SimPM: A Simple Patch Masking Contrastive Learning Framework for Time Series Forecasting

Jinjun Zhang[1,2] , Tianyi Wang[1,2] and Xiaolin Qin[1,2(✉)]

[1] Chengdu Institute of Computer Applications, Chinese Academy of Sciences
[2] School of Computer Science and Technology, University of Chinese Academy of Sciences
qinxl2001@126.com

Abstract. Time series forecasting plays a critical role in numerous practical industries, where effectively learning and extracting meaningful representations has always been a significant and challenging problem. Although contrastive learning methods have shown outstanding ability in learning meaningful representations in computer vision and natural language processing domains, their performance in time series forecasting tasks is weaker. This weakness can mainly be attributed to their failure to fully consider the characteristics of time series data, leading to information loss. Specifically, existing data augmentation strategies primarily operate at the timestamp level, which cannot fully exploit and utilize local semantic information. Moreover, previous research has not taken into account the sharing of information between independent channels when dealing with inter-channel information. This limitation, to some extent, restricts the integrity of the learned representations. To address these issues, we propose a new method called SimPM, a simple patch masking contrastive learning framework for time series forecasting that effectively mitigates information loss. In our experiments on seven benchmark time series forecasting datasets, SimPM demonstrates competitive performance compared to existing contrastive learning methods.

Keywords: time series forecasting, contrastive learning, patch masking.

1 Introduction

In recent years, significant progress has been made in time series forecasting, with extensive applications in weather forecasting, finance, and traffic prediction [1-4].It is crucial to fully utilize the large volume of time series data to obtain meaningful representations. Contrastive learning, an effective unsupervised representation learning method, has achieved remarkable success in CV and NLP domains [5-8].

By learning representations by contrasting similar and dissimilar data samples, contrastive learning can capture the latent structure and relationships in the data [9]. However, the data augmentation strategies commonly utilized in contrastive learning are designed based on inherent characteristics of images and text, such as positional and semantic equivariance. As such, these strategies cannot be directly adapted to time

Jinjun Zhang and Tianyi Wang—These authors contributed equally to this work.

series forecasting tasks without appropriate modifications. Recently, researchers have proposed a series of augmentation methods for time series [10-14], but most focus on the timestamp-level and have difficulty extracting local semantic information. Moreover, due to widespread periodicity in time series data, false negative samples often occur, not addressed in previous methods.

In the realm of time series forecasting, the role of inter-channel correlation is of significant importance. Despite this, Dlinear [15] has managed to achieve notable success by introducing the concept of channel independence (CI), which has led to enhanced efficiency. However, this approach may inadvertently result in information loss and create bottlenecks. An excessive focus on channel correlation could also give rise to issues related to computational complexity. Consequently, it is of paramount importance to strike a balance by fully leveraging both the aspects of independence and correlation.

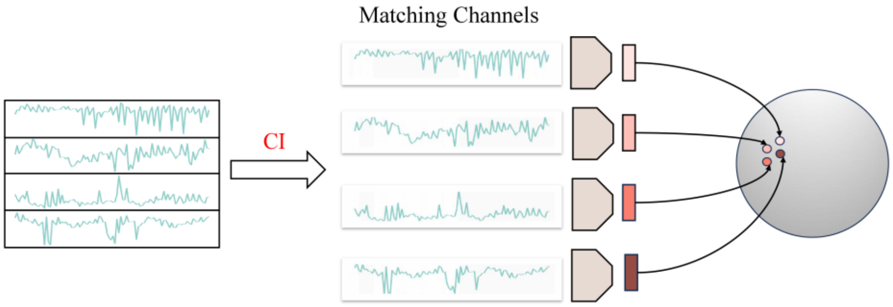

Fig. 1. Given a set of time series channels, we combine the channels of the same time series in a high-dimensional embedding space to learn deep representations. We present different channels of the same time series from the ETTh1 dataset.

In order to address these challenges, we propose SimPM, a **Sim**ple **P**atch **M**asking contrastive learning framework for time series forecasting. The main contributions of this paper are summarized as follows:

- We propose a multi-channel contrastive learning framework to capture shared representations across channels while maintaining compactness by discarding channel interference factors. **错误!未找到引用源。** shows our strategy regarding the channel. To avoid false negative samples, we incorporate a self-supervised framework without negative samples.
- We propose a new data augmentation strategy that aggregates time steps into univariate subsequence patches and applies random masking to enhance locality and capture comprehensive semantic not obtainable at the timestamp level.
- We evaluate on 7 datasets, in which the mean squared error (MSE) is 17.4% lower than the baseline model. We conduct extensive ablation studies to demonstrate

the generalizability of each proposed module and the robustness of SimPM to different encoder architectures.

2 RELATED WORK

2.1 Contrastive Learning for Time Series Forecasting

In recent years, end-to-end models have achieved better performance in time series prediction tasks compared to traditional models. Additionally, two-stage methods have begun to show potential due to their ability to learn better representations for downstream tasks. Among these, contrastive learning optimizes self-discrimination tasks to learn meaningful time series representations by contrasting augmented positive samples against negative samples. In recent years, researchers have explored data augmentation strategies to obtain reliable positive and negative samples. Specifically, TNC [11] uses time-invariance by defining local windows as positive samples and distal signals as negative samples, mitigating sampling bias through Positive-Unlabeled (PU) learning. TS-TCC [10] and CA-TCC [16] perform cross-view prediction with strong and weak augmentations to improve the forecasting results. CoST [12] transforms different augmented views into representations of amplitude and phase through Fast Fourier Transform, enhancing the interpretability of the representations. LaST [17] uses variational inference to disentangle and learn the seasonal-trend representations in the latent space of time series data. However, these timestamp-level methods have difficulty capturing local semantic information. Moreover, negative sample selection is not well-considered, which may result in false exclusions [18].

