Measurement 204 (2022) 112041

Contents lists available at ScienceDirect

Measurement

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

NTScatNet: An interpretable convolutional neural network for domain generalization diagnosis across different transmission paths

Chao Liu [a], Xiaolong Ma [b], Tianyu Han [a], Xi Shi [a,*], Chengjin Qin [a], Songtao Hu [a]

[a] State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China
[b] Shanghai Institute of Aerospace System Engineering, Shanghai, PR China

ARTICLE INFO

Keywords:
Intelligent fault diagnosis
Interpretable convolutional neural network
Wavelet scattering transform
Domain generalization
Transducers across different transmission paths

ABSTRACT

The past decade has witnessed the convolutional neural network(CNN)'s significant progress in intelligent fault diagnosis research. Nevertheless, CNN's weak interpretability and poor domain generalization capability have been preventing its application in industrial practice. The current work develops an interpretable domain generalization diagnosis model, *i.e*, the normalized wavelet scattering convolutional network (NTScatNet), to remedy these gaps. The architecture of NTScatNet is similar to a standard CNN, while NTScatNet's distinctiveness is that it takes Morlet wavelet convolutional kernel, modulo activation function, and moving averaging pooling layer. This article gives a detailed physical interpretation of each layer of NTScatNet and shows that NTScatNet's feature extractor well characterizes the multi-scale cyclostationarity information of the fault signals. Besides, this work theoretically illustrates the normalized scattering feature's invariance to a linear time-invariant system, indicating that NTScatNet holds the domain generalization diagnosis capability across different transmission paths. Finally, NTScatNet's domain generalization ability across transmission paths is experimentally verified on the transfer diagnosis tasks across transducers and the tasks of detecting foreign objects on the escalator guide rail. Incidentally, this manuscript is a prelude to the second manuscript entitled "Scattering Moment Matching-based Interpretable Domain Adaptation for Transfer Diagnosis Tasks across Bearing specifications and Transducers", which proposes a novel interpretable domain adaptation method.

1. Introduction

Applying machine learning to machine fault diagnosis, referred to as the intelligent fault diagnosis(IFD) technology, has been a research hot in vibration-based machine condition monitoring [1]. IFD replaces diagnosis tasks with a classification problem based on the idea that diagnosis experience from the training dataset could generalize to new testing tasks. According to Antoni's overview paper [2], the IFD technique follows a "Filterbank-Feature-Decision" methodology. The "Filterbank" phase aims to enhance diagnostic information, which is conveyed by the cyclostationarity of the fault signals, with advanced signal processing techniques. Then the "Feature" phase codes diagnostic information into low-dimensional features. Finally, the "Decision" phase maps the handcrafted features to the health conditions of the monitored machine with a machine learning classifier. The rapidly developing convolutional neural network(CNN) technique skips the exhausting handcraft feature stage and fuses the "Filterbank-Feature-Decision" methodology into an end-to-end pipeline, providing an approach to automatically abstract fault representation from training data. However, the weak interpretability and poor domain generalization capability are two main obstacles preventing applying CNN-based diagnosis techniques into industrial practice.

The standard CNN architecture consists of a hierarchical parameterized feature extractor and a softmax layer. The softmax layer is an interpretable linear classifier. However, the representation learning process is not transparent to the diagnosis engineers due to complex architecture and training data-dependent filters' characteristics. Enhancing the interpretability of the feature representation modules helps enhance engineers' confidence in CNN's prediction and has attracted increasing research interest. One solution is post hoc explaining the trained CNN through weights visualization. For example, Lei et al. [3] found that the trained sparse filtering layer behaves as the Gabor wavelet. Jia *et al.*[4] work revealed that the convolutional layer in a deep network behaves similarly to multi-channel filterbanks. The other solution is enhancing CNN's intrinsic interpretability by restricting the complexity of network architecture. Li and Yan [5] introduced a continuous wavelet convolutional layer(CWConv) to replace CNN's first random initialization layer to simplify the representing learning

* Correspondence to: Shanghai Jiao Tong University, China.
E-mail address: xishi@sjtu.edu.cn (X. Shi).

https://doi.org/10.1016/j.measurement.2022.112041
Received 20 March 2022; Received in revised form 30 September 2022; Accepted 5 October 2022
Available online 13 October 2022

process. The CWConv layer is physically interpretable and outputs fault-related semantic information. However, the above work [3–5] focuses only on the first layer of CNN yet fails to whiten the other layers. Recently, Liu and Shi [6] introduced Mallat's wavelet scattering theory into IFD and developed a time scattering convolutional network(TScatNet) model for bearing diagnosis [7–12]. TScaNet's distinction from WaveletKernelNet is that all convolutional layers are wavelet kernels, not just the first layer. Since both Morlet wavelet transform and softmax layer are intrinsic interpretable, TScatNet is a fully interpretable convolutional network.

Besides the interpretability, CNN's poor domain generalization capability is the other limitation. Two concepts, i.e., interpolation and extrapolation, are borrowed from the statistics to explain the IFD model's generalization performance. When new test samples are within the range of the training set, the prediction stage is called interpolation, while when the testing data falls outside the training set, the prediction stage is called extrapolation. Numerous works have witnessed IFD models work well under the interpolation assumption, i.e., test samples are sampled from the same distribution as the historical training samples [13,14]. However, the interpolation hypothesis requires training and testing samples collected from identical working conditions and the same vibration transducer, which is hard to meet in industrial practice. It is practically significant to enhance the IFD model's extrapolation capability to promote the IFD methodology's application to real industries. Domain adaptation has shown the potential to improve the IFD model's generalization capability [1,15–17] and has been applied to address several kinds of transfer diagnosis tasks, *e.g.*, transfer diagnosis across working conditions [18–20], and across transducers at different positions [21,22]. However, the above research generally requires unlabeled or semi-supervised target domain data to align data distribution between source and target domain during the training phase. Although recent research [23–33] developed advanced deep learning-based diagnosis methods for domain generalization tasks across different working conditions, these methods are uninterpretable due to complex deep network architecture.

Benefiting from the L_2 stability and time-warping stability of wavelet scattering, the interpretable TScatNet model proposed by Liu and Shi holds excellent domain generalization diagnosis performance across working loads, operation speeds, and additive noise interference. However, TScatNet [6] and Ref. [23–32] generally focus on domain generalization study across working conditions and do not concern with domain generalization tasks across different transmission paths. In fact, it is of practical significance to develop domain generalization diagnosis methods across different transmissions. In the security-sensitive industry scenario, multiple transducers are generally mounted to monitor the machine's health condition. Once one transducer malfunctions, the adjacent transducer could substitute the malfunctioned transducer immediately, improving the reliability margin of the health monitoring system. In the cost-sensitive scenario, *e.g.*, the tasks of detecting foreign objects on the escalator guide, the domain generalization diagnosis method could detect the abnormal state at different locations with fewer transducers, saving the transducer cost.

The current work improves TScatNet into a normalized wavelet scattering convolutional network(NTScatNet) to enhance the IFD model's generalization capability across transmission paths and gives a detailed interpretation of each layer of NTScatNet. The contributions of this article are summarized below.

(1) An interpretable normalized wavelet scattering convolutional network is developed to address domain generalization diagnosis tasks across different transmissions.
(2) The physical interpretation of each layer of NTScatNet is described in detail. Specifically, the Conv1 layer conducts a multi-scale Hilbert demodulation operation on the input signals and outputs a wavelet scalogram. The Conv2 layer analysis the cyclic frequency of the demodulated signals in a constant-Q way and outputs the "scalogram of scalogram". Besides, the pooling layer and the global average pooling layer output the wavelet scattering features and global averaged wavelet scattering features, respectively, characterizing multi-scale cyclostationary information of the fault signals.
(3) This work theoretically illustrates that the normalized scattering features output by the scattering normalization layer is invariant to a linear time-invariant system and experimentally verifies that NTScatNet holds the domain generalization diagnosis capability across different transmission paths.

The rest of this article is organized as follows. Section 2 reviews a standard CNN architecture applied to intelligent fault diagnosis. Section 3 develops a novel NTScatNet model, describes the operation and physical meaning of each layer of NTScatNet in detail, and theoretically illustrates the normalized wavelet scattering feature's invariance to different transmission paths. Section 4 experimentally verifies NTScatNet's domain generalization ability across transmission paths through the transfer diagnosis tasks across transducers and the tasks of detecting foreign objects on the escalator guide rail. Section 5 concludes the current article.

2. A standard CNN architecture applied to intelligent fault diagnosis

The convolutional neural network(CNN) is a special deep feed-forward neural network with a local connection and weight-sharing properties. Fig. 1 shows a standard CNN architecture applied to IFD tasks, which contains an input layer, several convolutional layers, pooling layers, two fully connected layers, and a Softmax output layer. The operation of each layer of CNN is described briefly below.

(1) Input Layer: The input layer receives a time-domain vibration signal $\boldsymbol{x}$ of length L.

(2) Convolutional layer: The convolutional layer(Conv layer) slides a group of learnable convolution kernels on the original input signal (or the intermediate feature map) to extract a new set of feature maps through the convolution operation. The convolution process can be expressed as:

$$\boldsymbol{x}_m^l = \mathrm{ReLU}\Big(\sum_n \boldsymbol{w}_{m,n}^l * \boldsymbol{x}_n^{l-1} + \boldsymbol{b}_m^l\Big), \tag{1}$$

where $\boldsymbol{x}_n^{l-1}$ is nth feature input of the lth layer, $\boldsymbol{w}_{m,n}^l$ is the weights of the nth channel of the mth convolution kernel in the lth layer, $\boldsymbol{b}_m^l$ is the bias vector of the mth kernel, $\boldsymbol{x}_m^l$ is the mth feature vector output by the lth convolutional layer, and $\mathrm{ReLU}(\cdot)$ is the ReLU activation function.

(3) Pooling layer: The pooling layer conducts dimensionality reduction operation on the feature map output by the Conv layer. The outputs of the pooling layer are features with local time-shift invariance and small deformation stability. The max pooling and the average pooling are two commonly used pooling operations. The dimensionality reduction with max pooling could be expressed as:

$$\boldsymbol{x}_m^l = \max\{\boldsymbol{x}_m^{l-1}, r\}, \tag{2}$$

where $\max\{\cdot\}$ is the maximum pooling function, $\boldsymbol{x}_m^{l-1}$ is the m feature map before pooling, $\boldsymbol{x}_m^l$ is the feature vector after pooling, and r represents the pooling parameters, *e.g.*, size of pooling and steps.

(4) Batch normalization: The batch normalization layer(BN) aims to reduce internal covariate shifting and speed up the convergence of CNN's training process. Let $B = \{(\boldsymbol{x}^i, y^i)\}_{i=1}^m$ be a batch of training data, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the mean value and standard deviation of B, respectively. Eq. (3) gives the operation in the BN layer:

$$\hat{\boldsymbol{x}}^i = \boldsymbol{\gamma}\Big(\frac{\boldsymbol{x}^i - \boldsymbol{\mu}}{\boldsymbol{\sigma}}\Big) + \boldsymbol{\beta} \tag{3}$$

where $\boldsymbol{x}^i$ and $\hat{\boldsymbol{x}}^i$ are the input and output of the BN layer, respectively, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are learnable statistics of the BN layer.

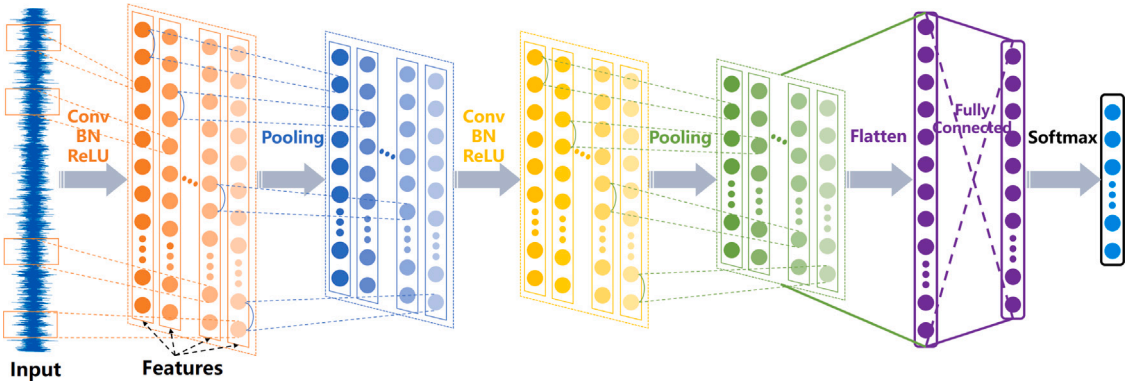


Fig. 1. A standard CNN applied to IFD.

(5) Fully connected layer: The operation in the lth fully connected(FC) layer could be expressed as:

$$\boldsymbol{x}^l = \sigma(\boldsymbol{W}^T \boldsymbol{x}^{l-1} + \boldsymbol{b}^l), \tag{4}$$

where $\boldsymbol{x}^{l-1}$ and $\boldsymbol{x}^l$ are the input and output of the lth FC layer, respectively; $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight matrix and bias vector of the lth FC layer; $\sigma(\cdot)$ is the activation function.

(6) Softmax Layer: The softmax layer maps a feature vector to probability values that the feature belongs to the cth category:

$$p_c(\boldsymbol{x}^l) = \frac{e^{\boldsymbol{w}_c^T \boldsymbol{x}^l + \boldsymbol{b}_c}}{\sum_{i=1}^{C} e^{\boldsymbol{w}_c^T \boldsymbol{x}^l + \boldsymbol{b}_c}}, \tag{5}$$

where $\boldsymbol{x}^l$ is the input of the softmax layer, $\boldsymbol{w}_c$ and $\boldsymbol{b}_c$ are the weights and the bias of the softmax layer, respectively, $p_c(\boldsymbol{x}^l)$ is the probability value that $\boldsymbol{x}^l$ belongs to the cth category.

3. Proposed interpretable CNN

This section develops an interpretable CNN, *i.e.*, the normalized wavelet scattering convolutional network(NTScatNet), to enhance standard CNN's interpretability. Section 3.1 presents NTScatNet's architecture and describes the operation in each layer of NTScatNet. The physical interpretation of each layer of NTScatNet is given in Section 3.2 by visualizing how NTScatNet operates a simulated bearing outer race fault signal. Section 3.3 illustrates NTScatNet's domain generalization capability across transmission paths through theoretical derivation. Section 3.4 summarizes the training and testing processes of the NTScatNet-based IFD method and analyzes the computational complexity.

3.1. Architecture of NTScatNet

As Fig. 2 presents, the proposed NTScatNet consists of an input layer, two convolutional layers, a pooling layer, a global averaging pooling layer(GAP), a scattering normalization layer(SN), and a softmax layer. Despite being similar to conventional CNN architecture, NTScatNet's distinct from standard CNN is that all convolutional kernels are pre-defined Morlet wavelets as opposed to learnable kernels, the activation function is a modulo operator, and the pooling layer takes a moving averaging operation. The operation of each layer of NTScatNet is described below.

(1) Input Layer: The input layer receives a raw time-domain signal $\boldsymbol{x}$ of length L, where L is a hyperparameter.

(2) Conv1 Layer: The convolution kernel of the Conv1 layer is a group of Morlet wavelets $\{\psi_{\lambda_1}\}_{\lambda_1}$. The Morlet wavelet filterbank's quality factor is Q_1, and the center frequency is $\lambda_1 = 2^{k/Q_1}$, $k \in \mathbb{Z}$, where Q_1 is a hyperparameter that denotes the number of wavelet filters per octave. Eq. (6) gives the operation in the Conv1 layer, which

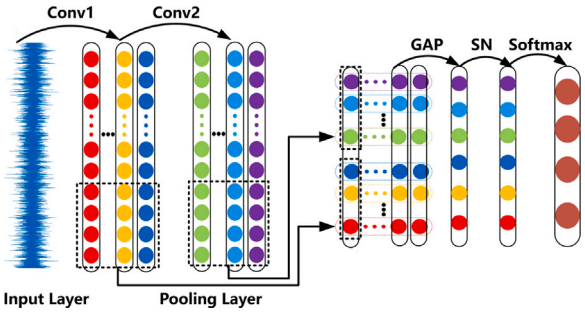


Fig. 2. Architecture of NTScatNet.

presents that the input signal $\boldsymbol{x}$ is first convolved with the Morlet wavelet filter and then activated by the modulo operator.

$$U_1(\boldsymbol{x}) = \text{Conv1}[\boldsymbol{x}] = |\boldsymbol{x} * \psi_{\lambda_1}| \tag{6}$$

(3) Conv2 Layer: The convolution kernel of the Conv2 layer is the other group of Morlet wavelets $\{\psi_{\lambda_2}\}_{\lambda_2}$, the quality factor and the center frequencies of which are Q_2 and $\lambda_2 = 2^{k/Q_2}$, $k \in \mathbb{Z}$, respectively. Conv2 layer performs Morlet wavelet convolution and modulo activation operations on each subband signals output by Conv1 layer.

$$U_2(\boldsymbol{x}) = \text{Conv2}[|\boldsymbol{x} * \psi_{\lambda_1}|] = ||\boldsymbol{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}| \tag{7}$$

(4) Pooling Layer: The pooling layer conducts moving average pooling operation on $U_1(\boldsymbol{x})$ and $U_2(\boldsymbol{x})$. The downsampling operator takes the scale filter ϕ, which is complementary to the Morlet wavelet filterbank. The recommended window width and the step size are $L/2$ and $L/8$, respectively. The output of the pooling layer is the first- and the second-order wavelet scattering features.

$$S_1(\boldsymbol{x}) = U_1(\boldsymbol{x}) * \phi = |\boldsymbol{x} * \psi_{\lambda_1}| * \phi \tag{8}$$

$$S_2(\boldsymbol{x}) = U_2(\boldsymbol{x}) * \phi = ||\boldsymbol{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi \tag{9}$$

(5) GAP Layer: The GAP layer performs global average pooling operation on the $S_1(\boldsymbol{x})$ and $S_2(\boldsymbol{x})$. The output of the GAP layer is the globally averaged wavelet scattering features.

$$\overline{S}_1(\boldsymbol{x}) = \text{GAP}(S_1(\boldsymbol{x})) \tag{10}$$

$$\overline{S}_2(\boldsymbol{x}) = \text{GAP}(S_2(\boldsymbol{x})) \tag{11}$$

(6) SN Layer: The SN layer conducts a scattering normalization operation on the global averaged wavelet scattering features and outputs the normalized wavelet scattering features.

$$\widetilde{S}_1(\boldsymbol{x}) = \overline{S}_1(\boldsymbol{x})/(|\boldsymbol{x}| * \phi) \tag{12}$$

Table 1
Information of NTScatNet.

