SimPSI: A Simple Strategy to Preserve Spectral Information in Time Series Data Augmentation

Hyun Ryu, Sunjae Yoon, Hee Suk Yoon, Eunseop Yoon, and Chang D. Yoo

Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea
{ryuhyun1905, sunjae.yoon, hskyoon, esyoon97, cd_yoo}@kaist.ac.kr

Abstract

Data augmentation is a crucial component in training neural networks to overcome the limitation imposed by data size, and several techniques have been studied for time series. Although these techniques are effective in certain tasks, they have yet to be generalized to time series benchmarks. We find that current data augmentation techniques ruin the core information contained within the frequency domain. To address this issue, we propose a simple strategy to preserve spectral information (`SimPSI`) in time series data augmentation. `SimPSI` preserves the spectral information by mixing the original and augmented input spectrum weighted by a preservation map, which indicates the importance score of each frequency. Specifically, our experimental contributions are to build three distinct preservation maps: magnitude spectrum, saliency map, and spectrum-preservative map. We apply `SimPSI` to various time series data augmentations and evaluate its effectiveness across a wide range of time series benchmarks. Our experimental results support that `SimPSI` considerably enhances the performance of time series data augmentations by preserving core spectral information. The source code used in the paper is available at https://github.com/Hyun-Ryu/simpsi.

Introduction

Time series data, whether univariate or multivariate, plays a crucial role in various domains such as medicine (Lipton et al. 2016), physiology (Jia et al. 2020), and sensory devices (Yao et al. 2017). Unfortunately, it is limited to collecting data samples under consideration of different types, constraining the performance and capabilities of neural networks that learn from it. To address this issue, data augmentation (Iwana and Uchida 2021; Um et al. 2017) is employed as a simple yet effective solution via artificially increasing the number of samples based on a slight variation or perturbation on the original samples.

Data augmentation techniques have been extensively studied for time series, incorporating methods such as Jittering, Scaling, Magnitude warping, Time warping, Permutation (Um et al. 2017), Shifting (Woo et al. 2022), and Dropout (Yang and Hong 2022). These perturbations have been popular choices in the time domain. The data augmentation is also considered in the frequency domain via applying the Fourier transform to time series data. The spectrum is then randomly perturbed before being converted back into the time domain through the inverse Fourier transform. Notable techniques in this category include Frequency masking, Frequency mixing (Chen et al. 2023), and Frequency adding (Zhang et al. 2022).

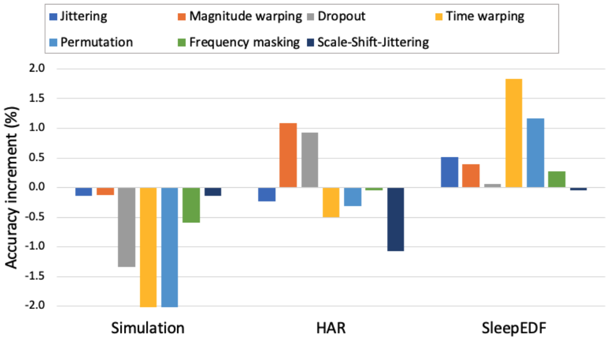

Figure 1: Dependency on data domain of time series data augmentation techniques. The plot shows the increment of classification accuracy of a baseline model after applying each data augmentation technique, which is evaluated on signal demodulation (Simulation), human activity recognition (HAR), and sleep stage detection (SleepEDF) tasks.

We have discovered that while the aforementioned data augmentation techniques show effectiveness in certain specific tasks (Um et al. 2017), they do not generalize well to time series classification benchmarks. Our experimental evidence in Fig. 1 presents the ungeneralized effectiveness of data augmentation techniques according to the datasets, such as signal demodulation, human activity recognition, and sleep stage detection.[1] Those techniques, though reliant on randomness, operate under the assumption that the core information within the data is preserved. However, the result suggests that perturbing the original time series data is heuristic and depends on the data domain, which leads to losing essential information necessary to solve the tasks.

The observed reduction in performance is attributed to

[1] Detailed information about the tasks and our experimental setup can be found in the Experiments section.

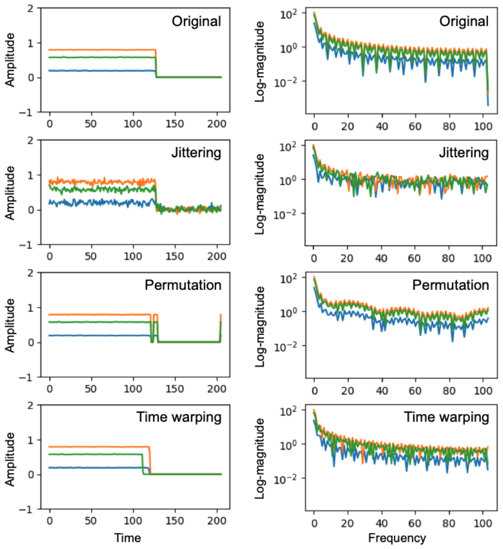

Figure 2: Visualization of a representative example from the HAR dataset in the time and frequency domain with various time series data augmentation techniques. Each color denotes a channel, and three channels are shown.

an implicit bias in the frequency domain introduced by each data augmentation technique. This bias alters the original data distribution. For example, Jittering adds a consistent amount of random noise across all frequencies, often obscuring subtle high-frequency components. Permutation, meanwhile, introduces abrupt changes at the boundaries of each fragment, consistently enhancing high-frequency components. Time warping globally distorts the temporal density of the original data, introducing even more spectral bias than Permutation. Fig. 2 provides illustrative examples of these data augmentation techniques.