2.2 Channel independence and Channel dependence

Channel-dependent (CD) methods make future data predictions by considering the historical data across all channels. Conversely, channel-independent (CI) methods treat multivariate time series as individual univariate time series and use univariate prediction functions to create multivariate predictors. With this approach, the prediction for a specific channel depends only on its own historical values, ignoring other channels' data. Dlinear [15] surpasses well-designed transformer-based models by training a simple linear model using a channel-independent training strategy. Further studies, such as PatchTST [19], have also proven that channel independence can boost performance, while mixed-channel models are more likely to overfit. CrossFormer [20] attempts to enhance the mixed-channel aspect of transformers but still faces challenges with high channel noise interaction and the inability to separate at the output layer.

2.3 Patching Strategies

Patching has been applied to tasks in various data modalities, reducing noise interference in models and helping capture local semantic information. In NLP, large pre-trained models like BERT [21] and GPT-3 [22] adopt subword-level segmentation methods, improving semantic understanding capabilities. In CV, transformer-based models like ViT [23], Swin Transformer [24], and CvT [25] employ image segmentation strategies. In the contrastive learning of time series, TS2Vec [13] divides multiple time series into several patches and defines hierarchical losses at the instance and patch levels. However, it focuses more on classification tasks and pays less attention to predictive features.

3 Method

3.1 Problem Definition

A time series $X = \{x_1, x_2, \dots, x_T\}$ is a sequence of time step observations over a time range T, where each $x_t \in \mathbb{R}^M$. We represent the i^{th} univariate subsequence $x_{1:T}^{(i)} = (x_1^{(i)}, \dots, x_T^{(i)})$, where $i = 1, \dots, M$. The input $(x_1, \dots, x_T)$ is divided into M univariate sequences $\hat{x}^{(i)} \in \mathbb{R}^{1\times T}$. We aim to predict the L future values $X_{T+1}, \dots, X_{T+L}$.

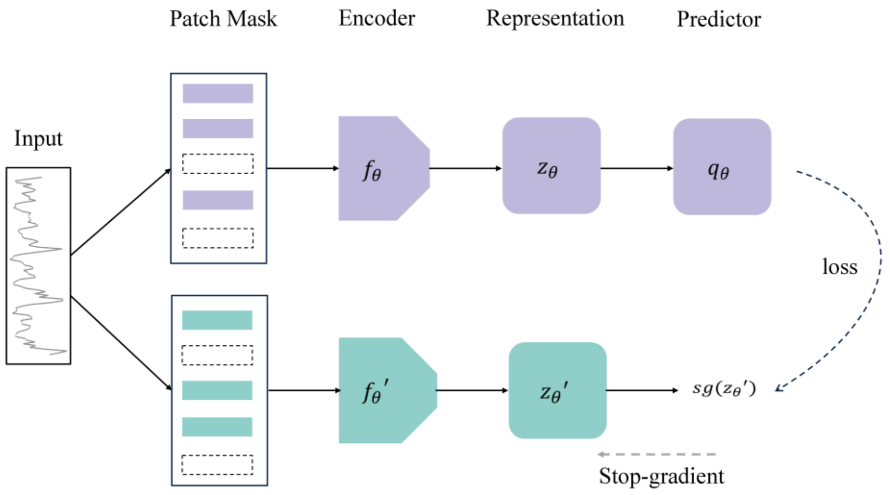

Fig. 2. Illustration of our proposed SimPM.

3.2 Overall Architecture

The sequence $\hat{x}^{(i)}$ is divided into non-overlapping patches. We represent the patch length as P and the non-overlapping region between two adjacent patches as stride S. The concatenation process generates the patch sequence $\hat{x}_p^{(i)} \in \mathbb{R}^{P\times N}$, where N is the number of patches, $N = \lfloor\frac{(T-P)}{S}\rfloor + 2$. Here, we pad the end of the original sequence with S repeated instances of the last value $x_T^{(i)} \in \mathbb{R}$ before concatenation.

$$\hat{x}_p^{(i)} = \{(\hat{x}_1^{(i)}, \dots, \hat{x}_P^{(i)}), (\hat{x}_{S+1}^{(i)}, \dots, \hat{x}_{S+P}^{(i)}), \dots\} \tag{1}$$

We apply two independent masks to the patch sequence $\hat{x}_p^{(i)}$ to generate two new contextual views. Specifically, let us assume we have a random variable $M_p \sim \text{Bernoulli}(p)$, representing the mask values sampled from a Bernoulli distribution. The mask function can be expressed as:

$$\text{mask}(\hat{x}_p^{(i)}, p) = \hat{x}_p^{(i)} \odot M_p \tag{2}$$

where $\odot$ denotes element-wise multiplication. Now, we can use the mask function to generate two new contextual views for the patch sequence, $\hat{x}_{p,1}^{(i)}$ and $\hat{x}_{p,2}^{(i)}$.

错误!未找到引用源。 shows the overall architecture of SimPM. We input two randomly augmented views, $\hat{x}_{p,1}^{(i)}$ and $\hat{x}_{p,2}^{(i)}$, into the network, which we will denote as x_1 and x_2 for simplicity. These two views are processed by the encoder network f. The encoder consists of two components: an input projection layer and a Transformer Encoder/MLP module. The encoder f shares weights between the two views. For each input w, the input projection layer is a fully connected layer that maps the input w to a high-dimensional latent variable z. A Transformer Encoder/MLP module is then applied to extract the contextual representation of each patch, z_1 and z_2.

