal{S} &= \bigcup_{i=1}^N \left(\{\mathbf{s}_i\} \cup \{\overline{\mathbf{s}}_i^j\}_{j=1}^M\right) = \mathrm{Projector}(\mathcal{Z}), \end{aligned} \tag{3}$$

where $\mathbf{z}_i, \overline{\mathbf{z}}_i^j \in \mathbb{R}^{L\times d_{\text{model}}}$ and $\mathbf{s}_i, \overline{\mathbf{s}}_i^j \in \mathbb{R}^{1\times d_{\text{model}}}$. We employ a simple MLP layer along the temporal dimension as the $\mathrm{Projector}(\cdot)$ to obtain series-wise representations. As for the $\mathrm{Encoder}(\cdot)$, we adopt the encoder part of Transformer (Vaswani et al., 2017), which will be transferred to downstream tasks during the fine-tuning process.

Series-wise similarity learning. Note that directly averaging multiple masked time series will result in the oversmoothing problem (Vincent et al., 2010). Thus, to precisely reconstruct the original time series, we attempt to utilize the similarities among series-wise representations $\mathcal{S}$ for weighted aggregation, namely exploiting the local structure of the time series manifold. For simplification, we formalize the calculation of series-wise similarities as:

$$\mathbf{R} = \mathrm{Sim}(\mathcal{S}), \tag{4}$$

where $\mathbf{R} \in \mathbb{R}^{(N\times(M+1))\times(N\times(M+1))}$ is the matrix of pairwise similarities for $(N\times(M+1))$ input samples in series-wise representation space, which are measured by the cosine distance. Concretely, for series-wise representations $\mathbf{u}, \mathbf{v} \in \mathcal{S}$, their similarity is calculated by $\mathbf{R}_{\mathbf{u},\mathbf{v}} = \frac{\mathbf{u}\mathbf{v}^{\mathsf{T}}}{\|\mathbf{u}\|\|\mathbf{v}\|}$.

Point-wise aggregation. As shown in Figure 2, based on the learned series-wise similarities, the aggregation process for the i-th original time series is:

$$\widehat{\mathbf{z}}_i = \sum_{\mathbf{s}'\in\mathcal{S}\backslash\{\mathbf{s}_i\}} \frac{\exp(\mathbf{R}_{\mathbf{s}_i,\mathbf{s}'}/\tau)}{\sum_{\mathbf{s}''\in\mathcal{S}\backslash\{\mathbf{s}_i\}}\exp(\mathbf{R}_{\mathbf{s}_i,\mathbf{s}''}/\tau)}\mathbf{z}', \tag{5}$$

where τ denotes the temperature hyperparameter. $\mathbf{z}'$ represents the corresponding point-wise representation of $\mathbf{s}'$ and $\widehat{\mathbf{z}}_i \in \mathbb{R}^{L\times d_{\text{model}}}$ is the reconstructed point-wise representation. After the decoder, we can obtain the reconstructed original time series, namely

$$\{\widehat{\mathbf{x}}_i\}_{i=1}^N = \mathrm{Decoder}(\{\widehat{\mathbf{z}}_i\}_{i=1}^N), \tag{6}$$

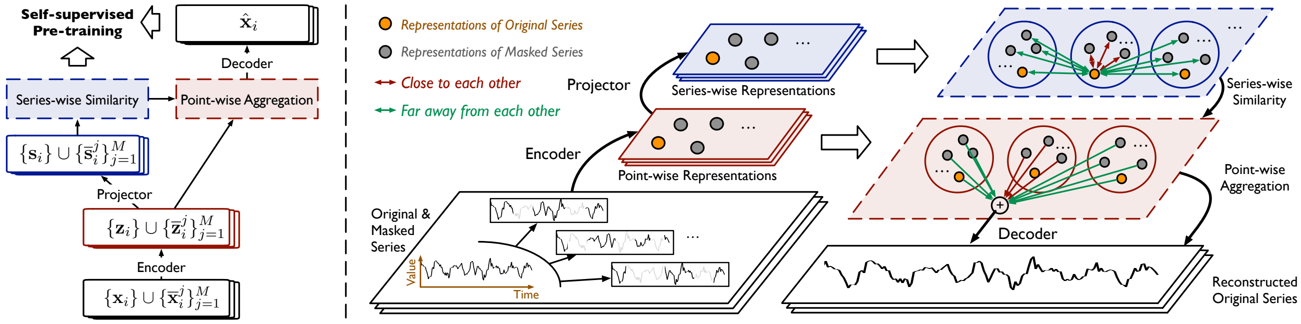

Self-supervised Pre-training
$\hat{\mathbf{x}}_i$
Decoder
Series-wise Similarity
Point-wise Aggregation
$\{\mathbf{s}_i\} \cup \{\overline{\mathbf{s}}_i^j\}_{j=1}^M$
Projector
$\{\mathbf{z}_i\} \cup \{\overline{\mathbf{z}}_i^j\}_{j=1}^M$
Encoder
$\{\mathbf{x}_i\} \cup \{\overline{\mathbf{x}}_i^j\}_{j=1}^M$
Representations of Original Series
Representations of Masked Series
Close to each other
Far away from each other
Projector
Series-wise Representations
Encoder
Point-wise Representations
Original & Masked Series
Value
Time
Series-wise Similarity
Point-wise Aggregation
Decoder
Reconstructed Original Series

Figure 2. Overall Architecture of SimMTM, which reconstructs the original time series by aggregating multiple masked time series.

where $\widehat{\mathbf{x}}_i \in \mathbb{R}^{L \times C}$ is the reconstruction to $\mathbf{x}_i$. $\mathrm{Decoder}(\cdot)$ is instantiated as a simple MLP layer along the channel dimension following (Xie et al., 2022b).

3.2. Self-supervised Pre-training

Following the masked modeling paradigm, SimMTM is supervised by a reconstruction loss:

$$\mathcal{L}_{\text{reconstruction}} = \sum_{i=1}^{N} \|\mathbf{x}_i - \widehat{\mathbf{x}}_i\|_2^2. \tag{7}$$

Note that the reconstruction process is directly based on the series-wise similarities, while it is hard to guarantee the model captures the precise similarities without explicit constraints in the series-wise representation space. Thus, to avoid trivial aggregation, we also utilize the neighborhood assumption of the time series manifold to calibrate the structure of series-wise representation space $\mathcal{S}$. For clarity, we formalize the neighborhood assumption as follows:

$$\begin{aligned} \left(\{\mathbf{s}_i\} \cup \{\overline{\mathbf{s}}_i^j\}_{j=1}^M\right) &\sim \left(\{\mathbf{s}_i\} \cup \{\overline{\mathbf{s}}_i^j\}_{j=1}^M\right) \\ \left(\{\mathbf{s}_i\} \cup \{\overline{\mathbf{s}}_i^j\}_{j=1}^M\right) &\nsim \left(\{\mathbf{s}_k\} \cup \{\overline{\mathbf{s}}_k^j\}_{j=1}^M\right), i \neq k \end{aligned} \tag{8}$$

where $\sim$ and $\nsim$ mean the elements among two sets are assumed as close to and far away from each other respectively. Eq. (8) indicates that the original time series and its masked series will present close representations and be far away from the representations from other series in $\mathcal{S}$. For each series-wise representation $\mathbf{s} \in \mathcal{S}$, we define the set of its assumed close series as $\mathbf{s}^+ \subset \mathcal{S}$. Note that to avoid the dominating representation, we assume that $\mathbf{s} \notin \mathbf{s}^+$. With the above formalization, we can define manifold constraint to series-wise representation space as

$$\mathcal{L}_{\text{constraint}} = -\sum_{\mathbf{s} \in \mathcal{S}} \left(\sum_{s' \in \mathbf{s}^+} \log \frac{\exp(\mathbf{R}_{\mathbf{s},\mathbf{s}'}/\tau)}{\sum_{\mathbf{s}'' \in \mathcal{S} \setminus \{\mathbf{s}\}} \exp(\mathbf{R}_{\mathbf{s},\mathbf{s}''}/\tau)} \right), \tag{9}$$

which can optimize the learned series-wise representation to satisfy the neighborhood assumption in Eq. (8) better.