Layer	Input	Output	Hyperparameters
Input layer	$\boldsymbol{x}$	$\boldsymbol{x}$	L
Conv1 layer	$\boldsymbol{x}$	$U_1(\boldsymbol{x})$	Q_1
Conv2 layer	$U_1(\boldsymbol{x})$	$U_2(\boldsymbol{x})$	Q_2
Pooling layer	$U_1(\boldsymbol{x}), U_2(\boldsymbol{x})$	$S(\boldsymbol{x}) = \{S_1(\boldsymbol{x}); S_2(\boldsymbol{x})\}$	$L/2, L/8$
GAP layer	$S(\boldsymbol{x})$	$\overline{S}(\boldsymbol{x}) = \{\overline{S}_1(\boldsymbol{x}); \overline{S}_2(\boldsymbol{x})\}$	–
SN layer	$\overline{S}(\boldsymbol{x})$	$\widetilde{S}(\boldsymbol{x}) = \{\widetilde{S}_1(\boldsymbol{x}); \widetilde{S}_2(\boldsymbol{x})\}$	–
Softmax layer	$\widetilde{S}(\boldsymbol{x})$	predicted label	C

$$\widetilde{S}_2(\boldsymbol{x}) = \overline{S}_2(\boldsymbol{x})/\overline{S}_1(\boldsymbol{x}) \tag{13}$$

The normalized wavelet scattering features should be flattened and cascaded before being fed to the softmax layer.

$$\widetilde{S}(\boldsymbol{x}) = \left\{\widetilde{S}_1(\boldsymbol{x}); \widetilde{S}_2(\boldsymbol{x})\right\} \tag{14}$$

(7) Softmax Layer: Let $D_S : \{\boldsymbol{x}_S^i, y_S^i\}_{i=1}^{N_S}$ the source domain dataset and $D_T : \{\boldsymbol{x}_T^j, y_T^j\}_{j=1}^{N_T}$ the target domain dataset. During the training phase, the Softmax layer maps the normalized wavelet scattering features of $\boldsymbol{x}_S^i$ to the probability values that $\boldsymbol{x}_S^i$ belongs to each fault category:

$$y_{\text{score}}^i = \frac{1}{\sum_{k=1}^{C} e^{(\boldsymbol{w}_k^T \widetilde{S}(\boldsymbol{x}_S^i)+b_k)}} \begin{bmatrix} e^{(\boldsymbol{w}_1^T \widetilde{S}(\boldsymbol{x}_S^i)+b_1)} \\ e^{(\boldsymbol{w}_2^T \widetilde{S}(\boldsymbol{x}_S^i)+b_2)} \\ \dots \\ e^{(\boldsymbol{w}_C^T \widetilde{S}(\boldsymbol{x}_S^i)+b_C)} \end{bmatrix}, \tag{15}$$

where $\boldsymbol{w}_k$ and b_k are the weight vector and bias value that corresponds to the kth category, respectively, and C is the total number of categories. The data loss of the source domain could be calculated according to Eq. (16), and the softmax layer's parameters could be optimized with the mini-batch gradient descent algorithm according to Eq. (17).

$$Loss = -\frac{1}{N_S}\left[\sum_{i=1}^{N_S}\sum_{k=1}^{C} I[y^i = k] \log \frac{e^{(\boldsymbol{w}_k^T \widetilde{S}(\boldsymbol{x}_S^i)+b_k)}}{\sum_{k=1}^{C} e^{(\boldsymbol{w}_k^T \widetilde{S}(\boldsymbol{x}_S^i)+b_k)}}\right] \tag{16}$$

$$W^* = \underset{W}{\operatorname{argmin}} Loss(W), \quad W \leftarrow W - \varepsilon \frac{\partial Loss}{\partial W} \tag{17}$$

During the testing phase, the trained softmax classifier predicts the label of the target domain sample according to Eq. (18).

$$y_{\text{pre}}^j = \underset{k}{\operatorname{argmax}} \frac{e^{(\boldsymbol{w}_k^{*T} \widetilde{S}(\boldsymbol{x}_T^j)+b_k^*)}}{\sum_{k=1}^{C} e^{(\boldsymbol{w}_k^{*T} \widetilde{S}(\boldsymbol{x}_T^j)+b_k^*)}} \tag{18}$$

The input, output, and hyperparameters of each layer of NTScatNet are summarized in Table 1.

3.2. Interpretation of each layer of NTScatNet

One limitation of the conventional CNN is that its feature extraction process is not transparent to diagnosis engineers due to the complex architecture. The proposed NTScatNet simplifies the standard CNN architecture and improves model interpretability. This subsection gives the physical interpretation of each layer of NTScatNet by visualizing how NTScatNet operates a simulated bearing outer race fault signal $x(t)$ presented in Fig. 3.

(1) Input Layer: The input layer receives the simulated bearing outer race fault signal $x(t)$.

(2) Conv1 Layer: Conv1 layer's convolution kernel is a family of Morlet wavelets, the frequency domain counterpart of which is a constant-Q bandpass filterbank. Fig. 4 presents the Morlet wavelet filterbank with $Q_1 = 4$. The Conv1 layer conducts the Morlet wavelet transform on $x(t)$ and decomposes $x(t)$ into P1 subband wavelet signals, where P1 is the number of Morlet wavelet filters of the filterbank shown in Fig. 4.

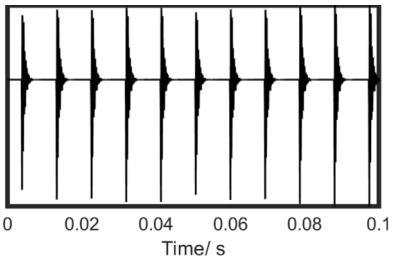


Fig. 3. The simulated bearing outer race fault signal $x(t)$.

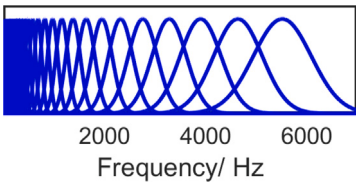


Fig. 4. Morlet wavelet filter bank with $Q_1 = 4$.

$$\begin{aligned} x(t) * \psi_{\lambda_1}(t) &= w_R(t, \lambda_1) + j w_I(t, \lambda_1) \\ &= w_R(t, \lambda_1) + j\mathcal{H}[w_R(t, \lambda_1)] \end{aligned} \tag{19}$$

In Eq. (19), $w_R(t, \lambda_1)$ and $w_I(t, \lambda_1)$ are the real and imaginary part of the wavelet coefficients, respectively, $\mathcal{H}[\cdot]$ is the Hilbert transform operator. The complex Morlet wavelet coefficients are then activated by the modulo operator, which extracts the Hilbert envelope at each wavelet scale.

$$\begin{aligned} U_1 x(t, \lambda_1) &= |x(t) * \psi_{\lambda_1}(t)| \\ &= \sqrt{w_R(t, \lambda_1)^2 + H[w_R(t, \lambda_1)^2]} \end{aligned} \tag{20}$$

This article terms the Conv1 layer "Demodulation Filterbank" since it performs multiscale Hilbert demodulation on the input signal. Fig. 6 presents the multiscale demodulated envelope signals output by the Conv1 layer.

(3) Conv2 Layer: Conv2 layer is the other constant-Q bandpass filterbank, which contains P2 Morlet wavelets. Fig. 7 presents a Morlet wavelet filterbank with $Q_2 = 3$. Conv2 layer conducts Morlet wavelet transform on $U_1 x(t, \lambda_1)$ and then activates the output with the modulo operator.

$$U_2 x(t, \lambda_1, \lambda_2) = \| x(t) * \psi_{\lambda_1} | * \psi_{\lambda_2} | \tag{21}$$

As Fig. 5 presents, the Conv1 layer decomposes $x(t)$ into a wavelet scalogram $U_1 x(t, \lambda_1)$, while the Conv2 layer transforms each demodulated Hilbert envelope into a new scalogram $U_2 x(t, \lambda_1, \lambda_2)$. This article terms $U_2 x(t, \lambda_1, \lambda_2)$ "Scalogram of Scalogram" and the Conv2 layer "Analysis Filterbank" since it analyzes the cyclic frequency of each demodulated envelope with a constant bandwidth filterbank.

(4) Pooling Layer: As Fig. 5 shows, the pooling layer downsamples the wavelet scalogram and "Scalogram of Scalogram" with a moving average operator ϕ to obtain local time shift invariant representation. The outputs of the pooling layer are the first- and second-order wavelet scattering feature maps.

$$S_1 x(t, \lambda_1) = U_1 x(t, \lambda_1) * \phi(t) \tag{22}$$

$$S_2 x(t, \lambda_1, \lambda_2) = U_2 x(t, \lambda_1, \lambda_2) * \phi(t) \tag{23}$$

For example, let $L/2$ the window length of $\phi(t)$, and $L/8$ the step size of the moving average. Then the size of $S_1 x(t, \lambda_1)$ and $S_2 x(t, \lambda_1, \lambda_2)$ are $5 \times P_1$ and $5 \times P_1 \times P_2$, respectively.

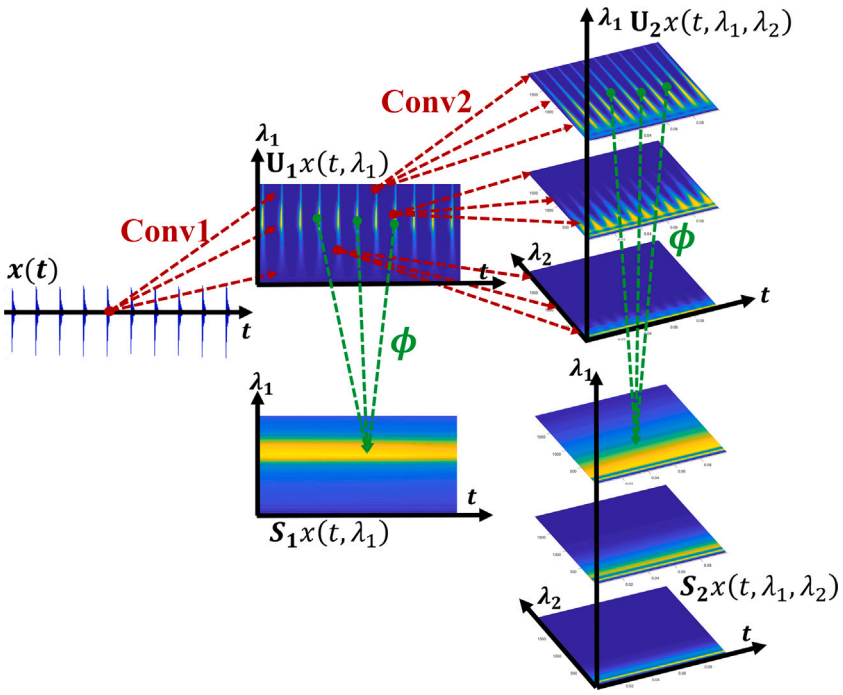


Fig. 5. Interpreting the feature extractor of NTScatNet as Demodulation-Analysis Filterbanks.

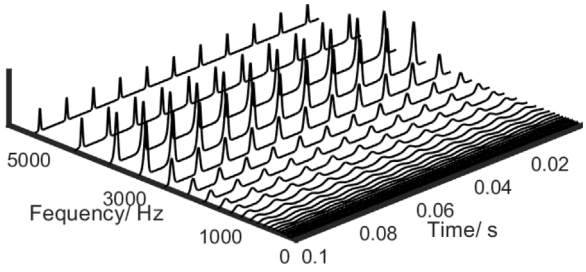


Fig. 6. Multiscale demodulated envelope signals.

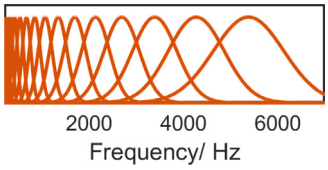


Fig. 7. Morlet wavelet filter bank with $Q_2 = 3$.

(5) GAP Layer: The GAP layer conducts global average pooling on $S_1x(t,\lambda_1)$ and $S_2x(t,\lambda_1,\lambda_2)$ to obtain global time-shift invariant representation.

$$\overline{S}_1x(\lambda_1) = \mathrm{GAP}(S_1x(t,\lambda_1)) \tag{24}$$

$$\overline{S}_2(\lambda_1,\lambda_2) = \mathrm{GAP}(S_2x(t,\lambda_1,\lambda_2)) \tag{25}$$

Fig. 8(a) shows $x(t)$'s Fourier spectrum, and Fig. 8(b)–8(d) present the globally averaged first-order wavelet scattering feature maps of $x(t)$ with $Q_1 = 4, 8, 16$, respectively. Fig. 8 illustrates that the globally averaged first-order wavelet scattering feature map could be considered an approximation of the Fourier spectrum, whose frequency resolution improves with the increase of the quality factor Q_1. In other words, $\overline{S}_1x(\lambda_1)$ captures the formant of $x(t)$ and the resonance of the monitored machine structure.

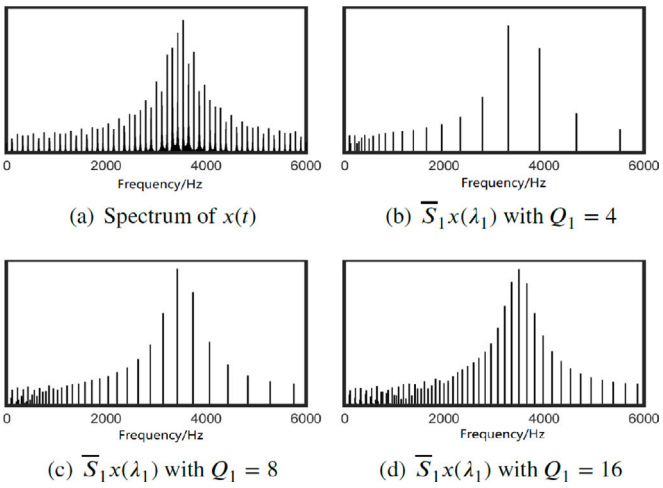


Fig. 8. The globally averaged first-order wavelet scattering feature maps.

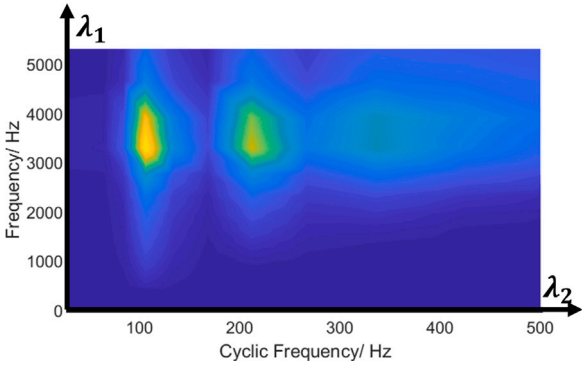


Fig. 9. Globally averaged second-order wavelet scattering feature.

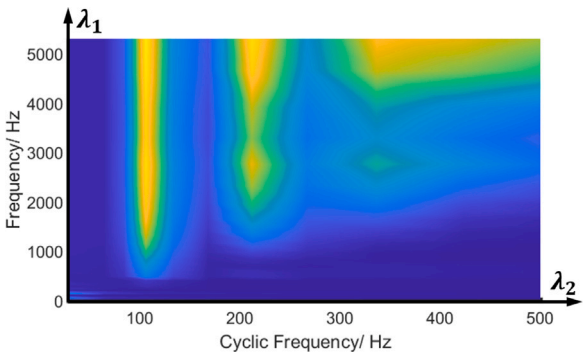


Fig. 10. Normalized second-order wavelet scattering feature.