Afterward, z_1 is transformed into q_1 by a MLP predictor. The MLP predictor operates on the global context information and is designed to produce a high-level representation. It is also designed to predict the mathematical expectation over the data augmentation distribution and correct the errors introduced by the randomness of data augmentation. This process is formalized as follows:

$$\begin{aligned} z_1, z_2 &= \text{Enc}(\text{Proj}(x_1),\text{Proj}(x_2)) \\ q_1 &= \text{Pred}(z_1) \end{aligned} \tag{3}$$

We aim to maximize the mutual information $I(q_1; z_2)$ between the high-level representation q_1 and the contextual representation z_2, which can be expressed as:

$$I(q_1; z_2) = H(q_1) - H(q_1|z_2) \tag{4}$$

Here, $H(\cdot)$ represents entropy, and $H(\cdot \mid \cdot)$ represents conditional entropy. In practice, as computing and maximizing the mutual information directly is challenging, we approximate this by minimizing the negative cosine similarity between q_1 and z_2:

$$\mathcal{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2} \tag{5}$$

$\|\cdot\|_2$ is ℓ_2-norm. To prevent model collapse, we perform a Stop-grad operation [7] on z_2. We define a symmetrized loss as:

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(p_1, stopgrad(z_2)) + \frac{1}{2}\mathcal{D}(p_2, stopgrad(z_1)) \tag{6}$$

We collect the loss of each channel and take the average to get the overall objective loss:

$$\mathcal{L}_{total} = \frac{1}{M}\sum_{i=1}^{M} \mathcal{L}_i \tag{7}$$

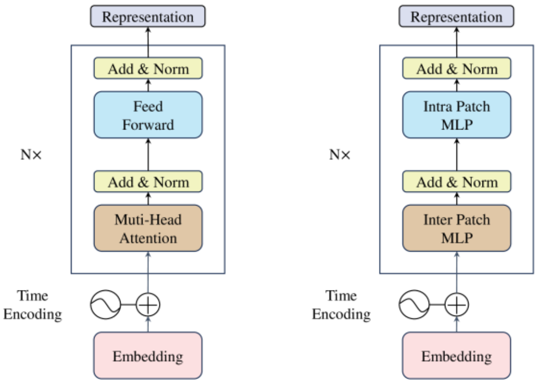

Fig. 3. Context Feature Extraction Module.

3.3 Model Variants

Two alternative variants of our model are available: SimPM/T, which is based on the transformer architecture, and SimPM/MLP, which utilizes a multilayer perceptron design. 错误!未找到引用源。 shows their encoders. The core mechanism of the transformer is the multi-head attention mechanism, which effectively captures long-range dependencies within sequences. However, due to the high computational complexity of the multi-head attention mechanism, the model's training and inference time costs are increased. Inspired by Dlinear [15], this paper offers the option of using lightweight and fast MLP layers. The MLP layers encode the input sequences through a series of fully connected layers and nonlinear activation functions, thereby extracting features from the sequences. By completely eliminating the computationally intensive multi-head attention, the MLP layers can achieve similar or even better performance compared to the transformer. A comprehensive description of the MLP block architecture is provided in Appendix 6.1.

4 Experiments

In this section, we conducted detailed experimental tests on SimPM and reported its comparison results with various time series representation learning methods. We also compared it with some of the most advanced end-to-end time series forecasting methods.

4.1 Experimental Setup

Datasets In our research, we utilized 7 of the most popular multivariate time series datasets for comparative representation learning. *Weather*[1] dataset collects 21 meteorological indicators such as humidity and temperature from Germany. *Traffic* [26] dataset records the road occupancy rate of different sensors on the San Francisco highway. *Electricity* [27] is a dataset describing the hourly electricity usage of 321 customers. *ETT* [28] is a crucial indicator for long-term power deployment. This dataset is composed of two years of data from two different counties.
ETTh1 and ETTh2 represent an hourly sampling frequency, while ETTm1 and ETTm2 labels represent a 15-minute sampling frequency. We emphasize that the Electricity, Traffic, and Weather datasets have a larger number of feature dimensions, which leads to more stable experimental results on these datasets. Detailed statistics of the datasets can be found in **Table 1**.

Table 1. Statistics of popular datasets for benchmark.

Datasets	Features	timestamps	ADF Test Statistic
ETTh1	7	17420	-5.909
ETTh2	7	17420	-4.136
ETTm1	7	69680	-14.985
ETTm2	7	69680	-6.225
Weather	21	52696	-26.661
Traffic	862	17544	-15.046
Electricity	321	26306	-8.483

Baselines To demonstrate the effectiveness of SimPM, we compared our method with the most recent and advanced representation learning methods and end-to-end learning methods. In our experiments, we compared our method with the following benchmarks:

- *LaST* [17] based on variational inference, is designed to separate the seasonal-trend representations in the latent space. It supervises and disentangles representations from their own perspectives and input reconstruction, and introduces a series of auxiliary objectives.