Finally, the overall optimization process of SimMTM can be represented as follows:

$$\min_{\Theta} \mathcal{L}_{\text{reconstruction}} + \lambda \mathcal{L}_{\text{constraint}}, \tag{10}$$

where Θ denotes the set of all parameters of the deep architecture. To trade off the two parts in Eq. (10), we adopt the tuning strategy presented by Kendall et al., which can adjust the hyperparameters λ adaptively according to the homoscedastic uncertainty of each loss.

4. Experiments

To fully evaluate SimMTM, we conduct experiments on two typical time series analysis tasks: forecasting and classification, which covers the learning of both low-level and high-level representations. Further, for each task, we present the model fine-tuning performance under both in- and cross-domain settings.

Benchmarks. We summarize the experiment benchmarks in Table 1, which involves nine real-world datasets in total, covering two mainstream time series analysis tasks: time series forecasting and classification. The detailed descriptions for each dataset are provided in Appendix A.1

Table 1. Summary of experiment benchmarks.

TASKS	DATASETS	SEMANTIC INFORMATION
FORECAST.	ETTH1 ETTH2	HOURLY ELECTRICITY DATA
	ETTM1 ETTM2	15-MINUTELY ELECTRICITY DATA
CLASSIFY.	SLEEPEEG	EEG DATA
	EPILEPSY	EEG DATA
	FD-B	FAULTY DETECTION FOR SYSTEMS
	GESTURE	PATHS OF HAND MOVEMENT
	EMG	SIGNAL OF MUSCLE RESPONSES

Baselines. We compare SimMTM with five competitive self-supervised time series pre-training methods, including the contrastive learning methods: TF-C (2022), TS-TCC (2021), Mixing-up (2022), TS2Vec (2022), and the masked

Table 2. In-domain setting of forecasting the future O time points based on the past 96 time points. All results are averaged from 4 different choices of $O \in \{96, 192, 336, 720\}$. A smaller MSE or MAE indicates a better prediction. Full results can be found in Table 12.

Models	Deep Model		Contrastive								Masking					
	NsTrans. (2022)		TF-C (2022)		TS-TCC (2021)		Mixing-up (2022)		TS2Vec (2022)		TST (2021)		Random Init.		**SimMTM (Ours)**	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.570	0.537	1.162	0.863	1.152	0.857	1.098	1.933	0.897	0.752	0.624	0.562	0.605	0.549	**0.497**	**0.476**
ETTh2	0.526	0.516	2.850	1.349	3.101	1.509	2.723	1.348	2.628	1.381	0.429	0.458	0.457	0.455	**0.415**	**0.428**
ETTm1	0.481	0.456	0.744	0.652	1.298	0.893	0.734	0.635	0.669	0.600	0.494	0.471	0.478	0.464	**0.414**	**0.422**
ETTm2	0.306	0.347	1.755	0.947	1.153	0.857	1.420	0.912	1.466	0.957	0.425	0.371	0.416	0.388	**0.302**	**0.342**

Table 3. Cross-domain setting of forecasting the future O time points based on the past 96 time points. All results are averaged from 4 different choices of $O \in \{96, 192, 336, 720\}$. A lower MSE or MAE indicates a better prediction. Full results are in Table 13.

Models	Contrastive								Masking					
	TF-C (2022)		TS-TCC (2021)		Mixing-up (2022)		TS2Vec (2022)		TST (2021)		Random Init.		**SimMTM (Ours)**	
Scenario	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2 → ETTh1	1.135	0.834	1.151	0.856	1.116	0.854	0.944	0.744	0.645	0.535	0.605	0.549	**0.499**	**0.479**
ETTm1 → ETTh1	1.200	0.877	1.162	0.864	1.094	0.825	0.957	0.768	0.645	0.533			**0.507**	**0.483**
ETTm2 → ETTh1	1.091	0.814	1.119	0.839	1.137	0.855	0.896	0.726	0.632	0.576			**0.508**	**0.484**
ETTh1 → ETTm1	0.746	0.652	0.699	0.625	0.731	0.632	0.697	0.616	0.482	0.444	0.478	0.464	**0.416**	**0.421**
ETTh2 → ETTm1	0.750	0.654	0.714	0.641	0.709	0.620	0.606	0.556	0.472	0.448			**0.424**	**0.426**
ETTm2 → ETTm1	0.758	0.669	0.700	0.630	0.734	0.634	0.756	0.638	0.480	0.455			**0.420**	**0.422**

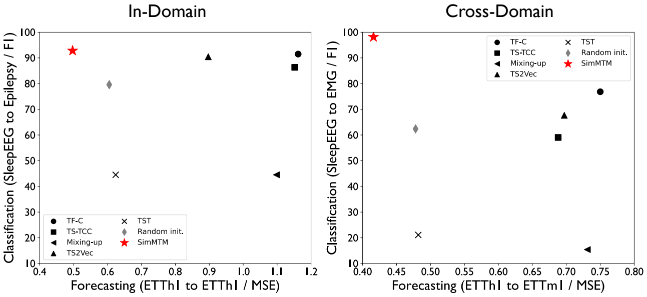


Figure 3. Performance comparison of time series pre-training methods in forecasting (MSE ↓) and classification (F1 ↑) tasks, including both in-domain (left) and cross-domain (right) settings.

modeling method: TST (2021). Besides, to demonstrate the generality of SimMTM, we also apply three advanced time series foundation models as the encoder, including NSTransformer (2022), Autoformer (2021), and vanilla Transformer (2017), where NSTransformer is the state-of-the-art model for time series forecasting. Without special clarifications, we adopt the vanilla Transformer (2017) as the encoder for forecasting. As for the classification, we use the 1D-ResNet (2016) following (Zhang et al., 2022).

Implementations. We present the fine-tuning performance under both in- and cross-domain settings. For the in-domain setting, we pre-train and fine-tune the model using the same or same-domain dataset. Especially for the classification task, since the SleepEEG and Epilepsy present quite similar semantic information, we view the "pre-training on SleepEEG and fine-tuning on Epilepsy" as the in-domain task, which is denoted as SleepEEG → Epilepsy for clarity. As for the cross-domain setting, we pre-train the model on a certain dataset and fine-tune the encoder to different datasets. More implementation details can be found in Appendix A.

4.1. Main results

We summarize the model performance in forecasting and classification tasks of in- and cross-domain settings in Figure 3. A lower MSE means better forecasting performance (x-axis of Figure 3), and a higher F1 means better classification performance (y-axis). In all these settings, SimMTM outperforms other baselines significantly. It is also notable that although the masking-based method TST (2021) achieves good performance in the forecasting task, it fails in the classification task. Besides, the previous contrastive-based methods fail in low-level forecasting tasks but perform well in high-level classification tasks. These results indicate that previous methods cannot cover both the high-level and low-level tasks simultaneously, highlighting the advantages of SimMTM in task generality.

Table 4. In-domain setting of classification. We pre-train the model on SleepEEG and then fine-tune it on the same-domain dataset: Epilepsy. Accuracy (*Acc. %*), Precision (*Pre. %*), Recall (*Rec. %*), F1 score (*F1. %*), and their average value (*Avg. %*) are recorded.