The globally averaged second-order wavelet scattering feature map, *i.e.*, $\overline{S}_2(\lambda_1, \lambda_2)$ is presented in Fig. 9. As Fig. 9 shows, $\overline{S}_2(\lambda_1, \lambda_2)$ is a bi-spectral map, where λ_1 is the "spectral frequency" axis, characterizing the resonance of the monitored machine, while λ_2 is the "cyclic frequency" axis, indicating $x(t)$'s cyclostationarity at the λ_1 scale. In other words, the globally averaged second-order wavelet scattering features $\overline{S}_2(\lambda_1, \lambda_2)$ characterizes the multiscale cyclostationarity of $x(t)$.

(6) SN Layer: The SN layer performs scattering normalization on the globally averaged wavelet scattering features to reduce the influence of the energy at different wavelet scales.

$$\widetilde{S}_1 x(\lambda_1) = \overline{S}_1 x(\lambda_1)/(|x(t)| * \phi(t)) \tag{26}$$

$$\widetilde{S}_2 x(\lambda_1, \lambda_2) = \overline{S}_2 x(\lambda_1, \lambda_2)/\overline{S}_1 x(\lambda_1) \tag{27}$$

The normalized second-order wavelet scattering feature of $x(t)$ is shown in Fig. 10, which illustrates that the normalized wavelet scattering-based cyclostationarity representation becomes insensitive to the energy at different wavelet scales.

(7) Softmax Layer: The softmax layer maps the normalized wavelet scattering features to the health condition category of the monitored machine structure. The softmax layer is interpretable since it is just a simple linear classifier. So far, the specific physical explanation of each layer of NTScatNet has been detailed, which indicates that the proposed NTScatNet is a fully interpretable convolutional neural network.

3.3. NtScatNet's generalization across transmission paths

This subsection verifies NTScatNet's domain generalization capability across transmission paths through a brief theory derivation. As is well known to the vibration-based condition monitoring research community, an impact excitation will be produced when a bearing defect strikes another surface, and the impact will excite one or more resonances of the monitored machine structure. The impulse response oscillates at the natural frequencies and decays rapidly due to structural damping, characterizing the characteristics of the transmission path between the impact point and the transducer. In practice, multiple transducers are generally mounted to monitor the machine's health condition. As shown in Fig. 11, the impulse response excited by the bearing defect is observed by two transducers at two positions, where $x(t)$ is the impact signal, $h_1(t)$ characterizes the transmission path between the impact point and transducer1, and $h_2(t)$ characterizes the transmission path between impact point and transducer2. Despite corresponding to the same bearing fault, significant data distribution discrepancy exists between $x(t) * h_1(t)$ and $x(t) * h_2(t)$ due to different transmission path characteristics.

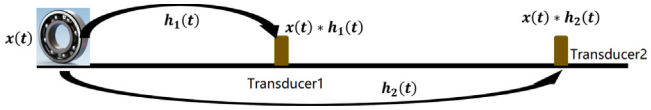


Fig. 11. Transmission path variability.

3.3.1. Linear time-invariant system's response to morlet wavelet excitation

Let $h(t)$ the impulse response function of a linear time-invariant system, and $H(\omega)$ is the Fourier transform of $h(t)$, *i.e.*, the linear time-invariant system's frequency response function. As Eq. (28) presents, $h(t)$'s response to a complex exponential signal is a complex exponential signal with gain $H(\omega)$.

$$h(t) * e^{i\omega t} = H(\omega)e^{i\omega t} \tag{28}$$

In other words, the characteristic function of a time-invariant system is a group of complex exponential functions $e^{i\omega t}$. Morlet wavelet $\psi_\lambda(t)$ is a complex exponential signal modulated by Gaussian window, and the frequency counterpart of $\psi_\lambda(t)$ is a bandpass Gaussian filter $\Psi_\lambda(\omega)$. For a real mechanical system, the amplitude frequency response $H(\omega)$ of the system generally vary gently with frequency ω. Therefore, $H(\omega)$ could be approximated by a constant $H(\lambda)$ in a narrow wavelet frequency band support $[\lambda - \lambda/Q, \lambda + \lambda/Q]$:

$$H(\omega) \approx H(\lambda), \tag{29}$$

where Q and λ are the quality factor and center frequency of the Morlet wavelet filter, respectively. The linear time-invariant system's response to a Morlet wavelet excitation could be approximately derived as:

$$\begin{aligned} h(t) * \psi_\lambda(t) &= F^{-1}[H(\omega)\Psi_\lambda(\omega)] \approx F^{-1}[H(\lambda)\Psi_\lambda(\omega)] \\ &= H(\lambda)F^{-1}[\Psi_\lambda(\omega)] = H(\lambda)\psi_\lambda(t) \end{aligned} \tag{30}$$

Eq. (30) implies that the linear time-invariant system's response to a Morlet wavelet is a Morlet wavelet with gain $H(\lambda)$.

3.3.2. Wavelet scattering features of signals observed by transducers at different positions

According to Eqs. (8), (22), and (30), the first-order wavelet scattering features of $x(t) * h_1(t)$ could be derived as:

$$\begin{aligned} S_1[x(t) * h_1(t)](t, \lambda_1) &= |x(t) * h_1(t) * \psi_{\lambda_1}(t)| * \phi(t) \\ &\approx |x(t) * H_1(\lambda_1)\psi_{\lambda_1}(t)| * \phi(t) \\ &= H_1(\lambda_1)|x(t) * \psi_{\lambda_1}(t)| * \phi(t) \\ &= H_1(\lambda_1)S_1[x(t)](t, \lambda_1) \end{aligned} \tag{31}$$

The second-order wavelet scattering features of $x(t) * h_1(t)$ could be represented as Eq. (32) according to Eqs. (9), (23), and (30).

$$\begin{aligned} S_2[x(t) * h_1(t)](t, \lambda_1, \lambda_2) &= \||x(t) * h_1(t) * \psi_{\lambda_1}(t)| * \psi_{\lambda_2}(t)| * \phi(t) \\ &\approx \||x(t) * H_1(\lambda_1)\psi_{\lambda_1}(t)| * \psi_{\lambda_2(t)}| * \phi(t) \\ &= H_1(\lambda_1)\||x(t) * \psi_{\lambda_1}(t)| * \psi_{\lambda_2}(t)| * \phi(t) \\ &= H_1(\lambda_1)S_2[x(t)](t, \lambda_1, \lambda_2) \end{aligned} \tag{32}$$

Similarly, the first-order and the second-order wavelet scattering features of $x(t) * h_2(t)$ could be derived as Eqs. (33) and (34), respectively.

$$S_1[x(t) * h_2(t)](t, \lambda_1) = H_2(\lambda_1)S_1[x(t)](t, \lambda_1) \tag{33}$$

$$S_2[x(t) * h_2(t)](t, \lambda_1, \lambda_2) = H_2(\lambda_1)S_2[x(t)](t, \lambda_1, \lambda_2) \tag{34}$$

Since $H_1(\lambda_1) \neq H_2(\lambda_1)$, the first- and the second-order scattering feature are not invariant to transmission path.

$$\begin{aligned} S_1[x(t) * h_1(t)](t, \lambda_1) &\neq S_1[x(t) * h_2(t)](t, \lambda_1) \\ S_2[x(t) * h_1(t)](t, \lambda_1, \lambda_2) &\neq S_2[x(t) * h_2(t)](t, \lambda_1, \lambda_2) \end{aligned} \tag{35}$$

3.3.3. Normalized wavelet scattering features of signals observed by transducers at different positions

The normalized second-order scattering feature of $x(t) * h_1(t)$ could be presented as Eq. (36).

$$\begin{aligned}\widetilde{S}_2[x(t) * h_1(t)](t,\lambda_1,\lambda_2) &= \frac{S_2[x(t) * h_1(t)](t,\lambda_1,\lambda_2)}{S_1[x(t) * h_1(t)](t,\lambda_1)} \\ = \frac{H_1(\lambda_1)S_2[x(t)](t,\lambda_1,\lambda_2)}{H_1(\lambda_1)S_1[x(t)](t,\lambda_1)} &= \frac{S_2[x(t)](t,\lambda_1,\lambda_2)}{S_1[x(t)](t,\lambda_1)}\end{aligned} \tag{36}$$

Similarly, the normalized second-order scattering feature of $x(t) * h_2(t)$ could be formulated as Eq. (37).

$$\begin{aligned}\widetilde{S}_2[x(t) * h_2(t)](t,\lambda_1,\lambda_2) &= \frac{S_2[x(t) * h_2(t)](t,\lambda_1,\lambda_2)}{S_1[x(t) * h_2(t)](t,\lambda_1)} \\ = \frac{H_2(\lambda_1)S_2[x(t)](t,\lambda_1,\lambda_2)}{H_2(\lambda_1)S_1[x(t)](t,\lambda_1)} &= \frac{S_2[x(t)](t,\lambda_1,\lambda_2)}{S_1[x(t)](t,\lambda_1)}\end{aligned} \tag{37}$$

Comparing Eqs. (36) and (37) reveals that the normalized second-order scattering feature is invariant to a linear time-invariant system. In other words, NTScatNet could extract features invariant to transmission path variability and thus holds the domain generalization capability across different transmission paths.

3.4. NtScatNet-based IFD algorithm

3.4.1. Training and testing process of NTScatNet

The training and testing processes of NTScatNet are summarized in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: The training process of NTScatNet

Input: $\{\boldsymbol{x}_S^i, y_S^i\}_{i=1}^{N_S}$; L, Q_1, Q_2; Optimizer; Training epochs; Learning rate

Output: The trained NTScatNet model

Initialization: Initializating NTScatNet;
Calculate $\overline{S}(\boldsymbol{x}_S^i)$ of the source domain according to Eqs. (6)–(9);
Calculate the normalized wavelet scattering features, *i.e.*, $\widetilde{S}(\boldsymbol{x}_S^i)$ according Eqs. (10)–(14);
for *Training epochs* < *Pre-defined epochs* **do**
  Calculate the data loss of the source domain according to Eqs. (15) and (16);
  Parameters optimization according to (17);
end
Output the trained NTScatNet model;

Algorithm 2: The testing process of NTScatNet

Input: Trained NTScatNet; $\{\boldsymbol{x}_T^j\}_{j=1}^{N_T}$;

Output: The predicted label: y_{pre}^j;

Calculate $\overline{S}(\boldsymbol{x}_T^j)$ of the target domain samples according to Eqs. (6)–(9);
Calculate the normalized wavelet scattering features $\widetilde{S}(\boldsymbol{x}_T^j)$ according Eqs. (10)–(14);
Label prediction according to (18);
Output the predicted label y_{pre}^j;

3.4.2. Computational complexity analysis

Let $\boldsymbol{x}$ the input sample, L the dimension of $\boldsymbol{x}$, $N = L/2$ the length of moving averaging widow, and $s = L/8$ the size of steps. The number of time frame is $(L - N)/s + 1 = 5$. Besides, let Q_1, Q_2 are the number of wavelets per octave of the first filterbank and second filterbank, respectively and Fs the sampling frequency of $\boldsymbol{x}$. For each moving averaging window, the number of first-order wavelets ψ_{λ_1} is about $Q_1 log_2 N$, and there are about $Q_1 log_2 N$ first-order wavelet scattering coefficients to be computed. For each first-order wavelet scale, the number of second-order wavelets ψ_{λ_2} is about $Q_2 log_2 N$, while the number of non-negligible second-order wavelet scattering coefficients is $Q_2 log_2 N/2$ since the second-order wavelet scattering coefficients with $\lambda_2 > \lambda_1$ are trivial. In total, the number of wavelet scattering coefficients to be computed is: $5(Q_1 log_2 N + Q_1 Q_2 (log_2 N)^2)$. In order to compute the wavelet scattering coefficients, the modulus wavelet coefficients should be first calculated and then averaged with ϕ. The computation complexity of modulus wavelet coefficients is the order of $O(N log_2 N)$ with a fast Fourier transform(FFT) algorithm. The computation complexity of each wavelet scattering coefficient is also $O(N log_2 N)$. The number of operations for computing all non-negligible wavelet scattering coefficients is:

$$\text{Complexity} = O\left(5Q_1 N(log_2 N)^2 + 5\frac{Q_1 Q_2 N(log_2 N)^3}{2}\right) \tag{38}$$

We reasonably assume that the rotation speed range of the concerned machine is from 60 rpm/min to 6000rpm/min, so the period range is from 0.01 s to 1 s. Besides, let Fs = 12000 Hz, $Q_1 = 16$, $Q_2 = 4$, each time frame of $\boldsymbol{x}$ contains $4 - 6$ cycles to calculate its scattering coefficients, and the CPU calculates $5 * 10^9$ complex addition operation per second. Fig. 12 plots the calculation time as the function of the dimension of the input signal. As Fig. 12 presents, the wavelet scattering coefficients of a signal with a dimension below 144 000 could be calculated within 0.15 s, demonstrating NTScatNet's computational feasibility from an industrial perspective.

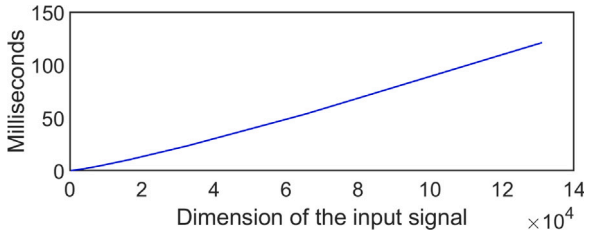


Fig. 12. NTScatNet's calculation time under various dimensions of the input signal.

4. Experimental verification

4.1. Introduction to datasets

CWRU datasets are provided by CWRU bearing data center. Fig. 13 presents the experimental platform where the datasets are collected. Although the CWRU dataset is very popular in machine health monitoring, most research concerns motor drive-end bearing's health condition using the acceleration signals measured by the drive-end transducers. This article terms this popular dataset CWRU-DD dataset for convenience. In fact, the bearing data center provided the other three datasets, *i.e.*, CWRU-DF, CWRU-FD, and CWRU-FD datasets.

CWRU-DD dataset concerns the health condition of motor drive-end bearing, the specification of which is SKF-6205. Four bearing health conditions, i.e., Normal(N), ball fault(BF), inner race fault(IF), and outer race fault(OF), are considered. Each bearing fault is machined with a diameter of 7 mils using electrical discharge machining. The transducer is mounted on the drive-end bearing house to monitor the drive-end bearing's health condition. The vibration signals are collected under four working loads(0hp, 1hp, 2hp, and 3hp) at the sampling frequency of 12 kHz. Let $x_{\text{D}}(t)$ the signal excited by drive-end bearing, $h_{\text{DD}}(t)$ the transmission path from the drive-end bearing to the drive-end transducer. The signals of CWRU-DD dataset could be presented as $x_{\text{D}}(t) * h_{\text{DD}}(t)$. In total, $W_c \times C = 4 \times 4 = 16$ time series are recorded, where W_c is the number of working conditions and C is the number of health conditions. Each recorded signal is segmented into 100 samples of length L with an overlapping trick. The CWRU-DD dataset totally contains 4 sub-datasets,i.e., DD-0hp, DD-1hp, DD-2hp, and DD-3hp, where each sub-dataset contains $100 \times 4 = 400$ samples.

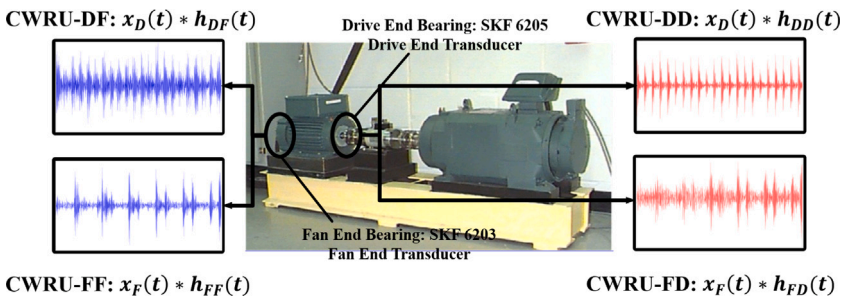


Fig. 13. CWRU experimental platform.

Table 2
Information of CWRU-DD, CWRU-DF, CWRU-FF, and CWRU-FD datasets.

Datasets	CWRU-DD dataset	CWRU-DF dataset	CWRU-FF dataset	CWRU-FD dataset
Bearing position	Motor drive-end	Motor drive-end	Motor fan-end	Motor fan-end
Specification	SKF 6205	SKF 6205	SKF 6203	SKF 6203
Transducer position	Motor drive-end	Motor fan-end	Motor fan-end	Motor drive-end
Working loads	0/1/2/3 hp	0/1/2/3 hp	0/1/2/3 hp	0/1/2/3 hp
Fault categories	N/BF/IF/OF	N/BF/IF/OF	N/BF/IF/OF	N/BF/IF/OF
Sample dimension	L	L	L	L
Dataset size	$4 \times 4 \times 100$	$4 \times 4 \times 100$	$4 \times 4 \times 100$	$4 \times 4 \times 100$
Signal symbol	$x_D(t) * h_{DD}(t)$	$x_D(t) * h_{DF}(t)$	$x_F(t) * h_{FF}(t)$	$x_F(t) * h_{FD}(t)$
Sub-datasets	DD-0hp/DD-1hp/DD-2hp/DD-3hp	DF-0hp/DF-1hp/DF-2hp/DF-3hp	FF-0hp/FF-1hp/FF-2hp/FF-3hp	FD-0hp/FD-1hp/FD-2hp/FD-3hp

CWRU-DF dataset concerns drive-end bearing's health condition using signals observed by the fan-end transducer. The vibration signals are collected under four working loads(0hp, 1hp, 2hp, and 3hp) at a sampling frequency of 12 kHz. Let $h_{DF}(t)$ the transmission path from the drive-end bearing to the fan-end transducer. The signals of CWRU-DF dataset could be presented as $x_D(t) * h_{DF}(t)$. Each recorded signal is segmented into 100 samples of length L with an overlapping trick. The CWRU-DF dataset totally contains 4 sub-datasets, i.e., DF-0hp, DF-1hp, DF-2hp, and DF-3hp, where each sub-dataset contains $100 \times 4 = 400$ samples.