In this paper, we introduce a simple strategy for preserving spectral information during time series data augmentation, which we refer to as `SimPSI`. Our strategy involves mixing the original spectral data and its augmented form, weighted by a preservation map. After applying any time series data augmentation technique, `SimPSI` converts the original and augmented time series to the frequency domain. It then combines the original spectrum with the augmented version based on the weightage given by the preservation map, which indicates the importance score for each frequency component. The combined spectrum is subsequently transformed back to the time domain, resulting in the final output of our framework. The remaining efforts concentrate on defining a well-structured preservation map. We propose three types of preservation maps: magnitude spectrum, saliency map, and spectrum-preservative map. The first two types use the given data's magnitude spectrum and saliency map (Simonyan, Vedaldi, and Zisserman 2013) as the preservation map. For the spectrum-preservative map, we developed a preservation map generator that takes input spectrum data and returns the preservation map. This map is learned through a preservation contrastive loss function that influences differentiated model output scores based on the preservation quality. We also propose a training strategy for improved optimization. To demonstrate the efficacy of `SimPSI`, we apply it to various time series data augmentation techniques and compare performance across different benchmarks. We also create a simulation to assess whether the proposed method correctly identifies spectral regions to preserve during data augmentation. Our experimental results demonstrate that `SimPSI` significantly enhances the effectiveness of time series data augmentation techniques by preserving essential spectral information, thereby preventing unintentional loss of core spectral details.

Related Works

Data Augmentation for Time Series

Various data augmentation techniques have been proposed for time series. One prevalent and intuitive strategy involves slightly altering the magnitude. For instance, Jittering (Um et al. 2017) introduces additive white Gaussian noise, Scaling (Um et al. 2017) multiplies by a random scalar value, Shifting (Woo et al. 2022) adds a random scalar value, Magnitude warping (Um et al. 2017) multiplies by a random polynomial curve, and Dropout (Yang and Hong 2022) masks random time indices. An alternative approach involves modifying the time scale rather than the magnitude. Time warping (Um et al. 2017), for instance, interpolates the time scale with a random polynomial curve, while Permutation (Um et al. 2017) rearranges the time order. An additional method involves perturbing the spectrum. Techniques such as Frequency masking, Frequency mixing (Chen et al. 2023), and Frequency adding (Zhang et al. 2022) serve as simple strategies that appropriately perturb global dependencies in the time domain.

Data Augmentation for Information Preservation

Data augmentation inherently introduces perturbations into the original data. If not appropriately managed, these perturbations could lead to significant information loss or an introduction of unnecessary noise and ambiguity. To mitigate this, studies have focused on information preservation. In the vision domain, KeepAugment (Gong et al. 2021) employs a saliency map (Simonyan, Vedaldi, and Zisserman 2013) of each image to identify and preserve informative regions during augmentation. AugMix (Hendrycks et al. 2020) generates a composite of various augmented views of the data and mixes it with the original data, weighted by a random scalar. This ensures the final image is not overly distanced from the original one. In natural language processing, SSMix (Yoon, Kim, and Park 2021) and SMSMix (Yoon et al. 2022) leverage saliency map to retain certain word sequences, ensuring that crucial information remains intact during the data augmentation process. For time series data, Input smoothing (Liu et al. 2022) scales high-frequency entries in the frequency domain by a random scalar, thereby reducing the impact of data noise. However, its application is limited to noise reduction, and the degree of reduction is randomly determined.

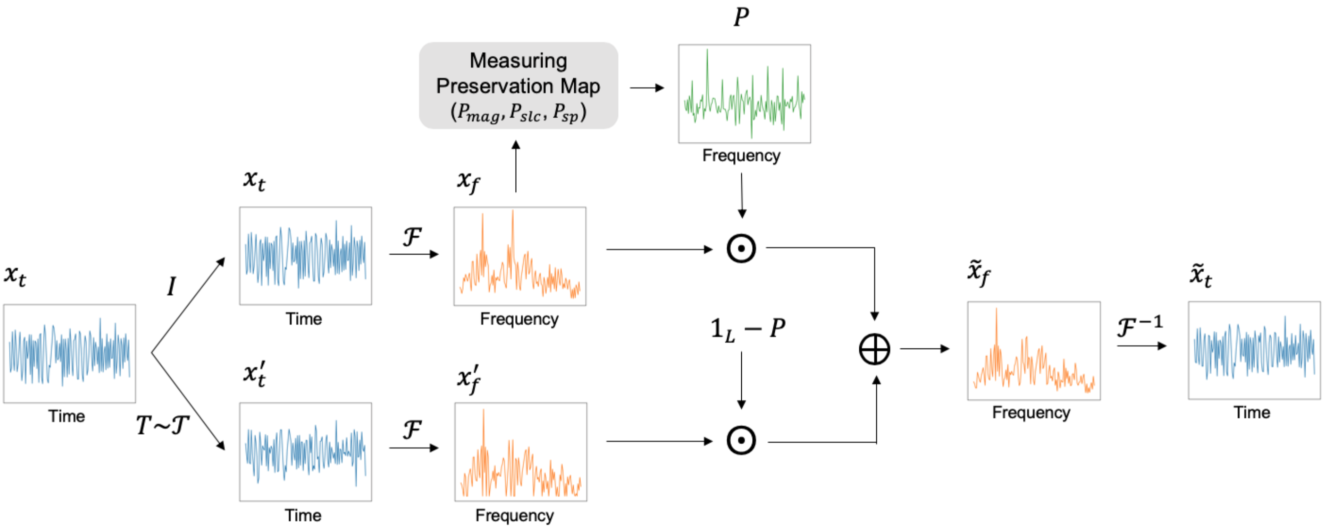

Figure 3: A SimPSI diagram. The original data is augmented randomly in the time domain. Then, the original and augmented data are both Fourier-transformed. The original spectrum is weighted by its preservation map, while the augmented spectrum is weighted by the negated preservation map, and those two are added. It is inverse-Fourier-transformed, which generates an information-preserved augmented view of the original time series data. We use a single-channel time series for better understanding, in which we visualize the real parts of the time series and magnitudes of spectra and omit channel-wise broadcasting.