Scenarios	Models		Acc.	Pre.	Rec.	F1.	Avg.
SleepEEG ↓ Epilepsy	Random Init.		89.83	92.13	74.47	79.59	84.00
	Contrastive	TS2vec (Yue et al., 2022)	93.95	90.59	90.39	90.45	91.35
		Mixing-up (Wickstrøm et al., 2022)	80.21	40.11	50.00	44.51	53.71
		TS-TCC (Eldele et al., 2021)	92.53	94.51	81.81	86.33	88.80
		TF-C (Zhang et al., 2022)	94.95	**94.56**	89.08	91.49	92.52
	Masking	TST (Zerveas et al., 2021)	80.21	40.11	50.00	44.51	53.71
	SimMTM (Ours)		**95.49**	93.36	**92.28**	**92.81**	**93.49**

Table 5. Cross-domain setting of classification. We pre-train a model on SleepEEG and fine-tune it to multiple target datasets. Accuracy (*Acc. %*), Precision (*Pre. %*), Recall (*Rec. %*), F1 score (*F1. %*), and their average value (*Avg. %*) are recorded.

Scenarios	Models		Acc.	Pre.	Rec.	F1.	Avg.
SleepEEG ↓ FD-B	Random Init.		47.36	48.29	52.35	49.11	49.28
	Contrastive	TS2vec (Yue et al., 2022)	47.90	43.39	48.42	43.89	45.90
		Mixing-up (Wickstrøm et al., 2022)	67.89	71.46	76.13	72.73	72.05
		TS-TCC (Eldele et al., 2021)	54.99	52.79	63.96	54.18	56.48
		TF-C (Zhang et al., 2022)	69.38	**75.59**	72.02	74.87	72.97
	Masking	TST (Zerveas et al., 2021)	46.40	41.58	45.50	41.34	43.71
	SimMTM (Ours)		**69.40**	74.18	**76.41**	**75.11**	**73.78**
SleepEEG ↓ Gesture	Random Init.		42.19	47.51	49.63	48.86	47.05
	Contrastive	TS2vec (Yue et al., 2022)	69.17	65.45	68.54	65.70	67.22
		Mixing-up (Wickstrøm et al., 2022)	69.33	67.19	69.33	64.97	67.71
		TS-TCC (Eldele et al., 2021)	71.88	71.35	71.67	69.84	71.19
		TF-C (Zhang et al., 2022)	76.42	77.31	74.29	75.72	75.94
	Masking	TST (Zerveas et al., 2021)	69.17	66.60	69.17	66.01	67.74
	SimMTM (Ours)		**80.00**	**79.03**	**80.00**	**78.67**	**79.43**
SleepEEG ↓ EMG	Random Init.		77.80	59.09	66.67	62.38	66.49
	Contrastive	TS2vec (Yue et al., 2022)	78.54	80.40	67.85	67.66	73.61
		Mixing-up (Wickstrøm et al., 2022)	30.24	10.99	25.83	15.41	20.62
		TS-TCC (Eldele et al., 2021)	78.89	58.51	63.10	59.04	64.89
		TF-C (Zhang et al., 2022)	81.71	72.65	81.59	76.83	78.20
	Masking	TST (Zerveas et al., 2021)	46.34	15.45	33.33	21.11	29.06
	SimMTM (Ours)		**97.56**	**98.33**	**98.04**	**98.14**	**98.02**

4.2. Forecasting

In-domain. As shown in Table 2, SimMTM outperforms all baselines consistently, regardless of masking-based or contrastive-based methods. On the average of all benchmarks, SimMTM achieves 71.2% MSE reduction and 54.8% MAE reduction compared to the advanced contrastive baseline TS2VeC, 17.4% MSE reduction and 15.1% MAE reduction compared to the masked modeling baseline TST. Besides, empowered by SimMTM pre-training, the model performance is promoted significantly (SimMTM vs. Random Init) and surpasses NSTransformer, which is the state-of-the-art deep model in time series forecasting.

It is also notable that TST (2021) outperforms all the contrastive-based baselines, where TST directly adopts the vanilla masking protocol presented by He et al. (2022) into time series. This indicates that masked modeling based on point-wise reconstruction will suit the forecasting task better than the series-wise contrastive pre-training.

Cross-domain. As shown in Table 3, we present multiple scenarios to verify the fine-tuning performance under the cross-domain setting, where SimMTM consistently outperforms other baselines across all scenarios. Especially on the cross-domain scenarios ETTh2 → ETTh1 and ETTh1 → ETTm1, SimMTM even achieves a comparable performance w.r.t. the corresponding in-domain pre-training settings. This indicates that SimMTM can learn transferable knowledge to improve the performance of target tasks in cross-domain scenarios and outperform other existing time series pre-training methods.

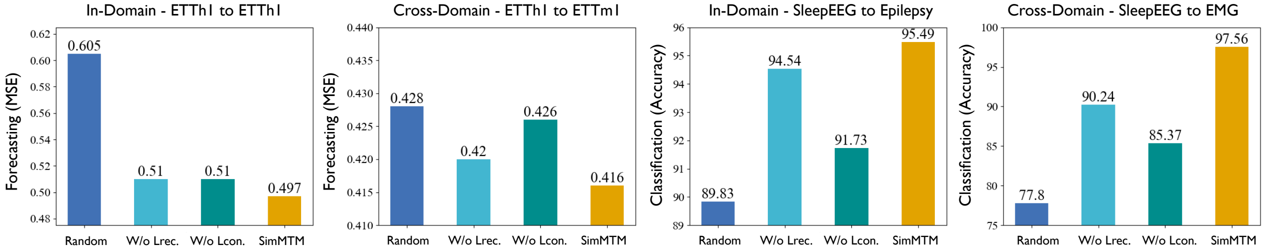


Figure 4. Ablations of SimMTM on the reconstruction loss ($\mathcal{L}_{\text{rec.}}$) and constraint loss ($\mathcal{L}_{\text{con.}}$) in time series forecasting (left part) and classification (right part) tasks under both in- and cross-domain settings. See Tables 14, 15 and 16 for full results.

Table 6. Representation analysis for different pre-training methods in classification and forecasting tasks. For each model, we calculate the centered kernel alignment (CKA) similarity (Kornblith et al., 2019) between representations from the first and the last layers. A higher CKA similarity means more similar representations. For comparison, we also calculate the $|\Delta_{\text{CKA}}|$ between pre-trained and fine-tuned models, where a smaller value indicates a smaller representation gap between pre-training and fine-tuning.

Models		Contrastive				Masking	
		TF-C (2022)	TS-TCC (2021)	Mixing-up (2022)	TS2Vec (2022)	TST (2021)	**SimMTM (Ours)**
Classification	CKA of Pre-trained Model	84.78%	41.78%	88.94%	70.01%	54.98%	**33.87%**
	CKA of Fine-tune Model	86.30%	43.14%	90.06%	69.79%	55.80%	**32.84%**
	$\lvert\Delta_{\text{CKA}}\rvert$	1.53%	1.35%	1.12%	**0.22%**	0.82%	1.04%
Forecasting	CKA of Pre-trained Model	59.35%	43.75%	58.62%	70.20%	99.76%	**97.79%**
	CKA of Fine-tuned Model	60.60%	60.42%	60.98%	83.73%	94.92%	**97.89%**
	$\lvert\Delta_{\text{CKA}}\rvert$	1.25%	16.67%	2.36%	13.53%	4.84%	**0.11%**
	Sum $\lvert\Delta_{\text{CKA}}\rvert$	2.77%	18.02%	3.48%	13.75%	5.66%	**1.15%**

4.3. Classification

In-domain. We investigate the in-domain pre-training effect on the time series classification tasks in Table 4, where we pre-train a model on SleepEEG, followed by the fine-tuning on Epilepsy. Note that different from forecasting, the classification task requires the model to learn the high-level representation of time series. As shown in Table 4, we can find that the contrastive pre-training baselines TS2Vec and TFC achieve competitive performances. In contrast, the vanilla masking-based model TST exhibits a negative transfer phenomenon in comparison to random initialization, indicating that contrastive learning is generally more suitable for classification tasks.