CWRU-FF dataset concerns the health condition of motor fan-end bearing, the specification of which is SKF-6203. Four bearing health conditions, i.e., Normal(N), ball fault(BF), inner race fault(IF), and outer race fault(OF), are considered. Each bearing fault is machined with a diameter of 7 mils using electrical discharge machining. The transducer is mounted on the fan-end bearing house to monitor the fan-end bearing's health condition. The vibration signals are collected under four working loads(0hp, 1hp, 2hp, and 3hp) at the sampling frequency of 12 kHz. Let $x_F(t)$ the signal excited by fan-end bearing, $h_{FF}(t)$ the transmission path from the fan-end bearing to the fan-end transducer. The signals of CWRU-FF dataset could be presented as $x_F(t) * h_{FF}(t)$. Each recorded signal is segmented into 100 samples of length L with an overlapping trick. The CWRU-FF dataset totally contains 4 sub-datasets,i.e., FF-0hp, FF-1hp, FF-2hp, and FF-3hp, where each sub-dataset contains $100 \times 4 = 400$ samples.

CWRU-FD dataset concerns fan-end bearing's health condition using signals observed by the drive-end transducer. The vibration signals are collected under four working loads(0hp, 1hp, 2hp, and 3hp) at a sampling frequency of 12 kHz. Let $h_{FD}(t)$ the transmission path from the fan-end bearing to the drive-end transducer. The signals of CWRU-FD dataset could be presented as $x_F(t) * h_{FD}(t)$. Each recorded signal is segmented into 100 samples of length L with an overlapping trick. The CWRU-FD dataset totally contains 4 sub-datasets, i.e., FD-0hp, FD-1hp, FD-2hp, and FD-3hp, where each sub-dataset contains $100 \times 4 = 400$ samples. Specific information about the above four bearing datasets are summarized in Table 2.

4.2. Feature visualization

This subsection aims to verify that NTScatNet could signification reduce data discrepancy due to transmission path variability by visualizing the actual shape of normalized scattering features.

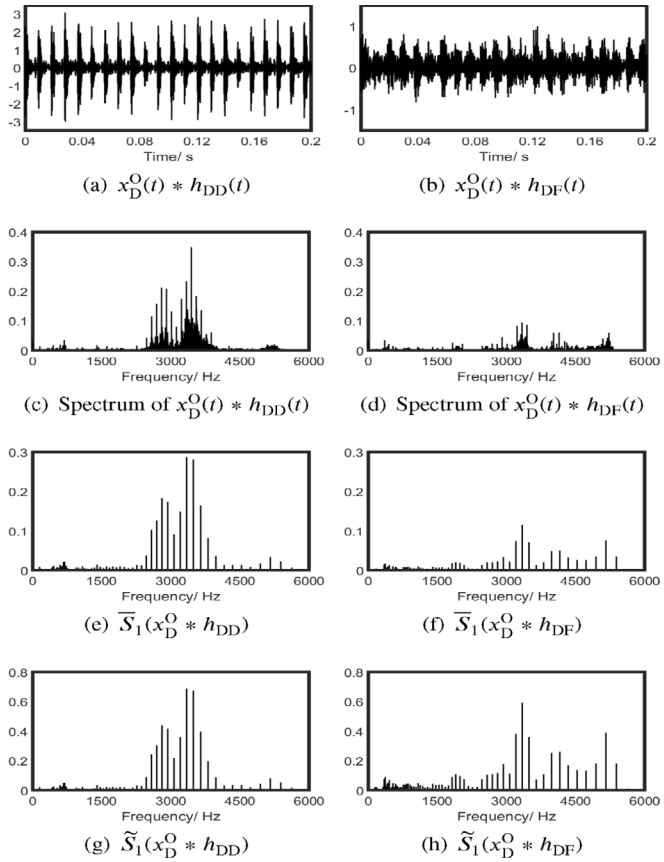
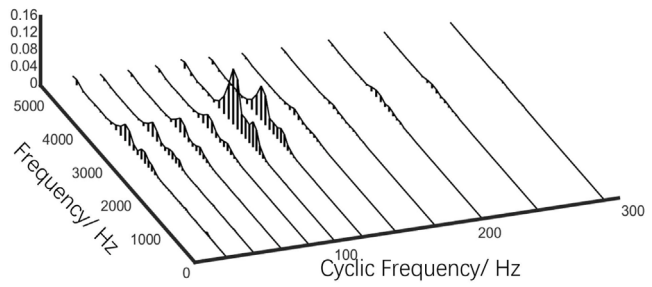


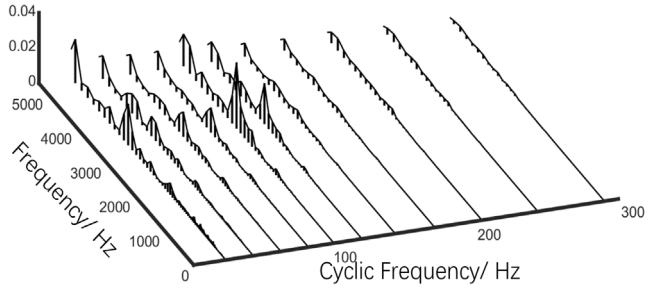
Fig. 14. Visualization of drive-end bearing signal's features.

4.2.1. Feature visualization of drive-end bearing signal

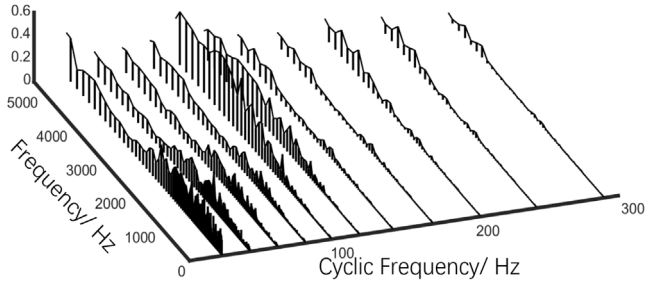
The first case will visualize features of the drive-end bearing with an outer race defect. The experimental working load is 0hp, and two transducers are separately mounted on the drive-end bearing house



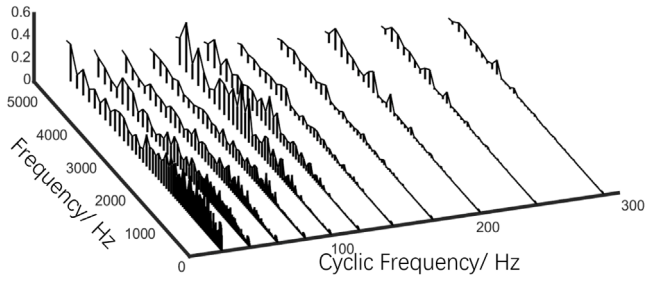
(a) $\overline{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DD}})$



(b) $\overline{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DF}})$



(c) $\widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DD}})$



(d) $\widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DF}})$

Fig. 15. Visualization of drive-end bearing signal's features.

and the fan-end bearing house to monitor the bearing's health condition. Fig. 14(a)–14(b) present the time domain signals observed by the drive-end transducer and the fan-end transducer, respectively, *i.e.*, $x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DD}}(t)$ and $x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DF}}(t)$. The amplitude of $x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DD}}(t)$ is significantly larger than that of $x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DF}}(t)$ since the drive-end transducer is near the drive-end fault bearing while the fan-end transducer is remote from the drive-end bearing. Fig. 14(c)–14(d) show the amplitude spectrum of $x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DD}}(t)$ and $x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DF}}(t)$, respectively. The amplitude spectrum of $x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DD}}(t)$ and $x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DF}}(t)$ present different formants and amplitude, illustrating that Fourier transform fails to remove the discrepancy due to transmission path variability. As illustrated in Section 3.2, the globally averaged first-order wavelet scattering feature approximates the Fourier spectrum. Therefore, it fails to remove the discrepancy due to transmission path variability, as Fig. 14(e)–14(f) present. By contrast, the normalized first-order scattering feature effectively reduced amplitude discrepancy, as shown in Fig. 14(g)–14(h).

As Section 3.2 implies, the globally averaged second-order wavelet scattering feature well characterizes the multiscale cyclostationarity of the fault signals. However, the cyclostationarity representation based on the globally averaged second-order wavelet scattering feature is sensitive to the energy of different wavelet scales, as Figs. 15(a) and 15(b) present. By contrast, the cyclostationarity representation based on the normalized second-order wavelet scattering feature become insensitive to the energy of different wavelet scales, as Figs. 15(c) and 15(d) illustrate. Moreover, Eqs. (39) and (40) calculated the normalized Euclidean distance as a metric to quantify transmission path variability in the scattering space and the normalized scattering space, respectively.

$$\frac{\|\overline{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DD}}) - \overline{S}_2(x_{\mathrm{D}}^{\mathrm{O}}(t) * h_{\mathrm{DF}})\|_2}{\|\overline{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DD}})\|_2 + \|\overline{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DF}})\|_2} = 0.56 \tag{39}$$

$$\frac{\|\widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DD}}) - \widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DF}})\|_2}{\|\widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DD}})\|_2 + \|\widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{O}} * h_{\mathrm{DF}})\|_2} = 0.22 \tag{40}$$

Comparing Eqs. (39) and (40) reveals that the normalized second-order wavelet scattering feature becomes insensitive to transmission path variability.

4.2.2. *Visualization of fan-end bearing signal's features*

The second case will visualize features of the fan-end bearing with an inner race defect. The experimental working load is 3hp, and two transducers are separately mounted on the drive-end bearing house and the fan-end bearing house to monitor the bearing's health condition.

Fig. 16(a)–16(b) present the time domain signals observed by the drive-end transducer and the fan-end transducer, respectively, from which one can find that the amplitude of $x_{\mathrm{F}}^{\mathrm{I}}(t) * h_{\mathrm{FF}}(t)$ is significantly larger than that of $x_{\mathrm{F}}^{\mathrm{I}}(t) * h_{\mathrm{FD}}(t)$. The amplitude spectrum of $x_{\mathrm{F}}^{\mathrm{I}}(t) * h_{\mathrm{FF}}(t)$ and $x_{\mathrm{F}}^{\mathrm{I}}(t) * h_{\mathrm{FD}}(t)$ are shown in Fig. 16(c) and Fig. 16(d), respectively. Besides, Fig. 16(e)–16(f) present the globally averaged first-order wavelet scattering features. However, both the Fourier spectrum and the globally averaged wavelet scattering features present significant discrepancies in amplitudes and formants due to different transmission path. In comparison, the normalized first-order scattering feature effectively reduced amplitude discrepancy, as shown in Fig. 16(g)–16(h).

Figs. 17(a) and 17(b) present $\overline{S}_2(x_{\mathrm{F}}^{\mathrm{I}} * h_{\mathrm{FF}})$ and $\overline{S}_2(x_{\mathrm{F}}^{\mathrm{I}} * h_{\mathrm{FD}})$, respectively, from which one can find a significant discrepancy in formant and amplitudes. By contrast, the cyclostationarity representation based on the normalized second-order wavelet scattering feature becomes insensitive to different transmission paths, as illustrated in Figs. 17(c) and 17(d). In addition, Eqs. (41) and (42) take the normalized Euclidean distance as a metric to quantify transmission path variability in the scattering space and the normalized scattering space.

$$\frac{\|\overline{S}_2(x_{\mathrm{F}}^{\mathrm{I}} * h_{\mathrm{FF}}) - \overline{S}_2(x_{\mathrm{F}}^{\mathrm{I}} * h_{\mathrm{FD}})\|_2}{\|\overline{S}_2(x_{\mathrm{F}}^{\mathrm{I}} * h_{\mathrm{FF}})\|_2 + \|\overline{S}_2(x_{\mathrm{F}}^{\mathrm{I}} * h_{\mathrm{FD}})\|_2} = 0.58 \tag{41}$$

$$\frac{\|\widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{I}} * h_{\mathrm{DD}}) - \widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{I}} * h_{\mathrm{DF}})\|_2}{\|\widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{I}} * h_{\mathrm{DD}})\|_2 + \|\widetilde{S}_2(x_{\mathrm{D}}^{\mathrm{I}} * h_{\mathrm{DF}})\|_2} = 0.21 \tag{42}$$

Comparing Eqs. (41) and (42) further verifies that the normalized second-order wavelet scattering feature becomes less sensitive to transmission path variability.

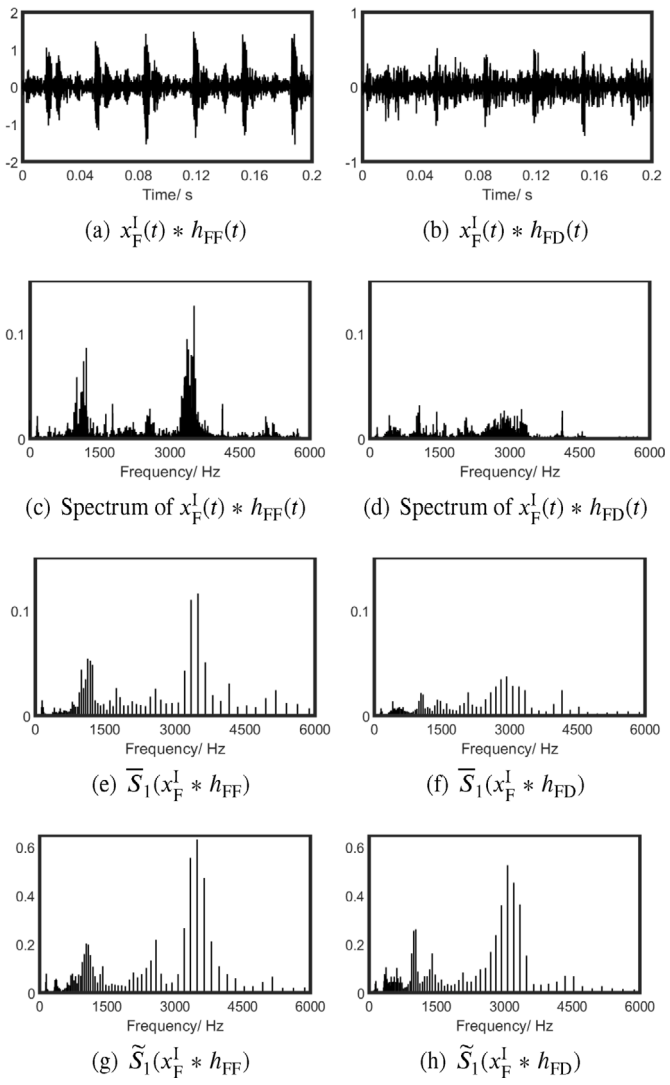


Fig. 16. Visualization of fan-end bearing signal's features.

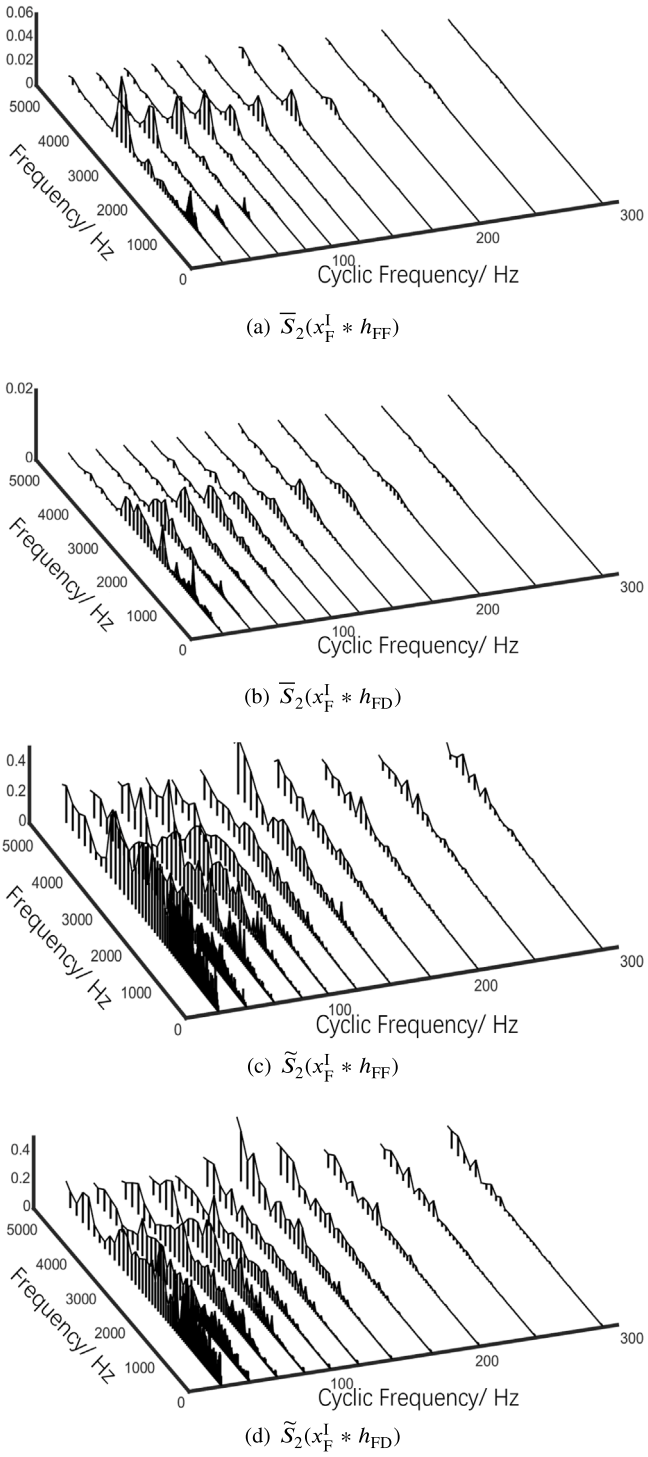


Fig. 17. Visualization of fan-end bearing signal's features.