Method

Mixing for Information Preservation

We transform an input time series $x_t \in \mathbb{C}^{C \times L}$, where C and L denote the number of channels and length of the input, to a spectrum $x_f \in \mathbb{C}^{C \times L}$ by the fast Fourier transform (FFT). Then, we apply data augmentation to x_t, which gives an augmented time series $x'_t \in \mathbb{C}^{C \times L}$. We transform x'_t to an augmented spectrum $x'_f \in \mathbb{C}^{C \times L}$ by the FFT. Then, we define a preservation map $P \in \mathbb{R}^L$ with the same length as the spectrum x_f, which indicates the importance score of each frequency component between 0 and 1. We mix the spectrum x_f and its augmented view x'_f with the preservation map P to produce an information-preserved spectrum $\tilde{x}_f \in \mathbb{C}^{C \times L}$ as follows:

$$\tilde{x}_f = (\mathbf{1}_C \cdot P^T) \odot x_f + (\mathbf{1}_C \cdot (\mathbf{1}_L - P)^T) \odot x'_f. \quad (1)$$

Since the preservation map P applies uniformly to different channels of the spectra, we broadcast it to the channel dimension to enable elementwise multiplication with the spectra. Frequencies with high importance score have a spectrum value closer to the data x_f than its augmentation x'_f, and those with low importance score have a spectrum value closer to x'_f than x_f. It enables us to retain important spectral regions and distort non-informative regions during augmentation. We transform $\tilde{x}_f$ back to an information-preserved time series $\tilde{x}_t \in \mathbb{C}^{C \times L}$ by applying inverse fast Fourier transform (IFFT), which is the final output of the proposed SimPSI. For classifier training, given a classifier $\hat{p}$, classification loss $\mathcal{L}_{cl}$ is calculated using the cross-entropy loss of the prediction score of $\tilde{x}_t$ and its label y as follows:

$$\mathcal{L}_{cl} = \mathcal{L}_{ce}(\hat{p}(y|\tilde{x}_t), y). \quad (2)$$

The following sections focus on defining the preservation map P, and we propose three methods: magnitude spectrum, saliency map, and spectrum-preservative map.

Efficient Implementation for Real-Valued Time Series. Most of the real-world time series data consists of real values. Using the conjugate symmetry property of the Fourier transform for real-valued time series, we take the first half of the spectrum, in which the dimensions of the spectrum reduce to $\{x_f, x'_f, \tilde{x}_f\} \in \mathbb{C}^{C \times (\lfloor L/2 \rfloor + 1)}$ while the dimensions of time series change to $\{x_t, x'_t, \tilde{x}_t\} \in \mathbb{R}^{C \times L}$. The dimension of the preservation map also reduces to $P \in \mathbb{R}^{\lfloor L/2 \rfloor + 1}$.

Magnitude Spectrum

We introduce a magnitude spectrum P_{mag} for preserving spectral information, assuming frequencies with large magnitudes are informative while those with small magnitudes are mainly non-informative noise. Given an input spectrum $x_f \in \mathbb{C}^{C \times L}$, we calculate the magnitude spectrum $|x_f| \in \mathbb{R}^{C \times L}$ and take the channel-wise maximum $|x_f|_{max} \in \mathbb{R}^L$ to aggregate the channel information as follows:

$$P_{mag} = Norm(|x_f|_{max}) \quad (3)$$

where $Norm$ is a min-max normalization so that values of the magnitude spectrum $P_{mag} \in \mathbb{R}^L$ are between 0 and 1. Preserving frequencies with large magnitudes makes the original and augmented data look alike, but the core information for solving the task might disappear. For instance, detecting abnormalities in the Electrocardiogram (ECG) signals relies on capturing the pattern of small high-frequency components (Tragardh and Schlegel 2006), whereas the magnitude spectrum P_{mag} eliminates the core frequencies for the classification during the data augmentation process just because those have a small magnitude.