It is surprising that while SimMTM follows the masked modeling paradigm, with our specially-designed reconstruction task, it can still achieve the best performance in the classification task. This is benefited from the neighborhood aggregation from *multiple* masked series, which enables the model to exploit the local structure of time series manifold.

Cross-domain. As presented in Table 5, we experiment with three cross-domain fine-tuning scenarios, namely from SleepEEG to FD-B, Gesture and EMG, where the target datasets are distinct from the pre-training dataset.

Due to the large gap between pre-training and fine-tuning datasets, the baselines perform poorly in most cases of the cross-domain setting, while SimMTM still surpasses other baselines and the random initialization significantly. These results demonstrate that SimMTM can precisely capture valuable knowledge from pre-training datasets and uniformly benefit extensive downstream datasets. Especially for the SleepEEG → EMG, SimMTM remarkably surpasses previous state-of-the-art TF-C (Avg.: 78.2% vs. 98.02%).

4.4. Model Analysis

Ablations. As shown in Figure 4, we provide ablations to the two parts of the training loss in SimMTM. It is observed that both $\mathcal{L}_{\text{reconstruction}}$ and $\mathcal{L}_{\text{constraint}}$ are essential to the final performance. Especially, for the SleepEEG → EMG experiment, SimMTM surpasses the random initialization remarkably, where reconstruction and constraint losses provide 7.32% and 12.19% absolute improvement respectively. Besides, we can also find that in comparison to $\mathcal{L}_{\text{reconstruction}}$, $\mathcal{L}_{\text{constraint}}$ provides more contributions to the final results. This comes from our design that the constraint loss uncovers a proper time series manifold helpful for reconstruction from multiple masked series, without which the neighborhood aggregation will degenerate to the trivial average.

Representation analysis. To illustrate the advantages of SimMTM intuitively, we provide a representation analysis in Table 6, where we can find the following observations. Firstly, we can find that the CKA value of SimMTM in the

Table 7. Performance by applying SimMTM to three advanced time series forecasting models under the in-domain setting. We report the MSE and MAE averaged from all prediction lengths.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2	
Model	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Trans.	1.088	0.836	4.103	1.612	0.901	0.704	1.624	0.901
+ Ours	**0.927**	**0.761**	**3.498**	**1.487**	**0.809**	**0.663**	**1.322**	**0.808**
Gain	12.3 %		12.8 %		8.3 %		15.6 %	
Auto.	0.573	0.573	0.550	0.559	0.615	0.528	0.324	0.368
+ Ours	**0.561**	**0.568**	**0.543**	**0.555**	**0.553**	**0.505**	**0.315**	**0.360**
Gain	1.5 %		1.0 %		7.4 %		2.5 %	
NSTrans.	0.570	0.537	0.526	0.516	0.481	0.456	0.306	0.347
+ Ours	**0.543**	**0.527**	**0.493**	**0.514**	**0.431**	**0.455**	**0.301**	**0.345**
Gain	3.4 %		3.3 %		6.5 %		1.0 %	

classification task is clearly smaller than the values in the forecasting task, where the former is a high-level task and the latter requires low-level representations. These results demonstrate that SimMTM can learn adaptive representations for different tasks, which can be benefited from our design in the pre-training loss. Concretely, the temporal variations of classification pre-training datasets are much more diverse than the forecasting datasets. Thus, the $\mathcal{L}_{\text{constraint}}$ will be easier for optimization in classification, deriving a smaller CKA value. Secondly, from $|\Delta_{\text{CKA}}|$, it is observed that the models pre-trained from SimMTM present a smaller representation gap w.r.t. the fine-tuned models, which is why SimMTM can consistently improve downstream tasks.

Model generality. From Table 7, we can find that as a general time series pre-training framework, SimMTM can consistently improve the forecasting performance of diverse base models, even for the state-of-the-art time series forecasting model NSTransformer (Liu et al., 2022). This generality also indicates that by employing advanced base models as encoders, we can further improve the model performance.

Fine-tuning to limited data scenarios. One essential application of pre-training models is to provide prior knowledge for downstream tasks, especially for limited data scenarios, which is important to the fast-adaption of deep models. Thus, to verify the effectiveness of SimMTM and other pre-training methods in data-limited scenarios, we pre-train a model on ETTh2 and fine-tune it to ETTh1 with different choices for the remaining proportions of training data. All results are presented in Figure 5. we can find that SimMTM achieves significant performance gains in different data proportions compared to other time series pre-training methods.

Masking strategy. Note that the difficulty of reconstructing the original time series increases along with the increase of the masked ratio, but decreases when the number of neighbor masked series increases. We explore the potential relationship between the masked ratio and the number of masked series used for reconstruction, namely r and M in Eq. (2) respectively. The experimental results in Figure 6 show that we need to set $M \propto r$ to obtain better results, namely larger masking ratio requires more masked series for reconstruction. Therefore, a reasonable balance between the masked ratio and the number of reconstructed series is critical. Experimentally, we choose the masking ratio as 50% and three masked series throughout this paper.

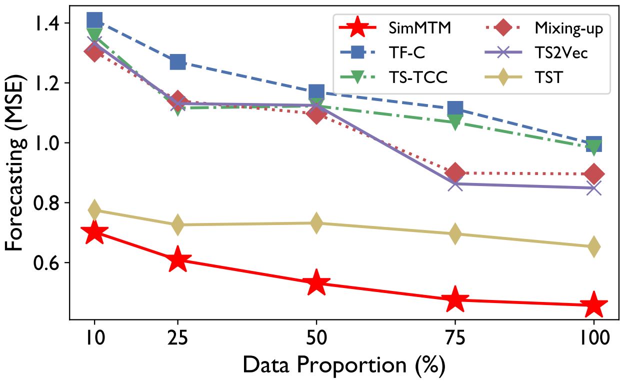
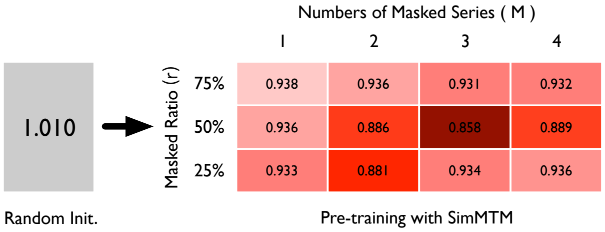


Figure 5. Fine-tuning ETTh2 pre-trained model to ETTh1 with limited data. A lower MSE means better forecasting performance.



	1	2	3	4
75%	0.938	0.936	0.931	0.932
50%	0.936	0.886	0.858	0.889
25%	0.933	0.881	0.934	0.936

Figure 6. Relationship between the masked ratio r and numbers of masked series M. All the results are the averaged MSE and MAE values for the in-domain setting of ETTh1 under the "input-96-predict-96" setting. A darker red means better performance.

Linear probing. As shown in Table 8, both fine-tuning and linear probing of SimMTM can outperform the fully supervised learning from scratch.

Table 8. Linear probing of SimMTM on in-domain forecasting. We report MSE and MAE averaged from all prediction lengths.