4.3. Case study 1: transfer diagnosis across transducers at different positions

4.3.1. Transfer diagnosis tasks

The current experiments involve four datasets, *i.e.*, CWRU-DD, CWRU-DF, CWRU-FF, and CWRU-FD, each containing data collected from four working loads(0/1/2/3 hp). As presented in Table 3, 64 transfer diagnosis tasks are constructed based on the above datasets to verify NTScatNet's domain generalization capability across transmission paths. For example, the meaning of task A05 is learning diagnosis experience from the DD-0hp subdataset and applying the learned diagnosis knowledge to the DF-1hp subdataset. The other tasks follow similar conventions. Before introducing the compared methods, we would like to distinguish between the domain adaptation and generalization settings. The domain adaptation setting generally requires semi-supervised or unsupervised target domain data are available during the training phase. In contrast, the domain generalization setting does not require target domain information before the testing phase.

4.3.2. Compared methods

In order to illustrate NTScatNet's technique advantages, the Fourier spectrum methods, TScatNet model, and three recently published deep transfer learning models are also evaluated on the above 64 transfer diagnosis tasks. The compared methods are described below.

(1) Spectrum + Softmax: This method first uses a L point fast Fourier transform algorithm to calculate the frequency spectrum of the time-domain signals and then trains a Softmax classifier with the spectrum of the source domain data. The method follows the domain generalization setting.

(2) WD-DTL [22]: WD-DTL takes the Fourier spectrum as input and follows the domain adaptation setting. Specifically, it requires labeled source domain data and unlabeled target domain data during the

Table 3
Illustration of the constructed transfer diagnosis tasks.

Index	Source→Target	Index	Source→Target	Index	Source→Target	Index	Source→Target
A01	DD-0hp→DF-0hp	B01	DF-0hp→DD-0hp	C01	FF-0hp→FD-0hp	D01	FD-0hp→FF-0hp
A02	DD-1hp→DF-1hp	B02	DF-1hp→DD-1hp	C02	FF-1hp→FD-1hp	D02	FD-1hp→FF-1hp
A03	DD-2hp→DF-2hp	B03	DF-2hp→DD-2hp	C03	FF-2hp→FD-2hp	D03	FD-2hp→FF-2hp
A04	DD-3hp→DF-3hp	B04	DF-3hp→DD-3hp	C04	FF-3hp→FD-3hp	D04	FD-3hp→FF-3hp
A05	DD-0hp→DF-1hp	B05	DF-0hp→DD-1hp	C05	FF-0hp→FD-1hp	D05	FD-0hp→FF-1hp
A06	DD-0hp→DF-2hp	B06	DF-0hp→DD-2hp	C06	FF-0hp→FD-2hp	D06	FD-0hp→FF-2hp
A07	DD-0hp→DF-3hp	B07	DF-0hp→DD-3hp	C07	FF-0hp→FD-3hp	D07	FD-0hp→FF-3hp
A08	DD-1hp→DF-0hp	B08	DF-1hp→DD-0hp	C08	FF-1hp→FD-0hp	D08	FD-1hp→FF-0hp
A09	DD-1hp→DF-2hp	B09	DF-1hp→DD-2hp	C09	FF-1hp→FD-2hp	D09	FD-1hp→FF-2hp
A10	DD-1hp→DF-3hp	B10	DF-1hp→DD-3hp	C10	FF-1hp→FD-3hp	D10	FD-1hp→FF-3hp
A11	DD-2hp→DF-0hp	B11	DF-2hp→DD-0hp	C11	FF-2hp→FD-0hp	D11	FD-2hp→FF-0hp
A12	DD-2hp→DF-1hp	B12	DF-2hp→DD-1hp	C12	FF-2hp→FD-1hp	D12	FD-2hp→FF-1hp
A13	DD-2hp→DF-3hp	B13	DF-2hp→DD-3hp	C13	FF-2hp→FD-3hp	D13	FD-2hp→FF-3hp
A14	DD-3hp→DF-0hp	B14	DF-3hp→DD-0hp	C14	FF-3hp→FD-0hp	D14	FD-3hp→FF-0hp
A15	DD-3hp→DF-1hp	B15	DF-3hp→DD-1hp	C15	FF-3hp→FD-1hp	D15	FD-3hp→FF-1hp
A16	DD-3hp→DF-2hp	B16	DF-3hp→DD-2hp	C16	FF-3hp→FD-2hp	D16	FD-3hp→FF-2hp

Table 4
Hyperparameters of NTScatNet.

Q_1	Q_2	L	Optimizer	Learning rate	Weight decay	Batch size	Epoch
16	4	2400	Adam	0.001	0.0001	16	500

training phase. The architecture, hyperparameters, and implementation detail are consistent with the original paper [22].

(3) GANPair-N [21]: This method takes the Fourier spectrum as input and obeys the domain adaptation setting. GANPair-N requires labeled source domain data and unlabeled target domain data during the training phase. Besides, parallel data in the normal state of both source and the target domain are also required. The architecture, hyperparameters, and implementation detail are consistent with the original paper [21].

(4) GANPair-all [21]: The architecture, hyperparameters, and implementation detail are identical to GANPair-N. However, the GANPair-all method requires labeled source domain data, unlabeled target domain data, and parallel data in all possible health conditions of the source and the target domain.

(5) TScatNet [6]: TScatNet takes raw time-domain signal as input and follows the domain generalization setting. The hyperparameters adopted in the current experiment are given in Table 4.

(6) NTScatNet: The proposed NTScatNet takes the raw time-domain signal as input and follows the domain generalization setting. The hyperparameters adopted in the current experiment are given in Table 4.

4.3.3. Results of the transfer diagnosis experiments

Table 5 reports the six methods' diagnosis accuracies on 64 transfer diagnosis tasks, where each experiment is repeated ten times to reduce the influence of the random factors.

In order to compare the diagnosis performance of six methods intuitively, Fig. 18 presents the experimental results in the box plot. Each box in Fig. 18 involves 64 transfer diagnosis results, comprehensively characterizing a method's transfer diagnosis performance across transducers at different positions. As shown in the first box in Fig. 18, the Fourier spectrum method realizes an average accuracy of 52.5% on 64 transfer diagnosis tasks, indicating a severe data distribution discrepancy between the source and the target domain. Ref. [6] shows that TScatNet possesses excellent domain generalization capability across working load variation. However, as shown in the fifth box in Fig. 18, the average diagnosis accuracy of TScatNet is less than 82%, illustrating that TScatNet fails to address the domain generalization tasks across transducers. As shown in the sixth box in Fig. 18, the proposed NTScatNet achieves an average accuracy of 98.4% on 64 transfer diagnosis tasks. It should be worthing that the source and target domain samples in tasks A01–A04, B01–B04, C01–C04, and D01–D04, are sampled from different transducers under identical working

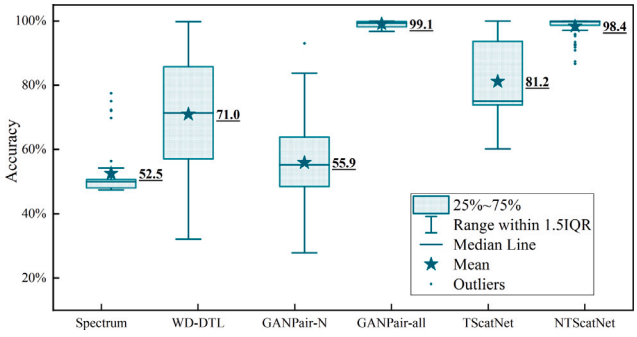


Fig. 18. Results of the transfer diagnosis experiments.

loads. By contrast, the source and target domain samples in tasks A05–A16, B05–B16, C05–C16, and D05–D16, are sampled from different transducers under different working loads. NTScatNet's excellent performance on 64 transfer diagnosis tasks illustrates that the proposed NTScatNet holds the domain generalization capability across different transducers and working loads simultaneously.

The second and the third boxes in Fig. 18 present the transfer diagnosis accuracies of WD-DTL and GANPair-N, respectively. WD-DTL achieves an average accuracy of 71% on 64 transfer diagnosis tasks. The best performance of WD-DTL is 99.8%(on task A04), while the worst accuracy is 50.7% accuracy(on task A13). GANPair-N achieves an average accuracy of 55.9% on 64 transfer diagnosis tasks. The best performance of GANPair-N is 93.0 ± 11.9%(on task C04), while the worst accuracy is only 27.9 ± 6.3% accuracy(on task B04). The above experiment results show that WD-DTL and GANPair-N fail to realize stable transfer diagnosis performance. As the fourth box in Fig. 18 presents, the GANPair-all achieves an average accuracy of 99.1% on 64 transfer diagnosis tasks, slightly outperforming NTScatNet by 0.7 percent. It should be noted that GANPair-all requires labeled source domain data, unlabeled target domain data, and parallel data of all source and target domain classes during the training phase. However, in practice, the target domain data is hard to obtain during the training phase. In contrast, the developed NTScatNet follows the domain generalization setting and only requires the labeled source domain data during the training phase, holding better practical application prospects than GANPair-all.

4.3.4. Results of unsupervised manifold embedding

Manifold embedding aims to recover the low-dimensional manifold structure from the high-dimensional data, where t-distribution stochastic neighbor embedding(t-SNE) is one of the most effective algorithms. This sub-section embeds wavelet scattering features and normalized

Table 5
Results of the transfer diagnosis experiments.

Index	Spectrum	WD-DTL	GANPair-N	GANPair-All	TScatNet	NTScatNet
A01	77.5 ± 7.8%	76.4 ± 23.7%	49.6 ± 23.6%	99.8 ± 0.3%	88.5 ± 6.5%	**99.8 ± 0.1%**
A02	47.5 ± 7.9%	86.3 ± 19.9%	54.8 ± 31.5%	99.0 ± 1.9%	83.2 ± 4.2%	**100 ± 0.1%**
A03	50.0 ± 0.1%	98.8 ± 2.0%	66.2 ± 30.1%	98.5 ± 1.5%	77.4 ± 2.2%	**99.7 ± 0.1%**
A04	50.0 ± 0.1%	99.8 ± 0.5%	57.5 ± 33.4%	100 ± 0.0%	79.2 ± 2.5%	**100 ± 0.1%**
A05	47.5 ± 7.9%	64.4 ± 16.8%	41.9 ± 11.8%	98.8 ± 2.8%	82.2 ± 4.9%	**100 ± 0.1%**
A06	47.5 ± 7.9%	54.3 ± 9.9%	40.0 ± 12.9%	99.6 ± 1.0%	73.5 ± 3.2%	**99.7 ± 0.1%**
A07	47.5 ± 7.9%	59.8 ± 12.2%	54.8 ± 28.4%	99.7 ± 0.8%	73.0 ± 4.3%	**98.2 ± 5.9%**
A08	77.5 ± 7.8%	53.4 ± 6.3%	64.8 ± 31.6%	99.7 ± 0.6%	90.5 ± 4.9%	**99.2 ± 0.2%**
A09	47.5 ± 7.9%	53.0 ± 18.4%	49.8 ± 22.7%	99.3 ± 1.1%	74.0 ± 3.5%	**99.4 ± 0.2%**
A10	47.5 ± 7.9%	68.5 ± 25.0%	52.0 ± 28.8%	99.9 ± 0.4%	73.5 ± 4.5%	**99.1 ± 0.3%**
A11	75.0 ± 0.1%	67.1 ± 20.9%	49.4 ± 23.7%	99.4 ± 1.5%	92.2 ± 6.4%	**99.8 ± 0.1%**
A12	50.0 ± 0.1%	57.1 ± 15.8%	54.1 ± 28.4%	98.2 ± 3.4%	85.9 ± 6.6%	**100 ± 0.1%**
A13	75.0 ± 0.1%	50.7 ± 18.4%	55.7 ± 21.4%	100 ± 0.0%	76.8 ± 2.1%	**100 ± 0.1%**
A14	50.0 ± 0.1%	57.4 ± 11.3%	42.3 ± 26.4%	99.4 ± 1.2%	93.0 ± 6.5%	**99.7 ± 0.1%**
A15	50.0 ± 0.1%	57.3 ± 15.3%	55.9 ± 29.2%	99.7 ± 0.6%	87.5 ± 7.5%	**100 ± 0.1%**
A16	50.0 ± 0.1%	64.7 ± 21.5%	50.7 ± 29.0%	98.0 ± 3.2%	80.3 ± 3.9%	**99.7 ± 0.1%**
B01	52.3 ± 8.0%	77.9 ± 15.7%	51.3 ± 18.7%	97.6 ± 1.2%	98.8 ± 2.2%	**100 ± 0.1%**
B02	72.3 ± 0.9%	95.9 ± 4.5%	47.3 ± 14.5%	97.4 ± 3.0%	99.9 ± 0.1%	**100 ± 0.1%**
B03	48.5 ± 1.7%	99.1 ± 1.3%	49.8 ± 25.6%	98.2 ± 0.8%	100 ± 0.1%	**99.9 ± 0.2%**
B04	54.3 ± 0.1%	97.9 ± 2.3%	27.9 ± 6.3%	97.0 ± 2.9%	99.7 ± 0.2%	**100 ± 0.1%**
B05	69.8 ± 7.0%	70.3 ± 13.6%	46.8 ± 25.0%	98.1 ± 1.2%	99.7 ± 1.0%	**98.0 ± 1.1%**
B06	48.2 ± 0.6%	85.2 ± 15.6%	58.7 ± 27.9%	96.8 ± 2.7%	99.9 ± 0.2%	**99.1 ± 1.6%**
B07	51.3 ± 0.9%	66.8 ± 12.1%	56.2 ± 20.3%	99.0 ± 1.3%	95.5 ± 3.9%	**99.7 ± 1.0%**
B08	52.3 ± 8.0%	74.1 ± 15.8%	40.0 ± 12.9%	97.9 ± 1.6%	97.4 ± 4.6%	**100 ± 0.1%**
B09	50.7 ± 0.9%	78.7 ± 16.7%	57.4 ± 25.9%	98.1 ± 2.2%	99.9 ± 0.4%	**100 ± 0.1%**
B10	56.3 ± 6.6%	73.2 ± 14.0%	50.6 ± 16.4%	98.9 ± 2.7%	94.4 ± 5.5%	**100 ± 0.1%**
B11	49.8 ± 0.1%	71.4 ± 13.7%	49.8 ± 23.9%	97.8 ± 1.0%	98.3 ± 2.1%	**100 ± 0.1%**
B12	69.8 ± 7.0%	71.5 ± 16.8%	45.0 ± 21.6%	98.3 ± 1.0%	100 ± 0.1%	**99.6 ± 0.3%**
B13	53.8 ± 1.3%	78.2 ± 18.5%	57.2 ± 24.4%	97.6 ± 1.2%	95.3 ± 2.2%	**100 ± 0.1%**
B14	49.8 ± 0.1%	64.1 ± 20.0%	31.5 ± 16.4%	97.5 ± 1.4%	100 ± 0.1%	**100 ± 0.1%**
B15	72.0 ± 0.1%	70.5 ± 12.0%	41.3 ± 24.4%	97.7 ± 2.5%	100 ± 0.1%	**99.9 ± 0.2%**
B16	48.0 ± 0.1%	73.3 ± 13.2%	42.3 ± 15.9%	98.5 ± 1.4%	100 ± 0.1%	**100 ± 0.1%**
C01	49.5 ± 10.6%	45.9 ± 8.4%	65.7 ± 24.4%	99.9 ± 0.1%	74.7 ± 0.6%	**95.9 ± 2.5%**
C02	47.5 ± 7.9%	97.1 ± 6.4%	79.4 ± 16.1%	99.9 ± 0.4%	75.0 ± 0.1%	**93.4 ± 0.7%**
C03	47.5 ± 7.9%	99.8 ± 0.2%	80.0 ± 19.9%	100 ± 0.0%	75.0 ± 0.1%	**95.8 ± 1.0%**
C04	50.0 ± 0.1%	96.6 ± 3.6%	93.0 ± 11.9%	97.7 ± 0.4%	75.0 ± 0.1%	**92.7 ± 1.1%**
C05	47.5 ± 7.9%	54.2 ± 16.5%	51.7 ± 16.1%	99.4 ± 0.8%	74.9 ± 0.3%	**92.5 ± 0.4%**
C06	47.5 ± 7.9%	72.1 ± 14.8%	56.3 ± 28.6%	100 ± 0.0%	74.8 ± 0.6%	**90.9 ± 1.1%**
C07	47.5 ± 7.9%	46.9 ± 14.3%	48.2 ± 23.8%	99.6 ± 0.4%	74.1 ± 2.9%	**86.7 ± 0.9%**
C08	49.5 ± 10.6%	40.7 ± 8.6%	83.7 ± 16.5%	100 ± 1.1%	75.0 ± 0.1%	**99.1 ± 0.2%**
C09	47.5 ± 7.9%	71.2 ± 25.6%	83.3 ± 11.7%	100 ± 0.1%	75.0 ± 0.1%	**91.8 ± 0.9%**
C10	47.5 ± 7.9%	86.5 ± 17.1%	58.3 ± 19.5%	100 ± 0.1%	74.5 ± 1.7%	**87.4 ± 1.0%**
C11	49.5 ± 10.6%	74.6 ± 20.3%	72.6 ± 11.7%	100 ± 0.1%	74.8 ± 0.1%	**100 ± 0.1%**
C12	47.5 ± 7.9%	74.5 ± 23.3%	70.8 ± 20.1%	99.4 ± 1.7%	75.0 ± 0.1%	**97.4 ± 0.9%**
C13	47.5 ± 7.9%	78.1 ± 23.4%	65.2 ± 23.9%	100 ± 0.1%	74.9 ± 0.2%	**92.8 ± 1.4%**
C14	50.0 ± 0.1%	57.1 ± 26.5%	67.8 ± 24.6%	100 ± 0.0%	74.1 ± 0.6%	**99.8 ± 0.1%**
C15	50.0 ± 0.1%	91.9 ± 14.9%	43.2 ± 12.8%	100 ± 0.0%	75.1 ± 0.1%	**97.1 ± 0.8%**
C16	50.0 ± 0.1%	73.9 ± 26.0%	66.0 ± 29.6%	100 ± 0.1%	75.0 ± 0.1%	**95.3 ± 0.6%**
D01	50.2 ± 0.6%	48.6 ± 12.6%	59.2 ± 10.3%	99.5 ± 1.0%	73.8 ± 10.5%	**100 ± 0.1%**
D02	50.0 ± 0.1%	97.5 ± 3.9%	50.3 ± 6.6%	99.4 ± 0.5%	77.3 ± 14.5%	**100 ± 0.1%**
D03	50.0 ± 0.1%	99.3 ± 0.6%	55.6 ± 15.3%	99.5 ± 1.0%	60.2 ± 9.3%	**100 ± 0.1%**
D04	50.1 ± 0.4%	96.2 ± 7.8%	60.1 ± 16.3%	100 ± 0.0%	71.3 ± 6.0%	**100 ± 0.1%**
D05	48.5 ± 4.8%	32.1 ± 8.3%	67.9 ± 9.7%	99.3 ± 0.3%	73.7 ± 10.6%	**99.4 ± 1.8%**
D06	48.4 ± 5.2%	50.1 ± 16.0%	65.4 ± 16.4%	99.4 ± 1.0%	68.8 ± 11.6%	**99.8 ± 0.6%**
D07	50.7 ± 2.2%	41.9 ± 13.6%	60.6 ± 25.1%	99.0 ± 1.2%	69.8 ± 11.8%	**98.5 ± 0.5%**
D08	50.0 ± 0.1%	48.7 ± 16.9%	39.2 ± 11.2%	99.7 ± 0.7%	76.2 ± 12.3%	**99.9 ± 0.1%**
D09	50.0 ± 0.1%	54.5 ± 28.8%	57.4 ± 23.2%	100 ± 0.0%	61.9 ± 12.7%	**100 ± 0.1%**
D10	50.0 ± 0.1%	86.5 ± 18.7%	48.9 ± 8.8%	99.3 ± 1.2%	64.9 ± 9.3%	**100 ± 0.1%**
D11	50.0 ± 0.1%	63.0 ± 15.1%	47.7 ± 15.2%	98.1 ± 2.3%	67.2 ± 11.2%	**99.0 ± 0.5%**
D12	50.0 ± 0.1%	63.2 ± 21.0%	58.6 ± 15.3%	99.7 ± 0.5%	66.0 ± 12.3%	**100 ± 0.1%**
D13	50.0 ± 0.1%	77.4 ± 19.8%	45.3 ± 14.2%	98.2 ± 2.2%	66.0 ± 8.7%	**100 ± 0.1%**
D14	49.4 ± 1.8%	43.4 ± 16.1%	54.7 ± 20.1%	99.9 ± 0.1%	74.1 ± 9.4%	**99.8 ± 0.1%**
D15	49.9 ± 0.5%	95.2 ± 2.1%	68.1 ± 16.3%	99.4 ± 0.1%	71.1 ± 9.1%	**100 ± 0.1%**
D16	50.6 ± 1.8%	63.3 ± 31.0%	63.0 ± 22.2%	99.5 ± 1.4%	66.9 ± 9.2%	**100 ± 0.1%**