[1] https://www.bgc-jena.mpg.de/wetter/

- *TF-C* [14] embeds a time-based neighborhood of a sample close to its frequency-based neighborhood. The objective of this approach is to ensure that the time-based and frequency-based representations of the same sample are proximate in the time-frequency space, thereby providing superior consistency during the pre-training phase.
- *CoST* [12] is a time series representation learning framework for long sequence time series forecasting, which comprises both time domain and frequency domain contrastive losses to learn discriminative trend and seasonal representations, respectively.
- *TS2Vec* [13] enhances the robustness of contextual representations for individual timestamps through its implementation of augmented hierarchical context views.
- *Autoformer* [27] is the first to apply seasonal trend decomposition after each neural block. This is a standard method in time series analysis that makes the raw data more predictive.
- *Informer* [28] analyzed the attention mechanism in time series forecasting, proposing a sparse attention mechanism to save computational and time costs. By combining feature distillation and one-step time series forecasting, it achieved improvements in both efficiency and forecasting accuracy compared to Transformer.

Evaluation Setup We perform more challenging multivariate time series forecasting, rather than univariate forecasting on different datasets. Multivariate forecasting requires considering all feature dimensions of the dataset. Following the setup in the PatchTST, we divided the datasets into training/validation/test sets, and used MSE and MAE as evaluation metrics. All models follow the same experimental setup, with time series inputs undergoing zero-mean normalization. The input sequence length $T = 336$, but when testing the crop data augmentation method, we used a sequence input length $T = 500$. The forecast sequence length $L \in \{96,192,336,720\}$ for all dataset evaluations.

Implementation Details We conducted 50 epochs of pre-training on the encoder using the framework shown in **Fig. 1**. During the pre-training process, patches were set to be non-overlapping, the input length was chosen as 336, and the patch size was set to 12, generating 28 patches. We used a conventional 15% masking ratio, and the results of different masking ratios were analyzed in Section 4.3. We adopted the line probing strategy from PatchTST for testing, combining the pre-trained encoder with a linear prediction head, freezing the encoder parameters, and training only the linear head for 10 epochs. A comprehensive description of the prediction head can be found in Appendix 6.1. All our training and testing processes were conducted on a single NVIDIA RTX 4090 GPU.

4.2 Main Results

错误!未找到引用源。 presents a comprehensive comparative analysis of the performance of our SimPM/MLP and SimPM/T models relative to well-established benchmarks in the field of representational learning across a range of time-series forecasting tasks. Notably, SimPM models outperform these benchmarks, achieving a 17.4% reduction in Mean Squared Error (MSE) and a 12.5% reduction in Mean Absolute Error (MAE) compared to the most competitive baseline model, LaST. This superior performance of SimPM is particularly evident within the ETTh1 and Traffic datasets, where both SimPM/MLP and SimPM/T demonstrate substantial improvements in average MSE and MAE metrics, surpassing traditional representational learning approaches such as TF-C, TNC, and TS2Vec, and even the contemporary end-to-end learning method Autoformer. For example, in the case of the Traffic dataset, SimPM/T achieves an average MAE of 0.305, which is significantly lower than the 0.384 MAE of Autoformer and the 0.399 MAE of LaST, indicating a notable enhancement in forecasting accuracy. Although LaST and Autoformer exhibit commendable performance in certain forecasting scenarios, the SimPM approach consistently delivers superior results across all examined datasets. This indicates the high universality and robustness of the SimPM models, capable of delivering consistent and superior predictive performance across diverse time spans, particularly within the highly variable Traffic dataset.