Algorithm 1: `SimPSI` (Spectrum-Preservative Map) Pseudocode

```
Input: Given an input time series $x_t$, label $y$, preservation map generator $G(\cdot)$, classifier $\hat{p}$, and data augmentation $\mathcal{T}$
function AugmentAndPreserve($x_t$, $x_f$, $P$)
    Sample operation $T \sim \mathcal{T}$
    $x'_t = T(x_t)$                          ▷ Apply data augmentation
    $x'_f = \text{FFT}(x'_t)$
    $\tilde{x}_f = (\mathbf{1}_C \cdot P^T) \odot x_f + (\mathbf{1}_C \cdot (\mathbf{1}_L - P)^T) \odot x'_f$
    $\tilde{x}_t = \text{IFFT}(\tilde{x}_f)$
    return $\tilde{x}_t$
end function
$x_f = \text{FFT}(x_t)$
$\tilde{x}_t$ = AugmentAndPreserve($x_t$, $x_f$, $G(x_f)$)
Compute classification loss $\mathcal{L}_{cl} = \mathcal{L}_{ce}(\hat{p}(y|\tilde{x}_t), y)$
Sample random preservation map $n_f \sim U(0, 1)$
$\tilde{x}_t^{rnd}$ = AugmentAndPreserve($x_t$, $x_f$, $n_f$)
$\tilde{x}_t^{+}$ = AugmentAndPreserve($x_t$, $x_f$, $G(x_f)$)     ▷ $\tilde{x}_t \neq \tilde{x}_t^{+}$
$\tilde{x}_t^{-}$ = AugmentAndPreserve($x_t$, $x_f$, $1 - G(x_f)$)
Compute classification loss $\mathcal{L}_{cl}^{rnd}$, $\mathcal{L}_{cl}^{+}$, and $\mathcal{L}_{cl}^{-}$
for $\tilde{x}_t^{rnd}$, $\tilde{x}_t^{+}$, and $\tilde{x}_t^{-}$, respectively
Compute preservation contrastive loss $\mathcal{L}_{pc}$
$= max(\mathcal{L}_{cl}^{+} - \mathcal{L}_{cl}^{rnd} + \beta_1, 0) + max(\mathcal{L}_{cl}^{+} - L_{cl}^{-} + \beta_2, 0)$
Loss output: $\mathcal{L}_{cl}$, $\mathcal{L}_{pc}$
```

Saliency Map

We present a saliency map for time series, P_{slc}, to find informative spectral regions regardless of their magnitudes. Given an input spectrum $x_f \in \mathbb{C}^{C \times L}$, we transform it to a time series $x_t \in \mathbb{C}^{C \times L}$ by the IFFT and feed x_t into the classifier $\hat{p}$ to obtain the corresponding label logit value $\hat{f}(y|x_t)$. Then, we calculate an absolute value of a gradient of the logit value $\hat{f}(y|x_t)$ with respect to the input spectrum x_f and take the channel-wise maximum $|\nabla_{x_f} \hat{f}(y|x_t)|_{max} \in \mathbb{R}^L$ to aggregate the channel information as follows:

$$P_{slc} = Norm(|\nabla_{x_f} \hat{f}(y|\mathcal{F}^{-1}(x_f))|_{max}) \tag{4}$$

where $Norm$ is a min-max normalization to make values of the saliency map $P_{slc} \in \mathbb{R}^L$ between 0 and 1. However, it has a practical problem that the preservation quality solely depends on the training dynamics of the classifier, which could lead to an unstable performance. In addition, calculating the saliency map takes a significant amount of time backpropagating the gradients, which incurs a computational burden.

Spectrum-Preservative Map

We introduce a spectrum-preservative map P_{sp}, incorporating a preservation map generator G on top of the classifier $\hat{p}$, alleviating the unstable training dynamics of the saliency map. It is also a feedforward network that does not require any additional backpropagation of the classifier $\hat{p}$ during estimating the preservation map, resolving the computational burden. The following describes how to design the preservation map generator G, what objective functions are used, and how to train it with the classifier $\hat{p}$.

Preservation Map Generator. Given an input spectrum $x_f \in \mathbb{C}^{C \times L}$, we concatenate real and imaginary parts of x_f into a channel dimension, which the dimension changes to $x_f \in \mathbb{R}^{2C \times L}$. Then, x_f is fed into a two-layer transformer encoder to capture the underlying context of spectral representation. The output of the last layer is averaged over the channel dimension to aggregate the channel information and passes through the sigmoid function to make the values between 0 and 1 as follows:

$$P_{sp} = G(x_f) = Sigmoid(Enc(x_f)_{mean}). \tag{5}$$

Preservation Contrastive Loss. To train the preservation map generator G, we introduce a preservation contrastive loss $\mathcal{L}_{pc}$. Assume that an input spectrum $x_f \in \mathbb{C}^{C \times L}$, augmented spectrum $x'_f \in \mathbb{C}^{C \times L}$, and corresponding spectrum-preservative map P_{sp} are given. We define an information-preserved spectrum $\tilde{x}_f^{+} = (\mathbf{1}_C \cdot P_{sp}^T) \odot x_f + (\mathbf{1}_C \cdot (\mathbf{1}_L - P_{sp})^T) \odot x'_f$ and a spectrum that preserves the inverted preservation map $\tilde{x}_f^{-} = (\mathbf{1}_C \cdot (\mathbf{1}_L - P_{sp})^T) \odot x_f + (\mathbf{1}_C \cdot P_{sp}^T) \odot x'_f$. Then, the classifier $\hat{p}$ should predict $\tilde{x}_t^{+}$ better than $\tilde{x}_t^{-}$. Furthermore, if we define a randomly-preserved spectrum $\tilde{x}_f^{rnd} = (\mathbf{1}_C \cdot n_f^T) \odot x_f + (\mathbf{1}_C \cdot (\mathbf{1}_L - n_f)^T) \odot x'_f$ where $n_f \in \mathbb{R}^L$ is a random noise sampled from $U(0, 1)$, then the prediction score of $\tilde{x}_t^{rnd}$ by the classifier $\hat{p}$ should be in between those of $\tilde{x}_t^{+}$ and $\tilde{x}_t^{-}$. We notate $\tilde{x}_t^{\{+,-,rnd\}}$ and $\tilde{x}_f^{\{+,-,rnd\}}$ as Fourier transform pairs of time series and corresponding spectrum. These constraints can be formulated as follows:

$$\hat{p}(y|\tilde{x}_t^{+}) > \hat{p}(y|\tilde{x}_t^{rnd}) > \hat{p}(y|\tilde{x}_t^{-}). \tag{6}$$