scattering features of the CWRU dataset into two-dimensional space to intuitively verify that normalized scattering features' stability against transmission path variability.

Fig. 19 presents the t-SNE results of the CWRU-DD and CWRU-DF datasets. As Fig. 19(a) shows, scattering features from different working conditions are gathered together while features of different categories are separated, illustrating that wavelet scattering features are robust against working load variation while preserving discriminative to different categories. However, wavelet scattering features collected by transducers at different locations are not compact, meaning wavelet scattering fails to remove data distribution discrepancy brought by transmission path variability. In comparison, as Fig. 19(b) presents,

Table 6
NTScatNet's accuracies under 24 hyperparameter settings.

Index	Hyperparameters	Accuracy	Index	Hyperparameters	Accuracy
H01	$L = 1200, Q_1 = 8, Q_2 = 1$	90.3 ± 8.7%	H13	$L = 3600, Q_1 = 8, Q_2 = 1$	94 ± 8.2%
H02	$L = 1200, Q_1 = 8, Q_2 = 2$	88.7 ± 8.4%	H14	$L = 3600, Q_1 = 8, Q_2 = 2$	96.1 ± 6.6%
H03	$L = 1200, Q_1 = 8, Q_2 = 4$	90.1 ± 8.5%	H15	$L = 3600, Q_1 = 8, Q_2 = 4$	99 ± 2.4%
H04	$L = 1200, Q_1 = 16, Q_2 = 1$	89.9 ± 8.5%	H16	$L = 3600, Q_1 = 16, Q_2 = 1$	97.1 ± 5.0%
H05	$L = 1200, Q_1 = 16, Q_2 = 2$	85.3 ± 10.6%	H17	$L = 3600, Q_1 = 16, Q_2 = 2$	98.7 ± 2.9%
H06	$L = 1200, Q_1 = 16, Q_2 = 4$	88.5 ± 9.9%	H18	$L = 3600, Q_1 = 16, Q_2 = 4$	98.6 ± 3.4%
H07	$L = 2400, Q_1 = 8, Q_2 = 1$	92.1 ± 8.1%	H19	$L = 4800, Q_1 = 8, Q_2 = 1$	94.9 ± 8.1%
H08	$L = 2400, Q_1 = 8, Q_2 = 2$	94.8 ± 7.1%	H20	$L = 4800, Q_1 = 8, Q_2 = 2$	96.8 ± 6.2%
H09	$L = 2400, Q_1 = 8, Q_2 = 4$	96.5 ± 5.4%	H21	$L = 4800, Q_1 = 8, Q_2 = 4$	99.6 ± 1.8%
H10	$L = 2400, Q_1 = 16, Q_2 = 1$	96.0 ± 5.9%	H22	$L = 4800, Q_1 = 16, Q_2 = 1$	98.1 ± 4.1%
H11	$L = 2400, Q_1 = 16, Q_2 = 2$	95.7 ± 6.0%	H23	$L = 4800, Q_1 = 16, Q_2 = 2$	99.4 ± 2.0%
H12	$L = 2400, Q_1 = 16, Q_2 = 4$	98.4 ± 3.3%	H24	$L = 4800, Q_1 = 16, Q_2 = 4$	99.1 ± 3.0%

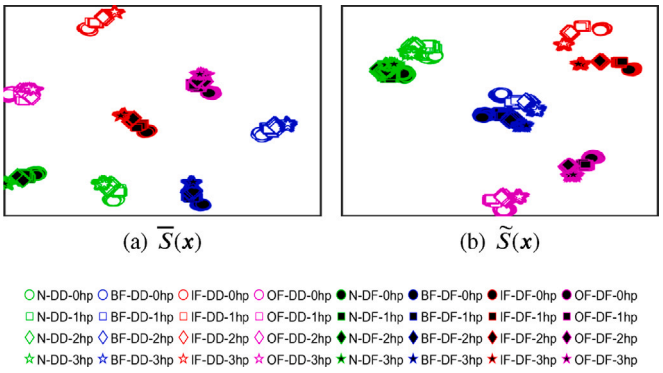


Fig. 19. t-SNE of CWRU-DD and CWRU-DF datasets.

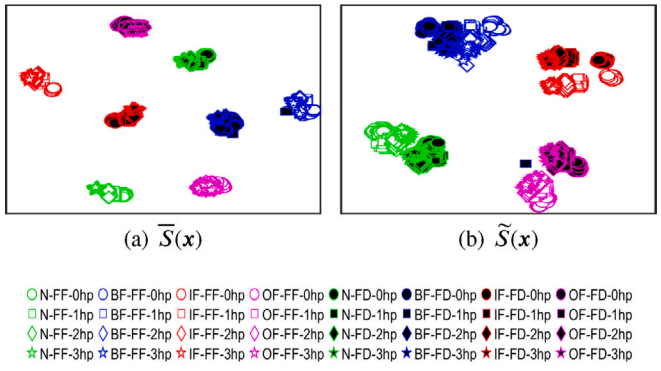


Fig. 20. t-SNE of CWRU-FF and CWRU-FD datasets.

normalized wavelet scattering features collected from different transducers become gathered together, implying that data distribution discrepancy due to transmission path variability has been significantly reduced. Fig. 20 shows the t-SNE results of the CWRU-FF and CWRU-FD datasets, which reports the same finding, *i.e.*, normalized wavelet scattering features successfully reduced domain discrepancy due to transmission path variability.

4.3.5. The effects of NTScatNet's hyperparameters

This subsection studies how hyperparameter settings affects NTScatNet's domain generalization diagnosis performance. Table 6 summarizes NTScatNet's average diagnosis accuracy on 64 transfer diagnosis tasks under 24 hyperparameter settings. As Table 6 reports, NTScatNet's performance is mainly influenced by the dimension of the input signal. NTScatNet's diagnosis accuracies with different input dimensions are presented in Fig. 21, which illustrates that NTScatNet's domain generalization performance improves as L increases. Specifically,

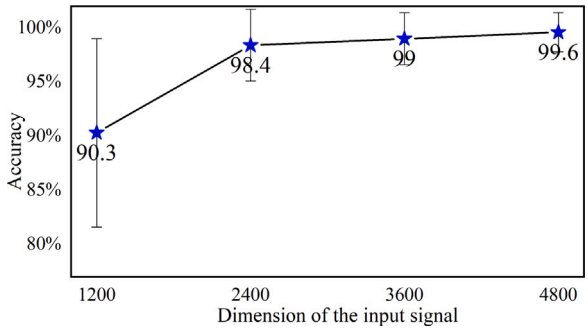


Fig. 21. The effects of L on NTScatNet's performance.

NTScatNet realizes 90.3±8.7% accuracy at hyperparameters H03, which is the best diagnosis accuracy when $L = 1200$. Besides, the proposed NTScatNet achieves 98.4 ± 3.3% and 99 ± 2.4% accuracy at hyperparameters H12 and H15, respectively, where 98.4 ± 3.3% is the best diagnosis accuracy when $L = 2400$ and 99 ± 2.4% is the best diagnosis accuracy when $L = 3600$. Finally, NTScatNet realizes 99.6 ± 1.8% accuracy at hyperparameters H21, which is the best diagnosis accuracy when $L = 4800$. Section 3.4.2 has derived that NTScatNet's computational complexity is approximately the order of $L(log_2 L)^3$, which grows not fastly as L increases. Therefore, in practice, selecting a large dimension of input signals is recommended to improve NTScatNet diagnosis performance as the computing resources allow.

4.4. Case study 2: detecting foreign objects left on the escalator guide rail

This case study applies the proposed NTScatNet to detect foreign objects at different positions of the escalator guide rail, which provides the other case to verify NTScatNet's domain generalization capability across transmission paths.

4.4.1. Background

The escalator is a vertical transportation system for transporting passengers over a short distance, consisting of a series of end-to-end steps, guide rails, et al. The guide rail is installed on the escalator truss and is used to support the escalator step. The steps are pulled by the step chain and circulate on the guide rails to transport passengers. In the daily use of escalators, some passengers may unintentionally discard foreign objects, and the foreign objects may fall on the escalator guide rails. It will generate a jolt when the escalator step rollers run over the foreign objects left on the escalator guide rail. It is practically significant to develop an IFD method to automatically detect foreign objects left on the escalator guide rail for escalator comfort and safety.

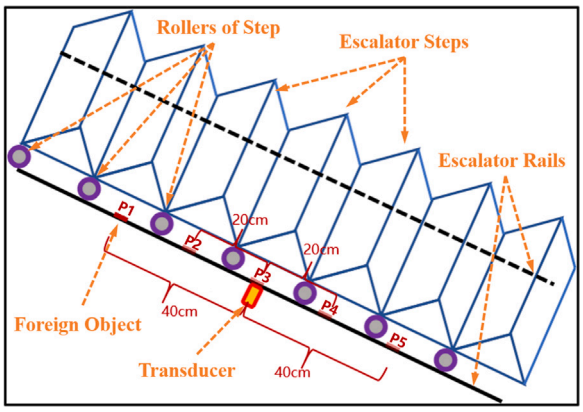
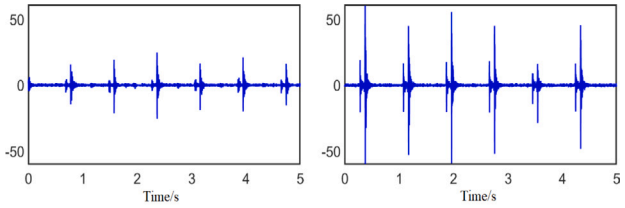


Fig. 22. Diagram of the experimental scheme.



(a) $x_{P1}(t)$ (b) $x_{P2}(t)$

Fig. 23. Time domain signals.

4.4.2. Principles of detecting foreign objects left on the escalator guide rail

The escalator guide rail length is usually more than 1 meter, and the foreign object may be left at various positions of the guide rail. Fig. 22 presents the diagram of the experimental scheme, which shows that the transducer is mounted on the middle of the guide rail, i.e., P3. Besides, the experimental scheme gives five potential positions of the foreign objects, i.e., P1–P5. Since the transmission path characteristics between the foreign object and the transducer vary with the position of the foreign object, a significant distribution discrepancy generally exists between fault signals generated by foreign objects at different positions. Fig. 23(a) and 23(b) present the signal generated by a foreign object at P3 and P1 positions, respectively, which shows a significant amplitude discrepancy. The domain discrepancy due to transmission variability brings difficulties to the IFD method and should be removed before the diagnostic decision.

4.4.3. Datasets and transfer diagnosis tasks

This subsection introduces the datasets and diagnosis tasks that are used to verify NTScatNet's domain generalization capability across transmission paths. Fig. 24 presents the escalator experimental platform where the datasets are collected. Fig. 25(a) presents the transducer installation position in the escalator guide rail. The condition monitoring signals of the escalator guide rail are collected at the sampling frequency of 1200 Hz. Firstly, the vibration signals of the escalator guide rail in the normal state are recorded for 5 min. Then a coin is pasted on the rail to simulate the foreign object left on the escalator guide rail. Fig. 22 gives five positions where the coin is pasted, *i.e.*, P1–P5. More specifically, position P3 is nearby the transducer, P2 and P4 are 20 centimeters from the transducer, and P1 and P5 are 40 centimeters from the transducer. Figs. 25(b) and 25(c) show the foreign body at P3 and P4, respectively. The signals of the guide rail in an abnormal state are recorded for 1 min for each foreign object position.