Methods		Contrastive Representation Learning														End-to-End Learning			
		SimPM/MLP		SimPM/T		LaST		TF-C		TNC		TS2Vec		CoST		Autoformer		Informer	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.383	0.413	**0.382**	0.418	0.409	**0.412**	0.685	0.668	0.775	0.636	0.639	0.569	0.515	0.512	0.435	0.446	0.941	0.769
	192	**0.435**	0.453	0.438	**0.449**	0.489	0.469	0.649	0.673	0.877	0.694	0.733	0.619	0.661	0.591	0.456	0.457	1.007	0.786
	336	0.473	0.478	**0.472**	**0.478**	0.572	0.518	0.596	0.651	1.010	0.762	0.931	0.728	0.812	0.679	0.486	0.487	1.038	0.784
	720	0.526	0.528	0.526	0.532	0.771	0.668	0.611	0.671	1.152	0.834	1.063	0.799	0.970	0.771	**0.515**	**0.517**	1.144	0.857
	avg	**0.454**	**0.468**	0.455	0.469	0.560	0.517	0.635	0.666	0.954	0.732	0.842	0.679	0.740	0.638	0.473	0.477	1.033	0.799
ETTh2	96	0.373	0.403	0.372	0.383	0.391	0.430	1.631	0.985	1.022	0.778	0.979	0.780	1.062	0.801	**0.332**	**0.368**	1.549	0.952
	192	0.452	0.468	**0.421**	0.436	0.741	0.685	3.524	1.571	1.887	1.063	2.065	1.124	1.669	1.008	0.426	**0.434**	3.792	1.542
	336	0.493	0.502	0.473	**0.497**	**0.410**	0.518	3.310	1.386	2.512	1.227	2.194	1.197	1.846	1.075	0.477	0.497	4.215	1.642
	720	0.587	0.585	0.579	0.581	0.512	0.695	3.005	1.336	2.320	1.206	2.636	1.370	2.070	1.110	**0.453**	**0.490**	3.646	1.619
	avg	0.476	0.490	0.461	0.474	0.514	0.582	2.867	1.319	1.935	1.069	1.969	1.118	1.662	0.999	**0.422**	**0.447**	3.301	1.439
ETTm1	96	**0.306**	0.363	0.338	0.389	0.321	**0.358**	0.681	0.595	0.660	0.551	0.581	0.528	0.382	0.425	0.510	0.492	0.626	0.560
	192	**0.342**	0.381	0.382	0.413	0.357	**0.378**	0.697	0.668	0.713	0.589	0.618	0.553	0.431	0.459	0.514	0.495	0.725	0.619
	336	**0.379**	**0.403**	0.432	0.446	0.411	0.426	0.782	0.625	0.756	0.624	0.693	0.597	0.497	0.504	0.510	0.492	1.005	0.741
	720	**0.441**	**0.442**	0.487	0.485	0.501	0.474	0.878	0.701	0.814	0.676	0.782	0.653	0.639	0.585	0.527	0.493	1.113	0.845
	avg	**0.367**	**0.397**	0.410	0.433	0.398	0.409	0.759	0.647	0.736	0.610	0.669	0.583	0.487	0.493	0.515	0.493	0.867	0.691
ETTm2	96	0.206	0.311	0.361	0.420	**0.184**	**0.273**	0.361	0.426	0.369	0.432	0.341	0.418	0.315	0.403	0.205	0.293	0.355	0.462
	192	**0.245**	0.331	0.427	0.459	0.250	**0.324**	0.791	0.703	0.533	0.542	0.497	0.518	0.523	0.531	0.278	0.339	0.595	0.586
	336	0.345	0.403	0.538	0.518	0.352	0.396	1.225	0.895	0.892	0.718	0.795	0.672	0.801	0.687	**0.343**	**0.379**	1.270	0.871
	720	0.447	0.458	0.599	0.594	0.458	0.492	4.592	1.738	1.922	1.102	1.926	1.054	1.161	0.979	**0.414**	**0.419**	3.001	1.267
	avg	0.311	0.376	0.481	0.498	0.311	0.371	1.742	0.940	0.929	0.699	0.890	0.666	0.450	0.650	**0.310**	**0.358**	1.305	0.797
Weather	96	0.175	0.239	**0.169**	0.232	0.171	**0.210**	0.237	0.294	0.397	0.464	0.392	0.423	0.417	0.453	0.249	0.329	0.354	0.405
	192	0.217	0.282	0.209	0.267	0.209	**0.251**	**0.199**	0.424	0.482	0.505	0.506	0.512	0.474	0.495	0.325	0.370	0.419	0.434
	336	0.261	0.314	0.263	0.318	**0.260**	**0.288**	0.337	0.387	0.505	0.514	0.525	0.530	0.497	0.517	0.351	0.391	0.583	0.543
	720	0.321	0.356	0.329	0.367	**0.316**	**0.331**	0.377	0.375	0.543	0.547	0.556	0.552	0.533	0.542	0.415	0.426	0.916	0.705
	avg	0.244	0.298	0.243	0.296	**0.239**	**0.270**	0.287	0.370	0.482	0.508	0.495	0.504	0.480	0.502	0.335	0.379	0.568	0.522
Eletricity	96	0.165	0.268	**0.152**	**0.241**	0.158	0.246	0.395	0.431	0.434	0.477	0.354	0.419	0.177	0.279	0.196	0.313	0.304	0.393
	192	0.179	0.279	**0.164**	**0.251**	0.168	0.259	0.339	0.575	0.431	0.479	0.357	0.422	0.190	0.290	0.211	0.324	0.327	0.417
	336	0.194	0.294	**0.182**	**0.269**	0.185	0.275	0.457	0.478	0.434	0.480	0.373	0.434	0.206	0.306	0.214	0.327	0.333	0.422
	720	0.226	0.322	**0.214**	**0.298**	0.223	0.305	0.333	0.435	0.445	0.489	0.402	0.453	0.241	0.336	0.236	0.342	0.351	0.427
	avg	0.191	0.291	**0.178**	**0.265**	0.184	0.271	0.381	0.480	0.436	0.481	0.372	0.432	0.204	0.303	0.214	0.327	0.329	0.415
Traffic	96	0.427	0.302	**0.421**	**0.292**	0.722	0.395	0.589	0.293	0.915	0.513	1.039	0.575	0.766	0.447	0.597	0.371	0.733	0.410
	192	0.445	0.311	**0.437**	**0.302**	0.717	0.391	0.588	0.495	0.904	0.507	1.082	0.604	0.765	0.440	0.607	0.382	0.777	0.435
	336	0.463	0.323	**0.453**	**0.304**	0.728	0.393	0.750	0.490	0.907	0.511	1.110	0.611	0.774	0.441	0.623	0.387	0.776	0.434
	720	0.498	0.336	**0.486**	**0.321**	0.756	0.417	0.864	0.518	0.937	0.518	1.135	0.624	0.793	0.450	0.639	0.395	0.827	0.466
	avg	0.458	0.318	**0.449**	**0.305**	0.731	0.399	0.698	0.449	0.916	0.512	1.092	0.604	0.775	0.445	0.617	0.384	0.778	0.436

Table 2. Ablation study of various mask ratios on ETT and Weather datasets.

Mask ratio	0.1		0.2		0.3		0.4		0.5		Variance	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.462	0.474	0.453	0.467	0.453	0.467	0.451	0.466	0.453	0.468	1.504×10^{-5}	8.239×10^{-6}
Weather	0.236	0.284	0.254	0.31	0.259	0.319	0.257	0.316	0.258	0.322	7.336×10^{-5}	1.870×10^{-6}

4.3 Ablation Study

Mask Ratio To investigate the learning ability of the model under different masking ratios, we present in **Table 2** the forecasting results of SimPM under different masking ratios on the ETTh1 and Weather datasets, and calculate the variance under different masking ratios. We find that the fluctuations in the time series forecasting results of SimPM under different masking ratios are minimal. This indicates that SimPM can accurately reconstruct the original time series, demonstrating strong representation learning capabilities. Full results are in Appendix 6.2.