We then define the corresponding classification loss using the cross-entropy loss for these three predictions, $\mathcal{L}_{cl}^{+} = \mathcal{L}_{ce}(\hat{p}(y|\tilde{x}_t^{+}), y)$, $\mathcal{L}_{cl}^{rnd} = \mathcal{L}_{ce}(\hat{p}(y|\tilde{x}_t^{rnd}), y)$, and $\mathcal{L}_{cl}^{-} = \mathcal{L}_{ce}(\hat{p}(y|\tilde{x}_t^{-}), y)$ where y is the label of the input time series x_t, and translate the constraints into an objective function as follows:

$$\begin{aligned} \mathcal{L}_{pc} = max(&\mathcal{L}_{cl}^{+} - \mathcal{L}_{cl}^{rnd} + \beta_1, 0) \\ &+ max(\mathcal{L}_{cl}^{+} - \mathcal{L}_{cl}^{-} + \beta_2, 0) \end{aligned} \tag{7}$$

where β_1 and β_2 are hyperparameters satisfying $\beta_1 < \beta_2$.

Model Training and Inference. We use two objective functions for model training, classification loss $\mathcal{L}_{cl}$ for classifier training and the preservation contrastive loss $\mathcal{L}_{pc}$ for preservation map generator training. We separate the training procedure, updating the classifier $\hat{p}$ by $\mathcal{L}_{cl}$ with the preservation map generator G froze, and then updating the preservation map generator G by $\mathcal{L}_{pc}$ with the classifier $\hat{p}$ froze. It can be formulated as follows:

$$\begin{aligned} \hat{\theta}_p &= \underset{\theta_p}{\text{argmin}}\, \mathcal{L}_{cl}(x_t|\theta_G, \theta_p) \\ \hat{\theta}_G &= \underset{\theta_G}{\text{argmin}}\, \mathcal{L}_{pc}(x_t|\theta_G, \theta_p) \end{aligned} \tag{8}$$

where θ_G and θ_p are the parameters of the preservation map generator G and the classifier $\hat{p}$, respectively. Note that θ_G

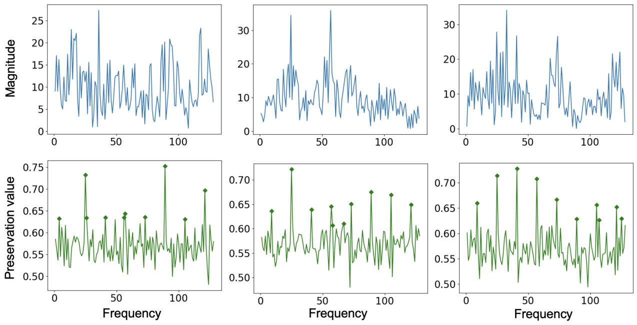

Figure 4: Finding a set of frequencies to preserve using SimPSI (Spectrum-preservative map) during Frequency masking. The top row shows representative input magnitude spectra from the FSK8 test set. The bottom row shows the corresponding learned preservation map where the ten largest values are marked as diamonds.

does not descend towards the gradient of $\mathcal{L}_{cl}$. This prevents G from learning undesirable local minima, such as returning the uniform scalar value across different frequencies or the same map across different samples, in which the preservation map is not adaptive to the input time series but acts as a uniform band-pass filter. Also, the preservation map generator G is removed during inference, so G updated by the classification loss might interrupt classifier training.

Experiments

Signal Demodulation (Simulation)

Experimental Setting. We verified if the proposed method improves the performance of existing data augmentation techniques by capturing important spectral regions and preserving them. To do that, we devised a simulation where information is carried on a set of known frequencies. Inspired by the wireless communication domain (Ryu and Choi 2023), we constructed a synthetic dataset by modulating a sequence of random bits into the corresponding frequencies of a signal, called frequency shift keying (FSK), and the task is demodulating it. We used 8 and 32 different frequencies for modulation (FSK8 and FSK32), and each dataset consists of 2,304 training signals, 288 validation signals, and 288 testing signals. We chose ResNet1D (Ramjee et al. 2020) as a baseline network, which was given a 128-length modulated signal and returned a 32-length M-ary (M=8, 32) sequence. We used the Adam optimizer (Kingma and Ba 2015) with the learning rate 10^{-3}, and the networks were trained for 50 epochs. The training was performed on a single NVIDIA RTX A6000 GPU. Appendix A provides more details about our experimental setup.

Performance Enhancement through SimPSI. The performance improvement of random augmentations by the proposed method on the FSK32 dataset is described in Table 1. The accuracies of Jittering, Scale-Shift-Jittering, and Frequency masking were increased by 1.5%, 1.4%, and 1.5%, respectively, using the spectrum-preservative map.

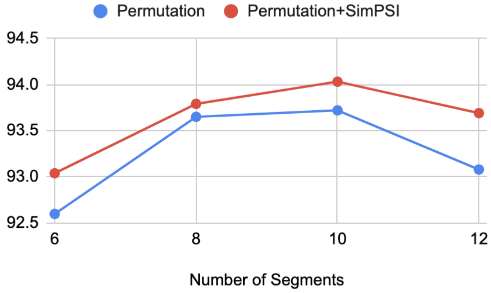

Figure 5: Testing accuracy of a 3-layer CNN model trained on the HAR dataset using Permutation with and without SimPSI (Spectrum-preservative map) while varying the maximum number of segments.