Dataset	ETTm1		ETTm2	
Methods	MSE	MAE	MSE	MAE
Supervised	0.478	0.464	0.416	0.388
Linear Probing	0.432	0.426	**0.296**	**0.337**
Fine-tuning	**0.414**	**0.422**	0.302	0.342

5. Conclusion

This paper presents SimMTM, a simple pre-training framework for masked time-series modeling. Going beyond the previous convention in reconstructing the original time series from unmasked time points, SimMTM proposes a new masked modeling task as reconstructing the original series from its multiple neighbor masked series. Concretely, SimMTM aggregates the point-wise representations based on the series-wise similarities, which are carefully constrained by the neighborhood assumption on the time series manifold. Experimentally, SimMTM can furthest bridge the gap between pre-trained and fine-tuned models, thereby achieving consistent state-of-the-art in distinct forecasting and classification tasks, covering both in- and cross-domain settings. In the future, we will further extend SimMTM to large-scale and diverse pre-training datasets in pursuing the foundation model for time series analysis.

References

Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., and Elger, C. E. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 2001.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. *NeurIPS*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL*, 2018.

Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., and Guan, C. Time-series representation learning via temporal and contextual contrasting. *IJCAI*, 2021.

Franceschi, J.-Y., Dieuleveut, A., and Jaggi, M. Unsupervised scalable representation learning for multivariate time series. *NeurIPS*, 2019.

Gao, T., Fisch, A., and Chen, D. Making pre-trained language models better few-shot learners. *IJCNLP*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CVPR*, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *CVPR*, 2022.

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. A survey on contrastive self-supervised learning. *Technologies*, 2020.

Jiang, J., Shu, Y., Wang, J., and Long, M. Transferability in deep learning: A survey. *arXiv preprint arXiv:2201.05867*, 2022.

Kemp, B., Zwinderman, A. H., Tuk, B., Kamphuisen, H. A., and Oberye, J. J. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 2000.

Kendall, A., Gal, Y., and Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. E. Similarity of neural network representations revisited. In *ICML*, 2019.

Lessmeier, C., Kimotho, J. K., Zimmer, D., and Sextro, W. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM Society European Conference*, 2016.

Li, Y., Fan, H., Hu, R., Feichtenhofer, C., and He, K. Scaling language-image pre-training via masking. *arXiv preprint arXiv:2212.00794*, 2022.

Liu, J., Zhong, L., Wickramasuriya, J., and Vasudevan, V. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 2009.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., and Tang, J. Self-supervised learning: Generative or contrastive. *TKDE*, 2021.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *NeurIPS*, 2022.

Mueen, A. A. and Keogh, E. J. Extracting optimal performance from dynamic time warping. *KDD*, 2016.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

PhysioBank, P. Physionet: components of a new research resource for complex physiologic signals. *Circulation*, 2000.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 2020.

Rebjock, Q., Kurt, B., Januschowski, T., and Callot, L. Online false discovery rate control for anomaly detection in time series. *NeurIPS*, 2021.

Sarkar, P. and Etemad, A. Self-supervised learning for ecg-based emotion recognition. ICASSP, 2020.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. *CVPR*, 2015.

Sun, F.-K., Lang, C., and Boning, D. Adjusting for autocorrelated errors in neural networks for time series. *NeurIPS*, 2021.

Tang, C. I., Perez-Pozuelo, I., Spathis, D., and Mascolo, C. Exploring contrastive learning in human activity recognition for healthcare. *arXiv preprint arXiv:2011.11542*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *NeurIPS*, 2017.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 2010.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

Wickstrøm, K., Kampffmeyer, M., Mikalsen, K. Ø., and Jenssen, R. Mixing up contrastive learning: Self-supervised representation learning for time series. *PRL*, 2022.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *NeurIPS*, 2021.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. *CVPR*, 2018.

Xie, Z., Geng, Z., Hu, J., Zhang, Z., Hu, H., and Cao, Y. Revealing the dark secrets of masked image modeling. *arXiv preprint arXiv:2205.13543*, 2022a.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022b.

Xu, J., Wu, H., Wang, J., and Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. In *ICLR*, 2021.

Yang, L. and Hong, S. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *ICML*, 2022.

Yang, X., Zhang, Z., and Cui, R. Timeclr: A self-supervised contrastive learning framework for univariate time series representation. *KBS*, 2022.

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. TS2Vec: Towards Universal Representation of Time Series. *AAAI*, 2022.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., and Eickhoff, C. A transformer-based framework for multivariate time series representation learning. In *SIGKDD*, 2021.

Zhang, X., Zhao, Z., Tsiligkaridis, T., and Zitnik, M. Self-supervised contrastive pre-training for time series via time-frequency consistency. *NeurIPS*, 2022.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.

A. Implementation Details

All the experiments are repeated five times and implemented in PyTorch (Paszke et al., 2019) and conducted on a single NVIDIA TITAN RTX 24GB GPU. We implement the baselines based on their official codes and follow the configuration from their original papers. For the metrics, we adopt the mean square error (MSE) and mean absolute error (MAE) for the time series forecasting. As for the classification, accuracy, precision, recall, F1 score, and their average value are recorded.

A.1. Dataset Description

We conduct experiments to evaluate the effect of our method under in- and cross-domain settings on nine real-world datasets for two typical time series analysis tasks: classification and forecasting, covering diverse application scenarios (electricity system, neurological healthcare, human activity recognition, mechanical fault detection, and physical status monitoring), different types of signals (ECG, EMG, acceleration, vibration, and power load), multivariate channel dimensions (from 1 to 7), varying times series lengths (from 96 to 5120) and large span sampling ratio (from 100 Hz to 4000 Hz). The detailed descriptions of these datasets are summarized in Table 9.

Table 9. Datasets in Forecasting (*Fore.*) and Classification (*Class.*) tasks. *Samples* are in the formalization of Train/Valid/Test.

Tasks	Datasets	Channels	Length	Samples	Classes	Information	Frequency
Fore.	ETTh1,ETTh2	7	{96, 192, 336, 720}	34465/11521/11521	-	Electricity	Hourly
	ETTm1,ETTm2	7	{96, 192, 336, 720}	8545/2881/2881	-	Electricity	15 Mins
Class.	SleepEEG	1	200	371005	5	EEG	100 Hz
	Epilepsy	1	178	60/20/11420	2	EEG	174 Hz
	FD-B	1	5120	60/21/135599	3	Faulty Detection	64K Hz
	Gesture	3	315	320/120/120	8	Hand Movement	100 Hz
	EMG	1	1500	122/41/41	3	Muscle responses	4000 Hz

(1) **ETT** (Zhou et al., 2021) contains the time series of oil temperature and power load collected by electricity transformers from July 2016 to July 2018. ETT is a group of four subsets with different recorded frequencies: ETTm1 / ETTm2 are recorded every 15 minutes, and ETTh1 / ETTh2 are recorded every hour.

(2) **SLEEPEEG** (Kemp et al., 2000) contains 153 whole-night sleeping electroencephalography (EEG) recordings from 82 healthy subjects. We follow the same data preprocessing approach as (Zhang et al., 2022) to segment the EEG signals without overlapping and get 371,055 univariate brainwaves. Each brainwave is sampled at a frequency of 100 Hz and associated with one of five sleeping stages: Wake, Non-rapid eye movement (3 sub-states), and Rapid Eye Movement.

(3) **EPILEPSY** (Andrzejak et al., 2001) monitors the brain activities of 500 subjects with a single-channel EEG sensor. Every subject is recorded for 23.6 seconds of brain activities. The dataset is sampled at 178 Hz and contains 11,500 samples in total. We follow the procedure described by (Zhang et al., 2022). The first four classes (eyes open, eyes closed, EEG measured in the healthy brain region, and EEG measured in the tumor region) of the original five categories of each sample are classified as positive, and the remaining classes (whether the subject has a seizure episode) are used as negative.

(4) **FD-B** (Lessmeier et al., 2016) is generated by electromechanical drive systems. It monitors the condition of rolling bearings and detects their failures based on the monitoring conditions, which include speed, load torque, and radial force. Concretely, FD-B has 13,640 samples in total. Each recording is sampled at 64k Hz with 3-class labels: undamaged, inner damaged, and outer damaged.