Data Augmentation Methods The use of data augmentation methods in contrastive learning is crucial, but due to the differences between modal data such as images and text and time series data, these methods can hardly be applied to time series data. Existing data augmentation methods are mostly used in the field of time series classification, and these methods often disrupt the sequential characteristics of time series in order to extract significant classification features. According to researches [29, 30], we conducted detailed ablation experiments on Crop, Jitter, Timewarp, timestamp Mask and Patch Mask methods on 5 datasets, further demonstrating the superiority of using the Patch Masking method. **Table 3** shows the average performance of different data augmentation methods. **错误!未找到引用源。** visualizes the different time series augmentation methods we selected.

Table 3. Ablation study of various data augments on all datasets.

Models	Patch Mask		Crop		Jitter		Timewarp		timestamp Mask	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.454**	**0.468**	0.459	0.473	0.457	0.517	0.455	0.469	0.459	0.473
ETTh2	**0.476**	**0.490**	0.534	0.546	0.495	0.526	0.538	0.552	0.500	0.529
ETTm1	0.367	0.397	0.375	0.403	**0.364**	**0.394**	0.368	0.397	0.669	0.583
ETTm2	0.311	0.376	0.311	0.377	**0.306**	**0.373**	0.309	0.375	0.323	0.546
Weather	0.244	0.298	0.240	0.290	**0.236**	**0.282**	0.240	0.294	0.241	0.291
Electricity	**0.191**	**0.291**	0.212	0.326	0.324	0.429	0.192	0.299	0.262	0.374
Traffic	**0.458**	**0.318**	0.472	0.323	0.513	0.339	0.549	0.367	0.514	0.347
Avg	**0.357**	**0.377**	0.372	0.391	0.385	0.409	0.379	0.393	0.424	0.449

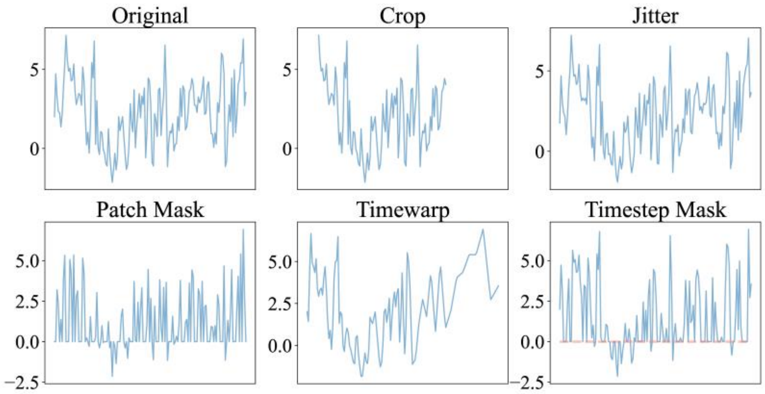

Fig. 4. Various data augmentations that are used in the experiments: jittering, time-warping, cropping, timestamp-masking, patch-masking methods.

Channel Independence In SimPM, we performed channel-independent operations on the data before feeding it into the encoder. In order to evaluate the overall impact of this operation on the model's performance, and to substantiate our explanation of the channel-independent method, we conducted an ablation study on the channel-independent operation. **Table 4** presents the results of both experiments, where "w/o CI" indicates that the channel-independent operation was not used. Our experimental results show that the application of channel-independent operations significantly enhances the learning capability of SimPM.

Table 4. Ablation study of channel independence operation on ETT and Weather datasets.

Method	SimPM		SimPM w/o CI	
Metrics	MSE	MAE	MSE	MAE
ETTh1	**0.454**	**0.468**	0.496	0.548
ETTh2	**0.476**	**0.490**	0.485	0.498
ETTm1	**0.367**	**0.397**	0.464	0.469
ETTm2	**0.311**	**0.376**	0.748	0.630
Weather	**0.244**	**0.298**	0.432	0.473

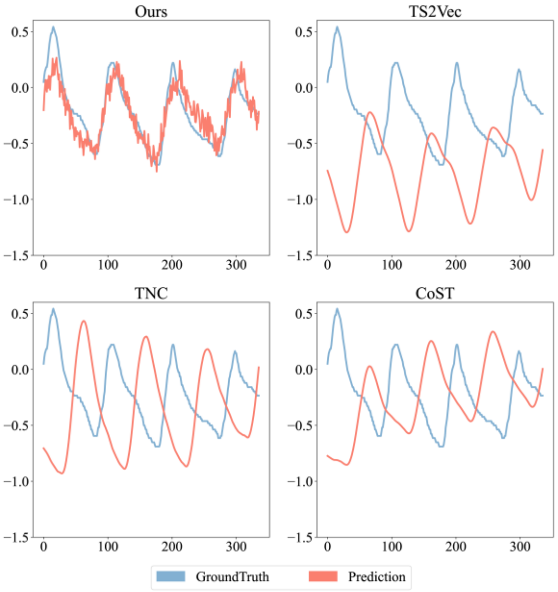

Fig. 5. Visualization of ETTm2 predictions by different models under the L= 336 setting. The blue lines stand for the ground truth and the red lines stand for predicted values.

4.4 Case Study

错误!未找到引用源。 provides a clear comparison of the results by different models. It can be seen that our proposed SimPM method can more accurately capture the changes in time. Other methods have effectively learned the seasonal information of the time series, but they are unable to accurately adjust for the trend fluctuations in the time series. More dataset case study are provided in Appendix 6.3.