Learned Preservation Map. We then verified whether the learned preservation map genuinely preserves the informative frequency components during augmentation. We displayed learned preservation maps of representative samples from the FSK8 test set in Fig. 4. We could observe eight equally-spaced frequencies that were preserved the most during Frequency masking. It perfectly matches the data generation process since we used those eight frequencies for signal modulation. The other frequencies did not contain information and showed a preservation value of around 0.5, meaning those components barely attributed to achieving Eq. (6).

Human Activity Recognition (HAR)

Experimental Setting. In the HAR dataset (Reyes-Ortiz et al. 2012), data is collected by the accelerometer and gyroscope of a smartphone mounted on a waist and sampled at 50 Hz, which aims to classify human activities. Following the data preprocessing in (Eldele et al. 2021), an input time series has a length of 128 and nine channels. The dataset consists of 7,352 training samples and 2,947 test samples labeled with six classes. We chose a 3-layer CNN model for classification, which was used in (Wang, Yan, and Oates 2017; Eldele et al. 2021; Zhang et al. 2022), and additionally included a 2-layer LSTM model and a 2-layer Transformer model for further verifications. We used the Adam optimizer with the learning rate 10^{-3}, and the networks were trained for 100 epochs. We adhered to configurations in (Eldele et al. 2021), and the training was performed on a single NVIDIA RTX A6000 GPU. Appendix B and C provide more details about our experimental setup.

Performance Enhancement through SimPSI. We compared the performance of the model with and without SimPSI to evaluate the impact of SimPSI on recognition accuracy. To inspect its impact thoroughly, we performed experiments on three perspectives: different random augmentations, model architectures, and distortion magnitudes.

The performance increase of data augmentations by SimPSI on the HAR dataset is described in Table 1. We also tested an intuitive method that mixes the original data and its augmented view with a random preservation map

Model	Simulation		HAR		SleepEDF	
	Accuracy	Δ	*Accuracy*	Δ	*Accuracy*	Δ
None	94.8 ± 0.1	N/A	94.0 ± 0.8	N/A	80.7 ± 0.1	N/A
Jittering (Um et al. 2017)	94.6 ± 0.1	–0.2	93.8 ± 0.8	–0.2	81.2 ± 0.4	+0.5
+ Random preservation map	96.2 ± 0.3	+1.4	93.6 ± 0.5	–0.4	81.2 ± 0.5	+0.5
+ `SimPSI` (Magnitude spectrum)	94.9 ± 0.3	+0.1	**94.2 ± 0.5**	**+0.2**	81.0 ± 0.3	+0.3
+ `SimPSI` (Saliency map)	96.2 ± 0.2	+1.4	93.7 ± 0.6	–0.3	**81.4 ± 0.4**	**+0.7**
+ `SimPSI` (Spectrum-preservative map)	**96.3 ± 0.1**	**+1.5**	94.1 ± 0.3	+0.1	81.3 ± 0.4	+0.6
Scale-Shift-Jittering (Woo et al. 2022)	94.6 ± 0.2	–0.2	92.8 ± 0.5	–1.2	80.7 ± 0.2	0
+ Random preservation map	95.3 ± 0.1	+0.5	94.0 ± 1.3	0	80.4 ± 1.5	–0.3
+ `SimPSI` (Magnitude spectrum)	94.9 ± 0.1	+0.1	**94.9 ± 1.0**	**+0.9**	**81.4 ± 0.7**	**+0.7**
+ `SimPSI` (Saliency map)	95.6 ± 0.3	+0.8	94.8 ± 0.2	+0.8	81.2 ± 0.5	+0.5
+ `SimPSI` (Spectrum-preservative map)	**96.2 ± 0.1**	**+1.4**	94.8 ± 0.6	+0.8	80.9 ± 1.0	+0.2
Frequency masking (Chen et al. 2023)	94.2 ± 0.3	–0.6	93.9 ± 1.7	–0.1	81.0 ± 0.4	+0.3
+ Random preservation map	96.2 ± 0.3	+1.4	**95.0 ± 0.6**	**+1.0**	81.5 ± 0.5	+0.8
+ `SimPSI` (Magnitude spectrum)	94.9 ± 0.1	+0.1	**95.0 ± 0.4**	**+1.0**	80.5 ± 0.4	–0.2
+ `SimPSI` (Saliency map)	**96.3 ± 0.1**	**+1.5**	**95.0 ± 0.5**	**+1.0**	80.2 ± 1.4	–0.5
+ `SimPSI` (Spectrum-preservative map)	**96.3 ± 0.2**	**+1.5**	**95.0 ± 0.5**	**+1.0**	**81.7 ± 0.2**	**+1.0**

Table 1: Performance on Signal Demodulation (Simulation test set), Human Activity Recognition (HAR test set), and Sleep Stage Detection (SleepEDF test set) using different random augmentations with and without `SimPSI`. Accuracy and its increment from not using augmentation are reported with three different seeds.

Model	3-layer CNN	2-layer LSTM	2-layer Transformer
Jittering (Um et al. 2017)	95.1	92.2	94.2
+ `SimPSI` (P_{sp})	**95.2**	**93.7**	**96.4**

Table 2: Performance on Human Activity Recognition using various model architectures with and without `SimPSI` (Spectrum-preservative map). AUPRC scores are averaged over three different seeds.

sampled from $U(0, 1)$. The accuracy of Jittering is enhanced by 0.2% using the magnitude spectrum, while the random preservation map decreases it by 0.4%. The accuracy of Scale-Shift-Jittering is improved by 0.9% using the magnitude spectrum, and Frequency masking is improved by 1.0% using all the proposed preservation maps. Appendix D provides more experimental results, and Appendix E provides the training time cost of the preservation maps.