(1) Datasets: The vibration signal in the normal state is divided into 500 samples of dimension L with the overlapping trick, while the vibration signal in the abnormal state is divided into 100 samples

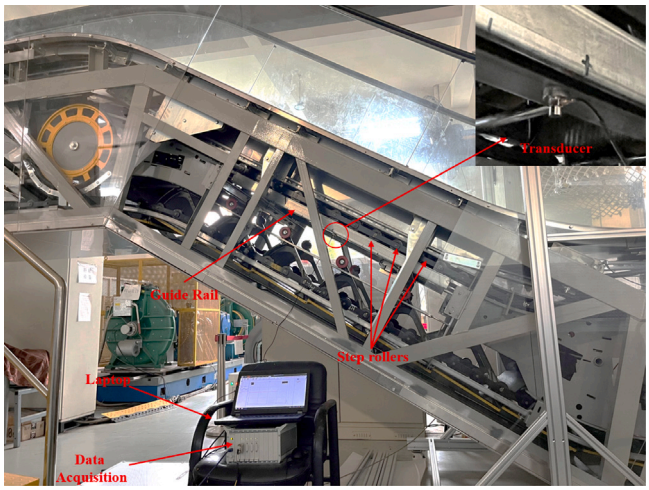


Fig. 24. Escalator experimental platform.

(a) Position of transducer (b) The foreign object at P3

(c) The foreign object at P4

Fig. 25. The transducer installation position and the position of the foreign body in the guide.

Table 7
Information of escalator datasets,belowfloat=15pt,abovefloat=15pt.

	ES_{P1}	ES_{P2}	ES_{P3}	ES_{P4}	ES_{P5}
Position of the foreign object	P1	P2	P3	P4	P5
Number of normal samples	100	100	100	100	100
Number of abnormal samples	100	100	100	100	100
Dimension of the sample	L	L	L	L	L

Table 8
Transfer diagnosis tasks on the escalator datasets.

Index	Source→Target domain	Index	Source→Target domain
T01	$ES_{P1} \rightarrow ES_{P2}$	T11	$ES_{P3} \rightarrow ES_{P4}$
T02	$ES_{P1} \rightarrow ES_{P3}$	T12	$ES_{P3} \rightarrow ES_{P5}$
T03	$ES_{P1} \rightarrow ES_{P4}$	T13	$ES_{P4} \rightarrow ES_{P1}$
T04	$ES_{P1} \rightarrow ES_{P5}$	T14	$ES_{P4} \rightarrow ES_{P2}$
T05	$ES_{P2} \rightarrow ES_{P1}$	T15	$ES_{P4} \rightarrow ES_{P3}$
T06	$ES_{P2} \rightarrow ES_{P3}$	T16	$ES_{P4} \rightarrow ES_{P5}$
T07	$ES_{P2} \rightarrow ES_{P4}$	T17	$ES_{P5} \rightarrow ES_{P1}$
T08	$ES_{P2} \rightarrow ES_{P5}$	T18	$ES_{P5} \rightarrow ES_{P2}$
T09	$ES_{P3} \rightarrow ES_{P1}$	T19	$ES_{P5} \rightarrow ES_{P3}$
T10	$ES_{P3} \rightarrow ES_{P2}$	T20	$ES_{P5} \rightarrow ES_{P4}$

Table 9
Hyperparameters of NTScatNet.

Q_1	Q_2	L	Optimizer	Learning rate	Weight decay	Batch size	Epoch
16	4	2400	Adam	0.001	0.0001	16	500

Table 10
Experimental results.

Tasks	Spectrum	TScatNet	NTScatNet
T01	100 ± 0.0%	50.9 ± 0.7%	**100 ± 0.0%**
T02	91.6 ± 9.3%	50.0 ± 0.0%	**100 ± 0.0%**
T03	91.0 ± 7.5%	50.0 ± 0.0%	**100 ± 0.0%**
T04	79.8 ± 13.7%	81.9 ± 1.1%	**100 ± 0.0%**
T05	83.5 ± 19.7%	100 ± 0.0%	**100 ± 0.0%**
T06	97.4 ± 6.9%	55.2 ± 2.0%	**100 ± 0.0%**
T07	99.9 ± 0.2%	99 ± 0.0%	**100 ± 0.0%**
T08	71.0 ± 19.1%	100 ± 0.0%	**100 ± 0.0%**
T09	50.5 ± 1.2%	100 ± 0.0%	**100 ± 0.0%**
T10	81.1 ± 20.9%	100 ± 0.0%	**100 ± 0.0%**
T11	85.9 ± 16.2%	100 ± 0.0%	**100 ± 0.0%**
T12	57.5 ± 14.2%	99.9 ± 0.3%	**100 ± 0.0%**
T13	59.1 ± 10.9%	99 ± 0.0%	**99.5 ± 0.5%**
T14	99.9 ± 0.3%	100 ± 0.0%	**100 ± 0.0%**
T15	94.1 ± 12.7%	88.0 ± 4.6%	**100 ± 0.0%**
T16	69.9 ± 21.2%	100 ± 0.0%	**100 ± 0.0%**
T17	92.1 ± 9.9%	100 ± 0.0%	**100 ± 0.0%**
T18	99.7 ± 0.8%	63.9 ± 2.2%	**100 ± 0.0%**
T19	79.4 ± 12.7%	50.0 ± 0.9%	**100 ± 0.0%**
T20	99.9 ± 0.2%	50.5 ± 0.2%	**100 ± 0.0%**

of dimension L. A total of 500 samples in the normal state and 500 samples in the abnormal state are obtained. As Table 7 presents, five datasets are constructed using the above 100 samples, where the ES_{Pi} dataset contains the $(i-1)*100+1$st to $(i*100)$th normal samples and the abnormal samples of position Pi.

(2) Transfer diagnosis tasks: As presented in Table 8, 20 transfer diagnosis tasks are constructed based on the escalator datasets to verify NTScatNet's domain generalization capability across transmission paths. For example, the meaning of task T04 is learning diagnosis experience from the ES_{P1} dataset and applying the learned diagnosis knowledge to the ES_{P5} subdataset. The other tasks follow similar conventions.

4.4.4. Experimental results

Table 10 and Fig. 26 report NTScatNet's accuracies on tasks T01–T20, where each experiment is repeated ten times to reduce the influence of random factors. NTScatNet's hyperparameters adopted in the current experiments are given in Table 9. The Fourier spectrum and the TScatNet methods were also evaluated on the above tasks as comparisons. As Fig. 26 shows, the Fourier spectrum method realized an average 84.2% accuracy on tasks T01–T20, which is inferior to the proposed NTScatNet 15.8%. Besides, TScatNet achieved an average 81.9% accuracy on tasks T01–T20, inferior to NTScatNet 18.1%. Specifically, as Table 10 presents, the Fourier spectrum method realized only 50.5 ± 1.2%, 57.5 ± 14.2%, and 59.1 ± 10.9% accuracies on tasks T09, T12, and T13, respectively. Moreover, the accuracies of the TScatNet method on tasks T01–T03, T06, and T18–T20 are generally below 70%, indicating significant domain discrepancies due to transmission variabilities. In comparison, the developed NTScatNet achieves above 99.5%±0.5% accuracy on all 20 transfer diagnosis tasks. The above experimental results verify that NTScatNet holds excellent domain generalization diagnosis capability across different transmission paths and is practical to detect foreign objects left on the escalator guide rail.

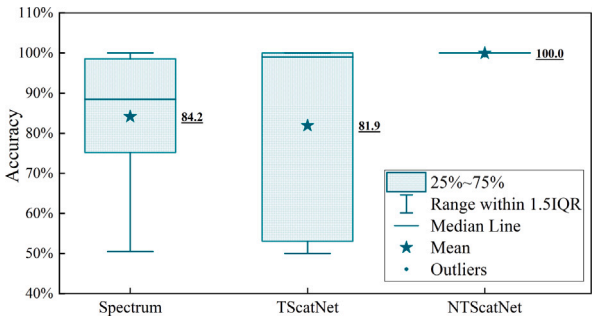


Fig. 26. Results of the transfer diagnosis experiments.

4.4.5. Effects of hyperparameters

Table 11 reports NTScatNet's average diagnosis accuracy on 20 diagnosis tasks under 24 hyperparameter settings. As Table 11 presents, NTScatNet realizes above 99.8% accuracy when the dimension of the input signal takes 1200 and achieves 100% accuracy when the input signal dimension is larger than 1200. The experimental results illustrate that the proposed NTScatNet hold excellent domain generalization capability across transmission paths under various hyperparameters.

5. Conclusion

This article developed an interpretable convolutional neural network, *i.e.*., NTScatNet, for domain generalization diagnosis across transmission paths. NTScatNet's architecture is similar to a standard CNN, while NTScatNet's distinctiveness is that it takes Morlet wavelet convolutional kernel, modulo activation function, and scaling function window-based average averaging pooling layer. This article gave a detailed physical interpretation of each layer of NTScatNet and showed that NTScatNet's feature extractor well characterizes the multi-scale cyclostationarity information of the fault signals. Besides, we theoretically illustrated the normalized scattering feature's invariance to a linear time-invariant system. Finally, we experimentally verified that NTScatNet holds the domain generalization diagnosis capability across different transmission paths through transfer diagnosis tasks across transducers and the tasks of detecting foreign objects on the escalator guide rail.

CRediT authorship contribution statement

Chao Liu: Idea, Methodology, Original draft, Software, Data analysis. **Xiaolong Ma:** Data collecting, Data analysis. **Tianyu Han:** Software, Data analysis. **Xi Shi:** Writing – review & editing. **Chengjin Qin:** Review, Supervision. **Songtao Hu:** Review & language editing.

Declaration of competing interest

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

Data availability

The data that has been used is confidential.

Acknowledgment

The authors acknowledge the supported by the National Natural Science Foundation of China under Grant No. 51935007.

Table 11
NTScatNet's accuracies under 24 hyperparameter settings.

Index	Hyperparameters	Accuracy	Index	Hyperparameters	Accuracy
H01	$L = 1200, Q_1 = 8, Q_2 = 1$	100 ± 0.1%	H13	$L = 3600, Q_1 = 8, Q_2 = 1$	100 ± 0.0%
H02	$L = 1200, Q_1 = 8, Q_2 = 2$	99.8 ± 0.5%	H14	$L = 3600, Q_1 = 8, Q_2 = 2$	100 ± 0.1%
H03	$L = 1200, Q_1 = 8, Q_2 = 4$	99.9 ± 0.3%	H15	$L = 3600, Q_1 = 8, Q_2 = 4$	100 ± 0.1%
H04	$L = 1200, Q_1 = 16, Q_2 = 1$	99.8 ± 0.6%	H16	$L = 3600, Q_1 = 16, Q_2 = 1$	100 ± 0.2%
H05	$L = 1200, Q_1 = 16, Q_2 = 2$	99.8 ± 0.6%	H17	$L = 3600, Q_1 = 16, Q_2 = 2$	100 ± 0.1%
H06	$L = 1200, Q_1 = 16, Q_2 = 4$	99.8 ± 0.4%	H18	$L = 3600, Q_1 = 16, Q_2 = 4$	100 ± 0.1%
H07	$L = 2400, Q_1 = 8, Q_2 = 1$	100 ± 0.0%	H19	$L = 4800, Q_1 = 8, Q_2 = 1$	100 ± 0.0%
H08	$L = 2400, Q_1 = 8, Q_2 = 2$	100 ± 0.0%	H20	$L = 4800, Q_1 = 8, Q_2 = 2$	100 ± 0.0%
H09	$L = 2400, Q_1 = 8, Q_2 = 4$	100 ± 0.0%	H21	$L = 4800, Q_1 = 8, Q_2 = 4$	100 ± 0.0%
H10	$L = 2400, Q_1 = 16, Q_2 = 1$	100 ± 0.0%	H22	$L = 4800, Q_1 = 16, Q_2 = 1$	100 ± 0.0%
H11	$L = 2400, Q_1 = 16, Q_2 = 2$	100 ± 0.0%	H23	$L = 4800, Q_1 = 16, Q_2 = 2$	100 ± 0.0%
H12	$L = 2400, Q_1 = 16, Q_2 = 4$	100 ± 0.0%	H24	$L = 4800, Q_1 = 16, Q_2 = 4$	100 ± 0.0%

References

[1] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A.K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mech. Syst. Signal Process. 138 (2020).

[2] J. Antoni, A critical overview of the filterbank-feature-decision methodology in machine condition monitoring, Acoust. Aust. 49 (2) (2021) 177–184, http://dx.doi.org/10.1007/s40857-021-00232-7.

[3] Y.G. Lei, F. Jia, J. Lin, S.B. Xing, S.X. Ding, An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data, Ieee Trans. Ind. Electron. 63 (5) (2016) 3137–3147.

[4] F. Jia, Y.G. Lei, N. Lu, S.B. Xing, Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization, Mech. Syst. Signal Process. 110 (2018) 349–367.

[5] T. Li, Z. Zhao, C. Sun, L. Cheng, X. Chen, R. Yan, R.X. Gao, Waveletkernelnet: An interpretable deep neural network for industrial intelligent diagnosis, IEEE Trans. Syst. Man Cybern.: Syst. (2021).

[6] C. Liu, C. Qin, X. Shi, Z. Wang, G. Zhang, Y. Han, Tscatnet: An interpretable cross-domain intelligent diagnosis model with anti-noise and few-shot learning capability, IEEE Trans. Instrum. Meas., 0 http://dx.doi.org/10.1109/TIM.2020.3041905.

[7] S. Mallat, Group invariant scattering, Comm. Pure Appl. Math. 65 (10) (2012) 1331–1398.

[8] J. Bruna, S. Mallat, Invariant scattering convolution networks, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1872–1886.

[9] J. Andén, S. Mallat, Deep scattering spectrum, IEEE Trans. Signal Process. 62 (16) (2014) 4114–4128, http://dx.doi.org/10.1109/TSP.2014.2326991.

[10] S. Mallat, Understanding deep convolutional networks, Phil. Trans. R. Soc. A 374 (2065) (2016).

[11] B.Y.J. Bruna, S.T. Mallat, E. Bacry, J.F.R. Muzy, Intermittent process analysis with scattering moments, Ann. Statist. 43 (1) (2015) 323–351, http://dx.doi.org/10.1214/14-AOS1276.

[12] M. Andreux, T. Angles, G. Exarchakis, R. Leonarduzzi, G. Rochette, L. Thiry, J. Zarka, S. Mallat, J. Andén, E. Belilovsky, J. Bruna, V. Lostanlen, M. Chaudhary, M.J. Hirn, E. Oyallon, S. Zhang, C. Cella, M. Eickenberg, Kymatio: Scattering transforms in python, J. Mach. Learn. Res. 21 (2020).

[13] R. Liu, B. Yang, E. Zio, X. Chen, Artificial intelligence for fault diagnosis of rotating machinery: A review, Mech. Syst. Signal Process. 108 (2018) 33–47.

[14] Z. Zhao, T. Li, J. Wu, C. Sun, S. Wang, R. Yan, X. Chen, Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study, ISA Trans. 107 (2020) 224–255.

[15] Z. Zhao, Q. Zhang, X. Yu, C. Sun, S. Wang, R. Yan, X. Chen, Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study, IEEE Trans. Instrum. Meas. 70 (2021) http://dx.doi.org/10.1109/TIM.2021.3116309.

[16] H. Zheng, R. Wang, Y. Yang, J. Yin, Y. Li, Y. Li, M. Xu, Cross-domain fault diagnosis using knowledge transfer strategy: A review, IEEE Access 7 (2019) 129260–129290.

[17] W. Li, R. Huang, J. Li, Y. Liao, Z. Chen, G. He, R. Yan, K. Gryllias, A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges, Mech. Syst. Signal Process. 167 (2022).

[18] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, T. Zhang, Deep model based domain adaptation for fault diagnosis, IEEE Trans. Ind. Electron. 64 (3) (2017) 2296–2305.

[19] X. Li, W. Zhang, Q. Ding, J.Q. Sun, Multi-layer domain adaptation method for rolling bearing fault diagnosis, Signal Process. 157 (2019) 180–197.

[20] X. Li, W. Zhang, Q. Ding, A robust intelligent fault diagnosis method for rolling element bearings based on deep distance metric learning, Neurocomputing 310 (2018) 77–95.

[21] X. Li, W. Zhang, N.X. Xu, Q. Ding, Deep learning-based machinery fault diagnostics with domain adaptation across sensors at different places, IEEE Trans. Ind. Electron. 67 (8) (2020) 6785–6794.

[22] C. Cheng, B. Zhou, G. Ma, D. Wu, Y. Yuan, Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis with unlabeled or insufficient labeled data, Neurocomputing 409 (2020) 35–45.

[23] L. Chen, Q. Li, C. Shen, J. Zhu, D. Wang, M. Xia, Adversarial domain-invariant generalization: A generic domain-regressive framework for bearing fault diagnosis under unseen conditions, IEEE Trans. Ind. Inf. 18 (3) (2022) 1790–1800, http://dx.doi.org/10.1109/TII.2021.3078712.

[24] T. Han, Y.F. Li, M. Qian, A hybrid generalization network for intelligent fault diagnosis of rotating machinery under unseen working conditions, IEEE Trans. Instrum. Meas. 70 (2021) http://dx.doi.org/10.1109/TIM.2021.3088489.

[25] J. Li, C. Shen, L. Kong, D. Wang, M. Xia, Z. Zhu, A new adversarial domain generalization network based on class boundary feature detection for bearing fault diagnosis, IEEE Trans. Instrum. Meas. 71 (2022) http://dx.doi.org/10.1109/TIM.2022.3164163.

[26] Y. Liao, R. Huang, J. Li, Z. Chen, W. Li, Deep semisupervised domain generalization network for rotary machinery fault diagnosis under variable speed, IEEE Trans. Instrum. Meas. 69 (10) (2020) 8064–8075, http://dx.doi.org/10.1109/TIM.2020.2992829.