5 Conclusion

In this paper, we have proposed a new framework called SimPM for time series forecasting. This method introduces the concept of channel independence and further proposes a patch-level mask data augmentation method to learn robust time features. Experiments show that our SimPM surpasses existing contrast learning methods and outperforms the state-of-the-art end-to-end models on some datasets. Ablation experiments have confirmed the effectiveness of the proposed data augmentation method and channel independence. In the future, we plan to apply SimPM to other downstream tasks (such as classification, anomaly detection, etc.), and further explore how to balance channel independence and channel relevance.

Acknowledgements. This research was partly supported by the National Key R&D Program of China (No.2023YFB3308601), the Talents by Sichuan provincial Party Committee Organization Department, and Chengdu - Chinese Academy of Sciences Science and Technology Cooperation Fund Project (Major Scientific and Technological Innovation Projects).

6 Supplement

6.1 Supplementary Figures

In this section, we supplement the structure of the MLP module in the model architecture diagram. Both the Intra-patch MLP and Inter-patch MLP mentioned in the main text adopt the structure shown in **Fig.** , with the difference being the exchange of the order of the partition and feature channels. **Fig.** illustrates the prediction head structure when testing with line probabilities.

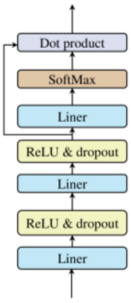

Fig. 6. MLP blocks

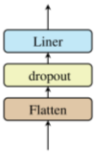

Fig. 7. Prediction head

6.2 Full Results

Table 6. Full results ablation study of various mask ratios on ETT and Weather datasets.

Mask ratio		0.1		0.2		0.3		0.4		0.5	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.385	0.415	0.382	0.412	0.382	0.412	0.382	0.412	0.383	0.412
	192	0.442	0.459	0.434	0.452	0.434	0.451	0.434	0.452	0.436	0.452
	336	0.480	0.484	0.471	0.477	0.471	0.477	0.472	0.477	0.477	0.481
	720	0.540	0.536	0.524	0.528	0.523	0.528	0.516	0.524	0.514	0.525
	avg	0.462	0.474	0.453	0.467	0.453	0.467	0.451	0.466	0.453	0.468
Weather	96	0.164	0.228	0.175	0.248	0.178	0.255	0.177	0.253	0.179	0.261
	192	0.206	0.261	0.222	0.288	0.224	0.294	0.224	0.294	0.224	0.298
	336	0.254	0.301	0.274	0.329	0.278	0.335	0.276	0.334	0.278	0.341
	720	0.321	0.347	0.346	0.377	0.355	0.391	0.352	0.384	0.349	0.387
	avg	0.236	0.284	0.254	0.311	0.259	0.319	0.257	0.316	0.258	0.322

6.3 Case Study

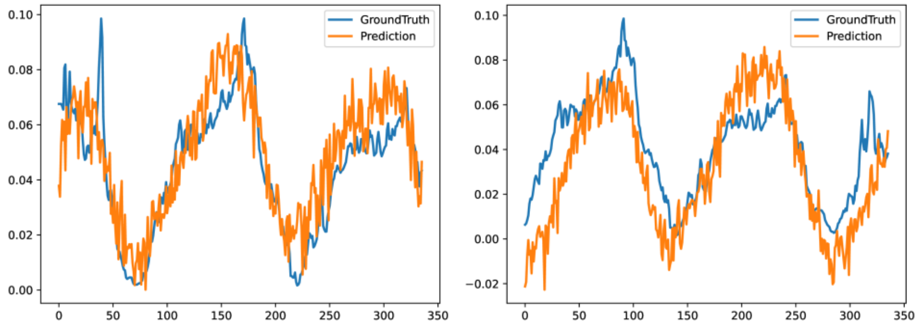

Fig. 8. Visualization of Weather predictions by different models under the \$L=336\$ setting.

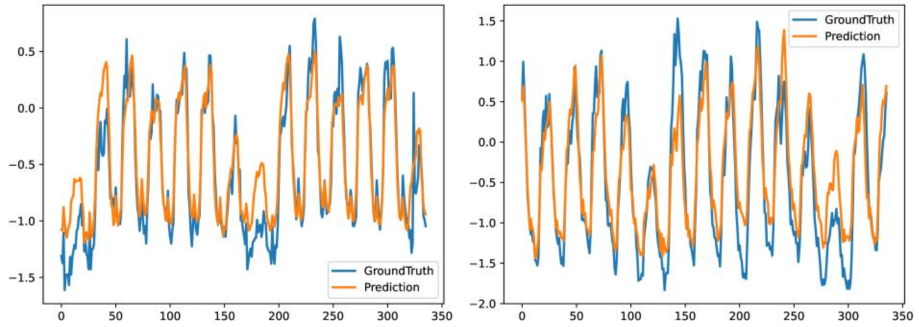

Fig. 9. Visualization of Electricity predictions by different models under the \$L=336\$ setting.

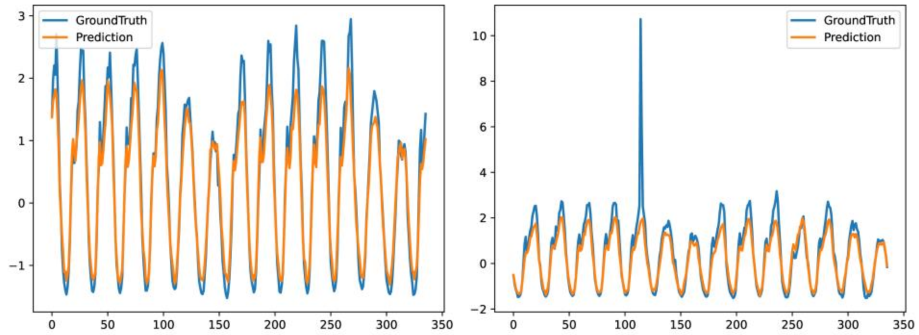

Fig. 10. Visualization of Traffic predictions by different models under the \$L=336\$ setting.