Using Jittering, we compared three types of networks for time series classification: CNN, LSTM, and Transformer (Table 2). `SimPSI` increased the area under the precision-recall curve (AUPRC) of a 3-layer CNN by 0.1, while it increased the AUPRC of a 2-layer LSTM by 1.5 and a 2-layer Transformer by 2.2.

Using Permutation, we also tested `SimPSI` while varying the distortion magnitude of data augmentation. We changed the maximum number of segments of Permutation and compared the accuracy with and without `SimPSI` (Fig. 5). `SimPSI` consistently improved the performance of Permutation regardless of its distortion strength, alleviating the performance drop while the number of segments increased. Specifically, comparing the performance at 10 and 12 segments, Permutation alone dropped the accuracy by 0.6, while Permutation with `SimPSI` dropped it by 0.3.

Sleep Stage Detection (SleepEDF)

Experimental Setting. We used the SleepEDF dataset (Goldberger et al. 2000) for classifying sleep stages from Electroencephalogram (EEG) signals sampled at 100 Hz. We followed the data preprocessing in (Eldele et al. 2021), where the input has a length of 3,000 and a single channel. The dataset comprises 35,503 training samples and 6,805 test samples labeled with five classes. We chose a 3-layer CNN model for classification, also used in the human activity recognition experiments. We used the Adam optimizer with the learning rate 10^{-3}, and the networks were trained for 40 epochs. We adhered to configurations in (Eldele et al. 2021) for other details. The training was performed on a single NVIDIA RTX A6000 GPU.

Performance Enhancement through SimPSI. The performance improvement of data augmentations by `SimPSI` on the SleepEDF dataset is summarized in Table 1. `SimPSI` increased the detection accuracy of Jittering by 0.7% using the saliency map, Scale-Shift-Jittering by 0.7% using the magnitude spectrum, and Frequency masking by 1.0% using the spectrum-preservative map. We note that the spectrum-preservative map outperformed the random preservation map regardless of the baseline augmentation techniques, supporting the effectiveness of the information-preserving approach.

Model	*Accuracy*	*AUPRC*
Scale-Shift-Jittering	92.0 ± 1.9	62.6 ± 2.3
+ SimPSI (P_{sp})	**95.2** ± **0.3**	**64.7** ± **2.0**
+ SimPSI (P_{sp}) w/o $\mathcal{L}_{pc}$	94.9 ± 0.1	63.7 ± 2.1
+ SimPSI (P_{sp}) w/ joint training	94.5 ± 0.3	62.9 ± 1.9

Table 3: Ablation of SimPSI (Spectrum-preservative map) on Atrial Fibrillation Classification. Accuracy and AUPRC scores are reported with three different seeds.

Atrial Fibrillation Classification (Waveform)

We used the Waveform dataset (Moody 1983) for classifying rhythm types from ECG recordings of human subjects with atrial fibrillation. It was sampled at 250 Hz, and we followed the data preprocessing step as in (Tonekaboni, Eytan, and Goldenberg 2021). Every input has a length of 2,500 and two channels. The dataset comprises 59,922 training samples and 16,645 test samples labeled with four classes. We chose a 1-dimensional strided CNN with six convolutional layers and a total down-sampling factor 16, proposed in (Tonekaboni, Eytan, and Goldenberg 2021). We used the Adam optimizer with the learning rate 10^{-4}, and the networks were trained for 8 epochs. We adhered to configurations in (Tonekaboni, Eytan, and Goldenberg 2021) for other details. The training was performed on a single NVIDIA RTX A6000 GPU. Performance enhancement through SimPSI is described in Appendix D.

Ablations

We ablated the proposed method from two perspectives, verifying the impact of the preservation contrastive loss and separate training strategy. We showed the performance of a 6-layer CNN model on the Waveform dataset while a composition of Scaling, Shifting, and Jittering (Woo et al. 2022) was applied (Table 3). Removing the preservation contrastive loss resulted in a 0.3 decrease in accuracy and a 1.0 decrease in AUPRC. Applying joint training of the cross-entropy loss and the preservation contrastive loss made a 0.7 decrease in accuracy and a 1.8 decrease in AUPRC.

Discussions

SimPSI's Dependency on Data Domain. We observed the data augmentation techniques did not generalize well to time series benchmarks (Fig. 1). Specifically, no augmentation increased the accuracy on the Simulation dataset. However, SimPSI resolved the issue in which the information-preserved approaches consistently improved the performance regardless of the tasks (Fig. 6). As a result, SimPSI encouraged augmentation independent of the data domain by preserving core spectral information. Appendix F provides more results using different preservation maps.

Comparison of Preservation Maps. The averaged spectrum-preservative map for the HAR dataset showed that the few lowest frequencies were preserved better than the higher frequencies. The averaged saliency map showed a similar tendency, where the saliency value was high at the few lowest frequencies and fell to zero as the frequency increased. For the SleepEDF dataset, there were four distinct frequency clusters in the spectrum-preservative map, and we could find corresponding clusters in the saliency map. For the Waveform dataset, unlike the previous two datasets, high-frequency components are preserved more than the lower ones in both preservation maps. These are displayed in Fig. 7.