(5) **GESTURE** (Liu et al., 2009) are collected from 8 hand gestures based on the paths of hand movement recorded by an accelerometer. The eight gestures are: hand swiping left, right, up, and down, hand waving in a counterclockwise or clockwise circle, hand waving in a square, and waving a right arrow respectively. This dataset contains 440 examples of balanced classification labels that can be used, and each sample contains eight different kinds of gesture categories.

(6) **EMG** (PhysioBank, 2000) is sampled with 4K Hz and consists of 163 single-channel EMG recordings from the tibialis anterior muscle of three healthy volunteers suffering from neuropathy and myopathy. Each patient is a classification category, so each sample is associated with one of three classes.

A.2. Pre-training and Fine-tuning Configuration

We built two types of pre-training and fine-tuning scenarios, in- and cross-domain, based on the benchmarks of forecasting and classification tasks to compare the effectiveness of our method and other time series pre-training methods.

For forecasting tasks, we pre-train a model on one of the ETT subsets and fine-tune it to the same dataset to build four in-domain transfer evaluation scenarios. In cross-domain evaluation, one certain ETT dataset is selected to pre-train a model, and then we use the other ETT datasets for fine-tuning. Based on the above settings, we constructed eight in- and cross-domain pre-training and fine-tuning experiments, covering the same dataset with the same sampled frequency, different datasets with the same sampled frequency, and different datasets with different sampled frequencies.

We pre-train a model for classification tasks on a univariate time series dataset SleepEEG, which has the most complex temporal dynamics and the most samples. And then fine-tune the model separately on Epilepsy, FD-B, Gesture, and EMG. We use SleepEEG and Epilepsy, which are both single-channel EEG sensor signals, to construct the in-domain setting for classification tasks. Furthermore, we constructed three cross-domain evaluation scenarios by pre-training from SleepEEG and fine-tuning to FD-B, Gesture, and EMG because of fewer commonalities and the enormous gap among these datasets. Detailed pre-training and fine-tuning settings are shown in Table 10.

Table 10. Pre-training and fine-tuning scenarios in Forecasting (*Fore.*) and Classification (*Class.*) tasks.

Tasks	Evaluation	Scenarios	Transfer
Fore.	In-domain	ETTh1 → ETTh1 ETTh2 → ETTh2 ETTm1 → ETTm1 ETTm2 → ETTm2	The same dataset with the same frequency
	Cross-domain	ETTh2 → ETTh1 ETTm2 → ETTm1	Different datasets with the same frequency
		{ETTm1,ETTm2} → ETTh1 {ETTh1,ETTh2} → ETTm1	Different datasets with different frequencies
Class.	In-domain	SleepEEG → Epilepsy	Different datasets in the same domain
	Cross-domain	SleepEEG → {FD-B, Gesture, EMG}	Different datasets in different domains

A.3. Model and Training Configuration

Following the previous convention, for forecasting tasks, we choose the encoder part of Transformer (Vaswani et al., 2017) as the feature extractor. For the classification tasks, we adopt 1D-ResNet (He et al., 2016) as the encoder following (Zhang et al., 2022). In the pre-training stages, we pre-train the model with different learning rates and batch sizes according to the pre-train datasets. Then we fine-tune it to downstream forecasting and classification tasks, which are supervised by L2 and Cross-Entropy losses respectively. The configuration details are shown in Table 11.

Table 11. Model and training configuration in Forecasting (*Fore.*) and Classification (*Class.*) tasks.

Tasks	Encoder		Pre-training			Fine-tuning			
	Layers	d_{model}	Learning rate	Batch size	Epochs	Learning rate	Loss function	Batch size	Epochs
Fore.	4	16	1e-5	64	50	1e-4	L2	32	10
Class.	4	128	1e-5	128	100	3e-4	Cross-Entropy	32	100

B. Comparison of Masked Modeling

To investigate the reconstruction process of different masked modeling methods, we plot both original and reconstructed time series from TST and SimMTM in Figure 7, where TST (Zerveas et al., 2021) reconstructs the masked time series based on the unmasked time points directly. From Figure 7, we can find that direct reconstruction is too difficult in time series, even for the 12.5% masking ratio. As for the 75% masking ratio, TST degenerates more seriously. In view of this poor reconstruction effect, direct reconstruction is hard to provide reliable guidance to model pre-training. In contrast, our proposed SimMTM can precisely reconstruct the original time series, thereby benefiting the representation learning. These results also support our design in neighborhood reconstruction.

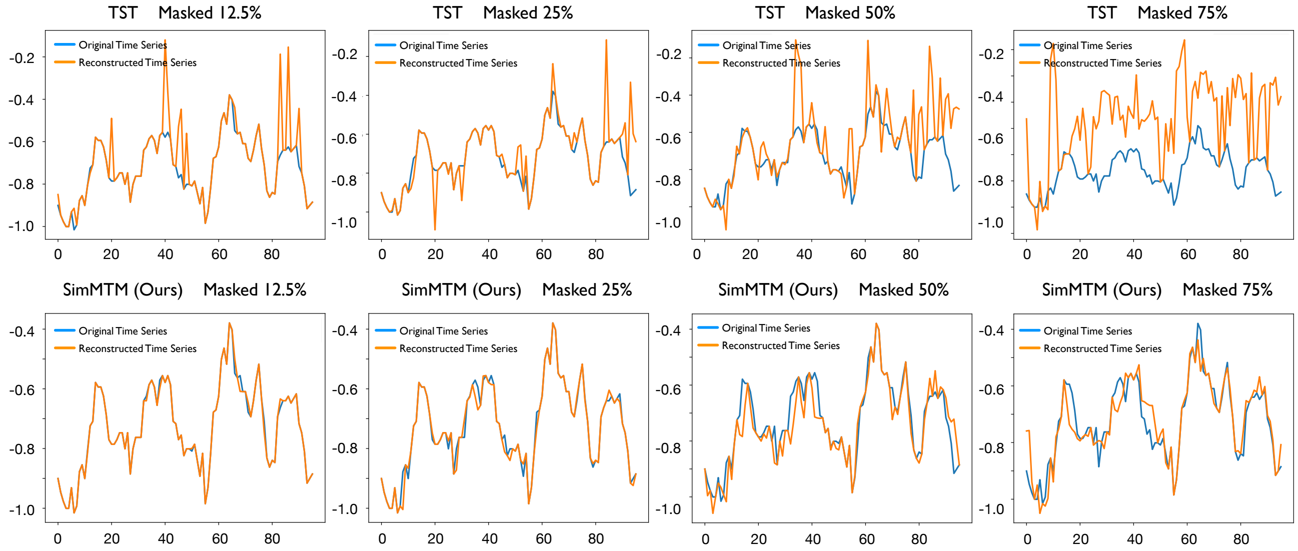


Figure 7. Comparison of different masked modeling pre-training methods in reconstructing time series. All the cases are from ETTh1.

C. Full Results

Due to the space limitation of the main text, we present the full results of all experiments in the main text as follows:

- Results for the in-domain setting of forecasting: Table 12.
- Results for the cross-domain setting of forecasting: Table 13.
- Ablations for the in-domain setting of forecasting: Table 14.
- Ablations for the cross-domain setting of forecasting: Table 15.
- Ablations for the in- and cross-domain setting of classification: Table 16.
- Results for fine-tuning to limited data scenarios: Table 17.

Table 12. Full results for the in-domain setting of forecasting. Pre-training and fine-tuning are performed on the same ETT datasets. The standard deviations of SimMTM are within 0.005 for MSE and within 0.004 for MAE.