References

1. Wang, C., Wang, Z., Zhang, F., Pan, Y.: A New PM2.5 Concentration Predication Study Based on CNN-LSTM Parallel Integration. In: Intelligent Computing Theories and Application, pp. 258-266. Springer International Publishing, (2022)
2. You, J., Han, T., Shen, L.: From Uniform Models To Generic Representations: Stock Return Prediction With Pre-training. In: 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. (2022)
3. Wu, J., Zhu, W., Xiao, J.: An Adaptive Method for Generating the Traffic State Thresholds on Road Networks. In: Advanced Intelligent Computing Technology and Applications, pp. 15-26. Springer Nature Singapore, (2023)
4. Liu, Y., Zhang, Z., Qin, S.: NeuralHMM: A Deep Markov Network for Health Risk Prediction using Electronic Health Records. In: 2023 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. (2023)
5. Tang, C.I., Perez-Pozuelo, I., Spathis, D., Mascolo, C.: Exploring Contrastive Learning in Human Activity Recognition for Healthcare. (2020)
6. Grill, J., Strub, F., Altch, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.v., Guo, Z., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. (2020)
7. Chen, X., He, K.: Exploring Simple Siamese Representation Learning. In: {IEEE} Conference on Computer Vision and Pattern Recognition, {CVPR} 2021, virtual, June 19-25, 2021, pp. 15750-15758. Computer Vision Foundation / {IEEE}, (2021)
8. Zhang, R., Ji, Y., Zhang, Y., Passonneau, R.J.: Contrastive Data and Learning for Natural Language Processing. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts, pp. 39-47. Association for Computational Linguistics, (2022)
9. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In: 2018 {IEEE} Conference on Computer Vision and Pattern Recognition, {CVPR} 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 3733-3742. {IEEE} Computer Society, (2018)
10. Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C.K., Li, X., Guan, C.: Time-Series Representation Learning via Temporal and Contextual Contrasting. In: International Joint Conference on Artificial Intelligence, pp. 2352-2359. (2021)
11. Tonekaboni, S., Eytan, D., Goldenberg, A.: Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. In: 9th International Conference on Learning Representations, {ICLR} 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, (2021)
12. Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S.C.H.: CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. In: The Tenth International Conference on Learning Representations, {ICLR} 2022, Virtual Event, April 25-29, 2022. OpenReview.net, (2022)
13. Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., Xu, B.: TS2Vec: Towards Universal Representation of Time Series. In: Thirty-Sixth {AAAI} Conference on Artificial Intelligence, {AAAI} 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, {IAAI} 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, {EAAI} 2022 Virtual Event, February 22 - March 1, 2022, pp. 8980-8987. {AAAI} Press, (2022)

14. Zhang, X., Zhao, Z., Tsiligkaridis, T., Zitnik, M.: Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency. In: Conference on Neural Information Processing Systems. (2022)
15. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are Transformers Effective for Time Series Forecasting? AAAI 11121-11128 (2022)
16. Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C.-K., Li, X., Guan, C.: Self-supervised Contrastive Representation Learning for Semi-supervised Time-Series Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence PP, 15604-15618 (2023)
17. Wang, Z., Xu, X., Trajcevski, G., Zhang, W., Zhong, T., Zhou, F.: Learning Latent Seasonal-Trend Representations for Time Series Forecasting. In: Conference on Neural Information Processing Systems. (2022)
18. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? Advances in neural information processing systems 33, 6827-6839 (2020)
19. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. ICLR (2023)
20. Rangapuram, S.S., Kapoor, S., Nirwan, R.-S., Mercado, P., Januschowski, T., Wang, Y., Bohlke-Schneider, M.: Coherent Probabilistic Forecasting of Temporal Hierarchies. AISTATS 9362-9376 (2023)
21. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171-4186. Association for Computational Linguistics, (2019)
22. Gao, T., Fisch, A., Chen, D.: Making Pre-trained Language Models Better Few-shot Learners. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 3816-3830. Association for Computational Linguistics, (2021)
23. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: 9th International Conference on Learning Representations, {ICLR} 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, (2021)
24. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: 2021 {IEEE/CVF} International Conference on Computer Vision, {ICCV} 2021, Montreal, QC, Canada, October 10-17, 2021, pp. 9992-10002. {IEEE}, (2021)
25. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: CvT: Introducing Convolutions to Vision Transformers. In: 2021 {IEEE/CVF} International Conference on Computer Vision, {ICCV} 2021, Montreal, QC, Canada, October 10-17, 2021, pp. 22-31. {IEEE}, (2021)
26. Zheng, J.a.H., Mingfang: Traffic flow forecast through time series analysis based on deep learning. IEEE Access 8, 82562--82570 (2020)
27. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. NeurIPS 34, 22419-22430 (2021)
28. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: AAAI. (2021)
29. Um, T.T., Pfister, F.M.J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., Kulic, D.: Data Augmentation of Wearable Sensor Data for Parkinson's Disease Monitoring using

Convolutional Neural Networks. In: International Conference on Multimodal Interaction, pp. 216-220. (2017)

30. Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H.: Time Series Data Augmentation for Deep Learning: A Survey. In: International Joint Conference on Artificial Intelligence, pp. 4653-4660. (2020)