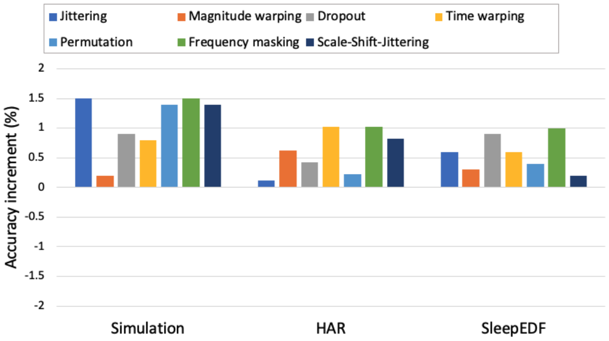

Figure 6: SimPSI's dependency on data domain. The plot shows the increment of classification accuracy of a baseline model after applying each data augmentation technique with SimPSI (Spectrum-preservative map), which is evaluated on Simulation, HAR, and SleepEDF datasets.

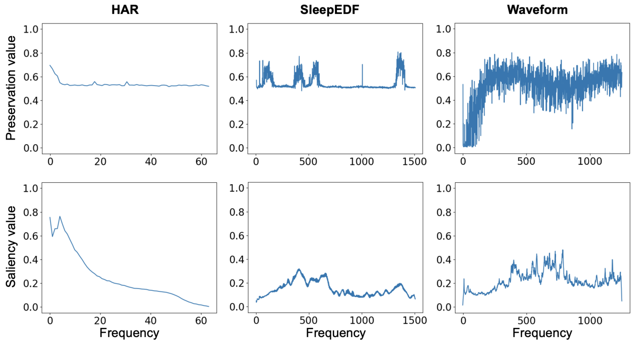

Figure 7: Comparison of spectrum-preservative maps and saliency maps. All the maps are averaged on the HAR, SleepEDF, and Waveform test sets. We used Jittering during training.

Conclusion

We presented a simple strategy to preserve spectral information (SimPSI) in time series data augmentation. Our investigation into the simulation task proved that the proposed method preserves the informative frequency components during augmentation. Our experimental results on various time series tasks with different data augmentation techniques illustrated the effectiveness of SimPSI in enhancing the model performance. We believe that SimPSI is a powerful tool to mitigate the data domain dependency of time series data augmentation techniques and improve the model performance in various time series tasks.

Acknowledgments

This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-01381, Development of Causal AI through Video Understanding and Reinforcement Learning, and Its Applications to Real Environments) and partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics).

References

Chen, M.; Xu, Z.; Zeng, A.; and Xu, Q. 2023. FrAug: Frequency Domain Augmentation for Time Series Forecasting. *arXiv preprint arXiv:2302.09292*.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C. K.; Li, X.; and Guan, C. 2021. Time-Series Representation Learning via Temporal and Contextual Contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2352–2359.

Goldberger, A. L.; Amaral, L. A.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C.-K.; and Stanley, H. E. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220.

Gong, C.; Wang, D.; Li, M.; Chandra, V.; and Liu, Q. 2021. Keepaugment: A simple information-preserving data augmentation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1055–1064.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Iwana, B. K.; and Uchida, S. 2021. An empirical survey of data augmentation for time series classification with neural networks. *Plos one*, 16(7): e0254841.

Jia, Z.; Cai, X.; Zheng, G.; Wang, J.; and Lin, Y. 2020. SleepPrintNet: A multivariate multimodal neural network based on physiological time-series for automatic sleep staging. *IEEE Transactions on Artificial Intelligence*, 1(3): 248–257.

Kingma, D.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*. San Diega, CA, USA.

Lipton, Z. C.; Kale, D. C.; Elkan, C.; and Wetzel, R. C. 2016. Learning to Diagnose with LSTM Recurrent Neural Networks. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Liu, X.; Liang, Y.; Huang, C.; Zheng, Y.; Hooi, B.; and Zimmermann, R. 2022. When do contrastive learning signals help spatio-temporal graph forecasting? In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 1–12.

Moody, G. 1983. A new method for detecting atrial fibrillation using RR intervals. *Proc. Comput. Cardiol.*, 10: 227–230.

Ramjee, S.; Ju, S.; Yang, D.; Liu, X.; El Gamal, A.; and Eldar, Y. C. 2020. Fast Deep Learning for Automatic Modulation Classification. *IEEE Transactions on Cognitive Communications and Networking*.

Reyes-Ortiz, J.; Anguita, D.; Ghio, A.; Oneto, L.; and Parra, X. 2012. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C54S4K.

Ryu, H.; and Choi, J. 2023. EMC^2-Net: Joint Equalization and Modulation Classification Based on Constellation Network. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Tonekaboni, S.; Eytan, D.; and Goldenberg, A. 2021. Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. In *International Conference on Learning Representations*.

Tragardh, E.; and Schlegel, T. T. 2006. High-frequency ECG.

Um, T. T.; Pfister, F. M.; Pichler, D.; Endo, S.; Lang, M.; Hirche, S.; Fietzek, U.; and Kulić, D. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*, 216–220.

Wang, Z.; Yan, W.; and Oates, T. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, 1578–1585. IEEE.

Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; and Hoi, S. 2022. CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. In *International Conference on Learning Representations*.

Yang, L.; and Hong, S. 2022. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*, 25038–25054. PMLR.

Yao, S.; Hu, S.; Zhao, Y.; Zhang, A.; and Abdelzaher, T. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th international conference on world wide web*, 351–360.

Yoon, H. S.; Yoon, E.; Harvill, J.; Yoon, S.; Hasegawa-Johnson, M.; and Yoo, C. 2022. SMSMix: Sense-Maintained Sentence Mixup for Word Sense Disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 1493–1502. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Yoon, S.; Kim, G.; and Park, K. 2021. SSMix: Saliency-Based Span Mixup for Text Classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3225–3234. Online: Association for Computational Linguistics.

Zhang, X.; Zhao, Z.; Tsiligkaridis, T.; and Zitnik, M. 2022. Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.