Models		**SimMTM**		NsTrans.		Random Init.		TST		TF-C		TS-TCC		Mixing-up		TS2Vec	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.445**	**0.445**	0.513	0.491	0.520	0.490	0.503	0.527	1.065	0.804	0.953	0.740	1.055	0.802	0.709	0.650
	192	**0.488**	**0.467**	0.534	0.504	0.596	0.544	0.601	0.552	1.130	0.840	1.108	0.827	1.003	0.797	0.927	0.757
	336	**0.514**	**0.478**	0.588	0.535	0.650	0.575	0.625	0.541	1.305	0.945	1.243	0.915	1.197	0.890	0.986	0.811
	720	**0.540**	**0.513**	0.643	0.616	0.653	0.588	0.768	0.628	1.147	0.862	1.306	0.945	1.138	0.849	0.967	0.790
	Avg	**0.497**	**0.476**	0.570	0.537	0.605	0.549	0.624	0.562	1.162	0.863	1.152	0.857	1.098	1.933	0.897	0.752
ETTh2	96	**0.328**	**0.371**	0.476	0.458	0.358	0.393	0.335	0.392	1.663	1.021	2.788	1.368	1.761	1.072	1.560	1.077
	192	**0.418**	**0.425**	0.512	0.493	0.491	0.468	0.444	0.441	3.525	1.561	3.178	1.519	2.465	1.223	3.507	1.647
	336	0.456	**0.455**	0.552	0.551	0.492	0.476	**0.455**	0.494	3.283	1.500	3.350	1.620	3.876	1.680	2.794	1.428
	720	**0.456**	**0.461**	0.562	0.560	0.486	0.482	0.481	0.504	2.930	1.316	3.089	1.527	2.790	1.415	2.650	1.373
	Avg	**0.415**	**0.428**	0.526	0.516	0.457	0.455	0.429	0.458	2.850	1.349	3.101	1.509	2.723	1.348	2.628	1.381
ETTm1	96	**0.348**	**0.384**	0.386	0.398	0.414	0.418	0.454	0.456	0.671	0.601	0.848	0.741	0.609	0.553	0.563	0.551
	192	**0.386**	**0.406**	0.459	0.444	0.467	0.469	0.471	0.490	0.719	0.638	0.704	0.675	0.674	0.608	0.599	0.558
	336	**0.434**	**0.435**	0.495	0.464	0.499	0.470	0.457	0.451	0.743	0.659	0.955	0.792	0.754	0.649	0.685	0.594
	720	**0.486**	**0.463**	0.585	0.516	0.533	0.500	0.594	0.488	0.842	0.708	2.683	1.363	0.898	0.729	0.831	0.698
	Avg	**0.414**	**0.422**	0.481	0.456	0.478	0.464	0.494	0.471	0.744	0.652	1.298	0.893	0.734	0.635	0.669	0.600
ETTm2	96	0.201	0.284	**0.192**	**0.274**	0.229	0.303	0.363	0.301	0.401	0.490	0.956	0.741	0.927	0.717	1.548	1.012
	192	**0.261**	**0.317**	0.280	0.339	0.396	0.392	0.342	0.364	0.822	0.677	1.110	0.828	1.358	0.882	1.145	0.836
	336	**0.323**	**0.355**	0.334	0.361	0.516	0.446	0.414	0.361	1.214	0.908	1.243	0.915	1.139	0.829	0.981	0.744
	720	0.424	**0.412**	**0.417**	0.413	0.521	0.412	0.580	0.456	4.584	1.711	1.302	0.944	2.257	1.220	2.191	1.237
	Avg	**0.302**	**0.342**	0.306	0.347	0.416	0.388	0.425	0.371	1.755	0.947	1.153	0.857	1.420	0.912	1.466	0.957

Table 13. Full results for the cross-domain setting of forecasting. The standard deviations of SimMTM are within 0.005 for MSE and within 0.004 for MAE.

Input-96		SimMTM		TST		TF-C		TS-TCC		Mixing-up		TS2Vec	
Predict-O		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2 ↓ ETTh1	96	0.457	0.455	0.653	0.468	0.996	0.769	0.983	0.763	0.896	0.761	0.849	0.694
	192	0.498	0.476	0.658	0.502	1.114	0.821	1.142	0.832	1.061	0.839	0.909	0.738
	336	0.516	0.480	0.631	0.561	1.194	0.864	1.259	0.926	1.370	0.940	1.082	0.775
	720	0.525	0.505	0.638	0.608	1.235	0.883	1.221	0.902	1.137	0.874	0.934	0.769
	Avg	**0.499**	**0.479**	0.645	0.535	1.135	0.834	1.151	0.856	1.116	0.854	0.944	0.744
ETTm1 ↓ ETTh1	96	0.471	0.460	0.627	0..477	1.166	0.847	1.024	0.793	1.000	0.789	0.991	0.765
	192	0.492	0.471	0.628	0.500	1.172	0.853	1.164	0.854	1.055	0.799	0.829	0.699
	336	0.527	0.489	0.683	0.554	1.226	0.911	1.291	0.939	1.217	0.899	0.971	0.787
	720	0.537	0.513	0.642	0.600	1.235	0.897	1.169	0.869	1.106	0.813	1.037	0.820
	Avg	**0.507**	**0.483**	0.645	0.533	1.200	0.877	1.162	0.864	1.094	0.825	0.957	0.768
ETTm2 ↓ ETTh1	96	0.474	0.463	0.559	0.489	0.968	0.738	0.959	0.745	1.070	0.795	0.783	0.669
	192	0.501	0.476	0.600	0.579	1.080	0.801	1.078	0.810	1.180	0.862	0.828	0.691
	336	0.528	0.490	0.677	0.572	1.091	0.824	1.242	0.913	1.233	0.922	0.990	0.762
	720	0.527	0.508	0.694	0.664	1.226	0.893	1.198	0.888	1.067	0.839	0.985	0.783
	Avg	**0.508**	**0.484**	0.632	0.576	1.091	0.814	1.119	0.839	1.137	0.855	0.896	0.726
ETTh1 ↓ ETTm1	96	0.349	0.384	0.425	0.381	0.672	0.600	0.607	0.554	0.607	0.550	0.605	0.561
	192	0.387	0.404	0.495	0.478	0.721	0.639	0.619	0.575	0.675	0.608	0.615	0.561
	336	0.438	0.433	0.456	0.441	0.755	0.664	0.781	0.688	0.752	0.647	0.763	0.677
	720	0.488	0.463	0.554	0.477	0.837	0.705	0.789	0.682	0.891	0.723	0.805	0.664
	Avg	**0.416**	**0.421**	0.482	0.444	0.746	0.652	0.699	0.625	0.731	0.632	0.697	0.616
ETTh2 ↓ ETTm1	96	0.359	0.392	0..449	0.343	0.677	0.603	0.584	0.545	0.594	0.540	0.466	0.480
	192	0.410	0.416	0.477	0..407	0.718	0.638	0.642	0.601	0.595	0.559	0.557	0.532
	336	0.430	0.430	0.407	0.519	0.755	0.663	0.821	0.715	0.750	0.651	0.646	0.576
	720	0.497	0.465	0.557	0.523	0.848	0.712	0.810	0.702	0.898	0.728	0.752	0.638
	Avg	**0.424**	**0.426**	0.472	0.448	0.750	0.654	0.714	0.641	0.709	0.620	0.606	0.556
ETTm2 ↓ ETTm1	96	0.352	0.384	0.471	0.422	0.610	0.577	0.600	0.553	0.616	0.557	0.586	0.515
	192	0.398	0.409	0.495	0.442	0.725	0.657	0.704	0.642	0.674	0.608	0.624	0.562
	336	0.430	0.430	0.455	0.424	0.768	0.684	0.743	0.668	0.751	0.646	1.035	0.806
	720	0.500	0.465	0.498	0.532	0.927	0.759	0.755	0695	0.896	0.727	0.780	0.669
	Avg	**0.420**	**0.422**	0.480	0.455	0.758	0.669	0.700	0.630	0.734	0.634	0.756	0.638

Table 14. Full ablation studies on ETT datasets for the in-domain setting of forecasting.

INPUT-96		RANDOM INIT.		W/O $\mathcal{L}_{\text{reconstruction}}$		W/O $\mathcal{L}_{\text{constraint}}$		**SIMMTM**	
PREDICT-O		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTH1	96	0.520	0.490	0.453	0.450	0.456	0.453	0.445	0.445
	192	0.596	0.544	0.512	0.484	0.512	0.484	0.488	0.467
	336	0.650	0.575	0.514	0.478	0.510	0.476	0.514	0.478
	720	0.653	0.588	0.559	0.530	0.560	0.528	0.540	0.513
	AVG	0.605	0.549	0.510	0.486	0.510	0.485	**0.497**	**0.476**
ETTH2	96	0.358	0.393	0.339	0.377	0.348	0.384	0.328	0.371
	192	0.491	0.468	0.432	0.431	0.432	0.432	0.418	0.425
	336	0.492	0.476	0.452	0.454	0.454	0.457	0.456	0.455
	720	0.486	0.482	0.478	0.475	0.472	0.471	0.456	0.461
	AVG	0.457	0.455	0.425	0.434	0.427	0.436	**0.415**	**0.428**
ETTM1	96	0.414	0.418	0.351	0.384	0.364	0.392	0.348	0.384
	192	0.467	0.469	0.407	0.414	0.406	0.416	0.386	0.406
	336	0.499	0.470	0.440	0.435	0.442	0.432	0.434	0.435
	720	0.533	0.500	0.493	0.465	0.503	0.472	0.486	0.463
	AVG	0.478	0.464	0.423	0.425	0.428	0.428	**0.414**	**0.422**
ETTM2	96	0.229	0.303	0.199	0.282	0.205	0.289	0.201	0.284
	192	0.396	0.392	0.268	0.325	0.267	0.321	0.261	0.317
	336	0.516	0.446	0.353	0.373	0.353	0.373	0.323	0.355
	720	0.521	0.412	0.439	0.422	0.438	0.421	0.424	0.412
	AVG	0.521	0.412	0.315	0.351	0.316	0.351	**0.302**	**0.342**

Table 15. Full ablation studies on ETT datasets to ETTm1 results for the cross-domain setting of forecasting.

INPUT-96		RANDOM INIT.		W/O $\mathcal{L}_{\text{reconstruction}}$		W/O $\mathcal{L}_{\text{constraint}}$		**SIMMTM**	
PREDICT-O		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTH1 ↓ ETTM1	96	0.356	0.388	0.344	0.379	0.366	0.398	0.349	0.384
	192	0.399	0.406	0.404	0.408	0.398	0.411	0.387	0.404
	336	0.461	0.441	0.431	0.431	0.447	0.439	0.438	0.433
	720	0.497	0.468	0.499	0.471	0.494	0.470	0.488	0.463
	AVG	0.428	0.426	0.420	0.422	0.426	0.430	**0.416**	**0.421**
ETTH2 ↓ ETTM1	96	0.356	0.388	0.354	0.387	0.355	0.387	0.359	0.392
	192	0.399	0.406	0.405	0.415	0.400	0.413	0.410	0.416
	336	0.461	0.441	0.438	0.434	0.444	0.439	0.430	0.430
	720	0.497	0.468	0.494	0.461	0.494	0.468	0.497	0.465
	AVG	0.428	0.426	0.422	0.424	0.423	0.427	**0.424**	**0.426**
ETTM2 ↓ ETTM1	96	0.356	0.388	0.344	0.379	0.350	0.384	0.352	0.384
	192	0.399	0.406	0.403	0.413	0.414	0.412	0.398	0.409
	336	0.461	0.441	0.462	0.449	0.440	0.436	0.430	0.430
	720	0.497	0.468	0.499	0.469	0.487	0.460	0.500	0.465
	AVG	0.428	0.426	0.427	0.428	0.423	0.423	**0.420**	**0.422**
ETT-MERGE ↓ ETTM1	96	0.365	0.388	0.354	0.387	0.362	0.394	0.353	0.383
	192	0.399	0.406	0.402	0.412	0.400	0.413	0.393	0.405
	336	0.461	0.441	0.427	0.428	0.442	0.440	0.437	0.433
	720	0.497	0.468	0.496	0.468	0.493	0.470	0.492	0.460
	AVG	0.428	0.426	0.420	0.424	0.424	0.429	**0.419**	**0.420**

Table 16. Full ablation studies for in- and cross-domain settings of classification. Under the *Avg* metric, the standard deviations of SimMTM are within 0.2% for Epilepsy, within 0.5% for FD-B, within 0.6% for Gesture, and within 0.1% for EMG.

Scenarios		Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Avg (%)
SleepEEG ↓ Epilepsy	Random Init.	89.83	92.13	74.47	79.59	84.00
	W/O $\mathcal{L}_{\text{reconstruction}}$	94.54	93.87	88.46	90.84	91.93
	W/O $\mathcal{L}_{\text{constraint}}$	91.73	90.57	82.21	85.53	87.51
	SimMTM	**95.49**	**93.36**	**92.28**	**92.81**	**93.49**
SleepEEG ↓ FD-B	Random Init.	47.36	48.29	52.35	49.11	49.28
	W/O $\mathcal{L}_{\text{reconstruction}}$	66.11	67.97	74.70	70.01	69.70
	W/O $\mathcal{L}_{\text{constraint}}$	53.71	69.48	62.67	50.86	59.18
	SimMTM	**69.40**	**74.18**	**76.41**	**75.11**	**73.78**
SleepEEG ↓ Gesture	Random Init.	42.19	47.51	49.63	48.86	47.05
	W/O $\mathcal{L}_{\text{reconstruction}}$	78.50	79.01	78.50	77.17	78.30
	W/O $\mathcal{L}_{\text{constraint}}$	76.67	74.91	76.67	74.80	75.76
	SimMTM	**80.00**	**79.03**	**80.00**	**78.67**	**79.43**
SleepEEG ↓ EMG	Random Init.	77.80	59.09	66.67	62.38	66.49
	W/O $\mathcal{L}_{\text{reconstruction}}$	90.24	94.20	78.04	81.53	86.00
	W/O $\mathcal{L}_{\text{constraint}}$	85.37	89.97	69.62	70.74	78.93
	SimMTM	**97.56**	**98.33**	**98.04**	**98.14**	**98.02**

Table 17. Full results for fine-tuning to limited data scenarios. The input and prediction sequence length is 96. We fine-tune the model pre-trained from ETTh2 to ETTh1 with different data proportions $\{10\%, 25\%, 50\%, 75\%, 100\%\}$.

Models		Contrastive								Masking			
		TF-C (2022)		TS-TCC (2021)		Mixing-up (2022)		TS2Vec (2022)		TST (2021)		**SimMTM**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2 ↓ ETTh1	10%	1.410	0.851	1.356	0.882	1.305	0.870	1.331	0.869	0.775	0.602	**0.702**	**0.547**
	25%	1.270	0.832	1.116	0.792	1.140	0.790	1.131	0.782	0.726	0.552	**0.609**	**0.510**
	50%	1.169	0.787	1.123	0.799	1.097	0.760	1.125	0.746	0.732	0.553	**0.531**	**0.483**
	75%	1.113	0.767	1.068	0.773	0.899	0.758	0.863	0.690	0.696	0.539	**0.475**	**0.464**
	100%	0.996	0.769	0.983	0.763	0.896	0.761	0.849	0.694	0.653	0.468	**0.457**	**0.455**