[27] M. Ragab, Z. Chen, W. Zhang, E. Eldele, M. Wu, C.K. Kwoh, X. Li, Conditional contrastive domain generalization for fault diagnosis, IEEE Trans. Instrum. Meas. 71 (2022) http://dx.doi.org/10.1109/TIM.2022.3154000.

[28] H. Wang, X. Bai, S. Wang, J. Tan, C. Liu, Generalization on unseen domains via model-agnostic learning for intelligent fault diagnosis, IEEE Trans. Instrum. Meas. 71 (2022) http://dx.doi.org/10.1109/TIM.2022.3152316.

[29] Q. Zhang, Z. Zhao, X. Zhang, Y. Liu, C. Sun, M. Li, S. Wang, X. Chen, Conditional adversarial domain generalization with a single discriminator for bearing fault diagnosis, IEEE Trans. Instrum. Meas. 70 (2021) http://dx.doi.org/10.1109/TIM.2021.3071350.

[30] C. Zhao, W. Shen, Adaptive open set domain generalization network: Learning to diagnose unknown faults under unknown working conditions, Reliab. Eng. Syst. Saf. 226 (2022) http://dx.doi.org/10.1016/j.ress.2022.108672.

[31] C. Zhao, W. Shen, Adversarial mutual information-guided single domain generalization network for intelligent fault diagnosis, IEEE Trans. Ind. Inf. (2022) 1, http://dx.doi.org/10.1109/TII.2022.3175018.

[32] C. Zhao, W. Shen, A domain generalization network combing invariance and specificity towards real-time intelligent fault diagnosis, Mech. Syst. Signal Process. 173 (2022) http://dx.doi.org/10.1016/j.ymssp.2022.108990.

[33] Chengjin Qin, Yanrui Jin, Jianfeng Tao, Dengyu Xiao, Honggan Yu, Chao Liu, Gang Shi, Junbo Lei, Chengliang Liu, Dtcnnmi: a deep twin convolutional neural networks with multi-domain inputs for strongly noisy diesel engine misfire detection, Measurement: Journal of the International Measurement Confederation 180 (2021) http://dx.doi.org/10.1016/j.measurement.2021.109548.

Mechanical Systems and Signal Processing 207 (2024) 110952

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

TFN: An interpretable neural network with time-frequency transform embedded for intelligent fault diagnosis

Qian Chen [a], Xingjian Dong [a,*], Guowei Tu [a], Dong Wang [a], Changming Cheng [a], Baoxuan Zhao [a], Zhike Peng [a,b]

[a] State Key Laboratory of Mechanical System and Vibration, Institue of Vibration Shock & Noise, Shanghai Jiao Tong University, Shanghai 200240, China

[b] School of Mechanical Engineering, Ningxia University, Ningxia 750021, China

ARTICLE INFO

Communicated by J. Antoni

Keywords:
Convolutional neural network (CNN)
Time-frequency transform
Interpretability
Intelligent fault diagnosis
Prognostics and health management

ABSTRACT

Convolutional neural network (CNN) is widely used in fault diagnosis of mechanical systems due to its powerful feature extraction and classification capabilities. However, the CNN is a typical black-box model, and the mechanism of CNN's decision-making is not clear, which limits its application in high-reliability-required fault diagnosis scenarios. To tackle this issue, we propose a novel interpretable neural network termed as time-frequency network (TFN), where the physically meaningful time-frequency transform (TFT) method is embedded into the traditional convolutional layer as a trainable preprocessing layer. This preprocessing layer named as time-frequency convolutional (TFconv) layer, is constrained by a well-designed kernel function to extract fault-related time-frequency information. It not only improves the diagnostic performance but also reveals the logical foundation of the CNN prediction in a frequency domain view. Different TFT methods correspond to different kernel functions of the TFconv layer. In this study, three typical TFT methods are considered to formulate the TFNs and their diagnostic effectiveness and interpretability are proved through three mechanical fault diagnosis experiments. Experimental results also show that the proposed TFconv layer has outstanding advantages in convergence speed and few-shot scenarios, and can be easily generalized to other CNNs with different depths to improve their diagnostic performances. The code of TFN is available on https://github.com/ChenQian0618/TFN.

1. Introduction

Nowadays, fault diagnosis of mechanical equipment is widely used in industry to reduce property damage and improve production efficiency [1,2]. With the development of sensor technology and the industrial internet, a large amount of operation and maintenance data could be obtained by various sensors [3], such as accelerometers, dynamometers, and microphones. Based on sufficient operation and maintenance data, the data-driven method [4], as a model-free solution with high diagnostic accuracy, has gradually gained more and more attention in the field of mechanical equipment fault diagnosis.

The process of data-driven fault diagnosis can be divided into three steps: data acquisition, feature extraction, and fault classification [5]. Among these steps, feature extraction is the key to fault diagnosis [6,7]. Traditional feature extraction mainly relies on signal processing methods, including Fourier transform (FT), short time Fourier transform (STFT), wavelet transform (WT) [8],

* Corresponding author.
E-mail addresses: chenqian2020@sjtu.edu.cn (Q. Chen), donxij@sjtu.edu.cn (X. Dong), guoweitu@sjtu.edu.cn (G. Tu), dongwang4-c@sjtu.edu.cn (D. Wang), ccming@sjtu.edu.cn (C. Cheng), bxzhao@sjtu.edu.cn (B. Zhao), z.peng@sjtu.edu.cn (Z. Peng).

https://doi.org/10.1016/j.ymssp.2023.110952
Received 13 June 2023; Received in revised form 6 November 2023; Accepted 14 November 2023
Available online 20 November 2023

and empirical mode decomposition (EMD) [9]. Using these methods, researchers can transform raw maintenance data (i.e., vibration signals) from the time domain into the frequency or time-frequency domain, and thus extract essential features for fault classification. However, such a strategy requires too much expertise and prior knowledge, which is difficult to be widely conducted in real industrial scenarios. Therefore, deep learning, conducting feature extraction and classification automatically, becomes the promising solution for mechanical fault diagnosis [5].

As a powerful representation learning technique, deep learning provides end-to-end solutions and is widely used in computer vision [10], natural language processing [11], game competition [12], and other fields. In terms of fault diagnosis, deep learning has a series of nonlinear mapping layers to extract hidden key information from vibration signals. Many deep learning models have been applied to mechanical fault diagnosis, such as deep belief network (DBN) [13], convolutional neural network (CNN) [14–16], and recurrent neural network (RNN) [17]. Among these neural networks, CNN-based models can fully extract the spatial information of the input signal and thus leads to high diagnostic accuracy in multiple public datasets for mechanical fault diagnosis [18].

Despite its superior diagnostic performance, the CNN has a weak spot — its interpretability [19]. It is difficult to find the logical foundation of the CNN model for feature extraction and classification. This reduces the credibility of the results and prohibits the breakthrough of the model performance, which in turn limits its application in high-reliability-required fault diagnosis scenarios (e.g. aero engine fault diagnosis [20]). Therefore, it is of great importance to develop interpretable CNNs for high-quality fault diagnosis of mechanical systems.

The research of interpretable CNN is a growing topic in the AI field [21,22]. Current interpreting methods can be roughly divided into four categories [23]: rule type, semantic type, attribution type, and example type. Rule type is to extract the mapping relationship of CNNs into specific logical rules (e.g., CEM [24], CDRP [25]). Semantic type is to completely analyze the meaning of specific hidden layers in CNNs (e.g., Network dissection [26]). Attribution type is to quantify the contribution (or negative effect) of the input and features (e.g., salient map [27], Grad-CAM [28], IG [29]). Example type is to summarize typical samples related to the category (e.g., Prototype Net [30]). The above interpreting methods are mainly designed for 2D images, and when it comes to 1D vibration signals in mechanical fault diagnosis, these techniques are not suitable.

At present, there are few studies to explore the interpretability of CNNs in the field of mechanical fault diagnosis. Li et al. [31] calculated the importance weight of each point of the vibration signal sample to the CNN prediction through the integrated gradient (IG) method, and they explained the foundation of prediction-making through the frequency spectra of calculated weights. Wu et al. [32] transformed 1D vibration signals into 2D time-frequency spectra as the input samples, so that the 2D interpreting method (i.e., Grad-CAM) could be adopted to obtain the model attention area. Wang et al. [33] extended the extreme learning machine (ELM) to an interpretable structure for machine state monitoring. With this strategy, the information frequency band can be automatically located. Zhao et al. [34] incorporated the reproducing kernel Hilbert space (RKHS) into the convolutional layer and built a specific interpretable denoising convolutional layer. Li et al. [35] combined continuous wavelet transform with convolutional layers to formulate a wavelet kernel network to extract interpretable features. The above studies explained the mechanism of CNNs in fault diagnosis to some extent, but their interpretations are usually ambiguous and need some subjective understandings (e.g., the similarity between different frequency spectra of inputs [31], the sensibility of CNN to impulsive signal [35]). On top of that, these strategies may bring some other issues: degraded diagnostic performance and poor generalizability.

To propose an interpretable CNN model for fault diagnosis, we set our sights on traditional signal processing methods which are physically meaningful and good at feature extraction. Considering that the Fourier-type transform and the convolutional layer can be both regarded as inner products, we embed the time-frequency transform (TFT) method into the traditional convolutional layer. This leads to a novel layer named time-frequency convolutional (TFconv) layer, which is constrained by a well-designed kernel function in order to extract fault-related time-frequency information. Using the TFconv layer as the preprocessing layer, we constructed the time-frequency network (TFN) for fault diagnosis tasks. With this interpretable TFN, we not only improve the accuracy of fault diagnosis, but also reveal the logical foundation of prediction-making in the frequency domain through the frequency response analysis. A series of mechanical fault diagnosis experiments verify the superior diagnostic performance and the clear interpretability of TFN.

The idea of combining signal processing methods with neural networks has already been proposed in the previous works as shown in Table 1, but our method has enough novelty and is distinguished in the following three aspects. 1) Distinctive motivation: Our method tries to parameterize the convolutional layer to simulate time-frequency transform which is very novel against the existing literature; 2) Considering complex value kernel: Previous works [35–37] are trying to initialize or parameterize real value kernel and are equivalent to FIR filters, whose outputs are the filtered sub-signals, while our work takes complex value kernel in consideration and is equivalent to time-frequency transform, whose output is the energy distribution of the signal in the time-frequency domain. 3) A win-win of performance and interpretability: our proposed method not only improves the accuracy of fault diagnosis, but also explains the focusing frequency of CNN models.

The main contributions of this study could be summarized as follows:

1. TFconv layer with excellent interpretability is proposed, which can extract time-frequency information to obtain a better diagnostic performance.
2. All the inner-product-based TFT methods can be embedded into the TFconv layer as kernel functions, and three typical TFT methods are considered to formulate corresponding TFconv layers and further be analyzed.
3. Frequency response analysis is performed on the well-trained TFconv layer to explain the logical foundation of feature extraction and prediction-making of TFN in the frequency domain. Besides, experimental results demonstrate that the proposed TFconv layer has outstanding advantages in convergence speed, few-shot ability, and generalizability.

Table 1
The representative works related to our method.

Method	Application	Motivation
SincNet, 2019 [36]	Speaker recognition	**Parameterizing real** value convolutional kernel by **trainable sine** function
W-CNN, 2020 [37]	Discharge detection	**Initializing real** value convolutional kernel by **wavelet** function
WKN, 2022 [35]	Fault diagnosis	**Parameterizing real** value convolutional kernel by **trainable wavelet** function
DeSpaWN, 2022 [38]	Unsupervised monitoring	Designing a new layer to simulate **wavelet decomposition**
TFN (our method)	Fault diagnosis	**Parameterizing complex** value convolutional kernel to simulate **time-frequency transform** and **interpreting the attention that CNN paid to different frequencies**

The rest of this article is organized as follows. Section 2 introduces TFT and CNN as the theoretical foundation. Based on Section 2, the construction of the TFconv layer, the interpreting method, and the diagnosis procedure using TFN are presented in Section 3. In Section 4, three mechanical dataset experiments are carried out to verify the diagnostic performance and interpretability of TFN. Section 5 introduces the essential differences between TFNs and contrast models, and conducts further analyses of training time, convergence speed, few-shot ability and generalizability of proposed TFNs. Finally, conclusions are given in Section 6.

2. Preliminary

2.1. Time-frequency transform

Time-frequency transform (TFT), based on the inner product, is a signal processing method and is widely used for mechanical fault diagnosis. Mathematically, the purpose of the inner product is to measure the similarity, and any vector could be decomposed into a set of orthogonal bases by means of the inner product. Considering the 1D vibration signal as a vector, we can decompose the signal in the same way. Fourier transform, the most fundamental signal processing method, takes the frequency-orthogonal sine function as the basis and decomposes the vibration signal through the inner product to obtain the energy of the vibration signal at different frequencies, i.e., the frequency spectrum:

$$X(f) = \int_{-\infty}^{+\infty} x(t)(e^{j2\pi ft})^* dt = \left\langle x(t), e^{j2\pi ft} \right\rangle \tag{1}$$

where $X(f)$ denotes the frequency spectrum, $x(t)$ denotes the analyzed signal, $e^{j2\pi ft}$ denotes the trigonometric bases and $*$ denotes the conjugate operator.

Although the frequency spectrum obtained by the inner product with frequency-orthogonal bases can effectively reveal the frequency domain information of the vibration signal, it leaves out the time domain information and is not suitable for non-stationary signals. To process non-stationary signals, it is necessary to obtain the energy of the signal at different times and different frequencies, i.e., the time-frequency distribution.

The time-frequency distribution can be obtained by the inner product with time-frequency-orthogonal bases, which is formulated as

$$TF(\tau, f) = \left| \int_{-\infty}^{+\infty} x(t)\psi_f{}^*(t-\tau)dt \right| = \left| \left\langle x(t), \psi_f(t-\tau) \right\rangle \right| \tag{2}$$

where $TF(\tau, f)$ denotes the time-frequency distribution, $X(t)$ denotes the analyzed signal, $\left\langle \psi_f(t-\tau) \right\rangle_{f,t}$ denotes the time-frequency-orthogonal bases. $\psi_f(t)$ denotes the inner product window function that has compact support in both time domain and frequency domain.

The process of inner-product-based TFT is demonstrated in Fig. 1. The input signal $x(t)$ and the inner product window functions $\psi_f(t)$ with different frequency bands are convolved to obtain the frequency spectrum of the signal at a specific time point τ. With the movement of the time point τ, the complete time-frequency distribution $TF(\tau, f)$ is gradually obtained. Among the process, parameter τ and f are used to adjust the focusing area of inner product window function in time domain and frequency domain respectively. It should be noted that the inner product window function is a complex function, so the obtained time-frequency distribution is also complex, that is, the spectrum includes both the energy information and the phase information. In order to separate the energy information, the complex modulo operation of the time-frequency distribution is required.

TFT methods include short-time Fourier transform (STFT), chirplet transform (CT) [39,40], and wavelet transform (WT) [41], etc. All these methods follow the above procedure and the only difference between them is the inner product window function. STFT constructs an inner product window function by multiplying the sine function by a time domain window. CT introduces a linear frequency modulation factor to STFT, thereby improving the accuracy and stability of the result under variable speed conditions. WT uses the wavelet family, which is generated by the mother wavelet through scaling and translation, as the inner product window function, to obtain a time-frequency window that varies with frequency. WT has low time domain resolution and high frequency domain resolution at the low frequency, and vice versa at the high frequency. The inner product window functions of the above four TFT methods are shown in Table 2.

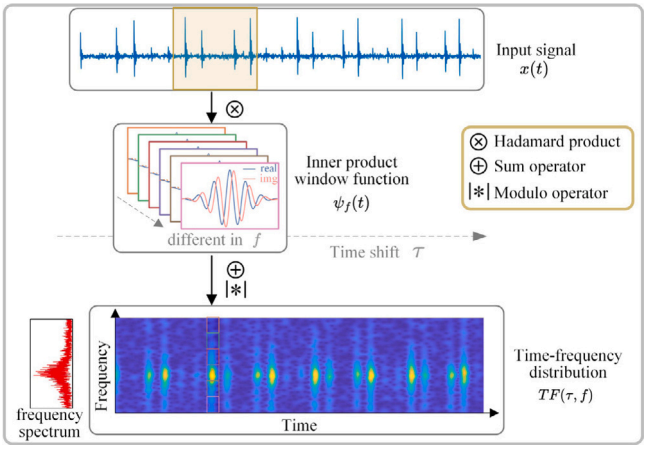


Fig. 1. The process of inner-product-based TFT.

Table 2
The inner product functions of time-frequency transform, the corresponding kernel functions of the tfconv layer and trainable parameter with their limits. (N_c denotes the channel number of TFconv layer.).

Kernel Function	Inner Product Function of Time-Frequency Transform	Kernel Function of TFconv Layer	Trainable Parameter θ with Its Limit
STTF	$\psi_{\omega,\sigma}(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2} \cdot e^{-j\omega t}$	$\psi_f[n] = e^{-\frac{1}{2}\left(\frac{n}{\sigma N_c}\right)^2} e^{j2\pi f n}$, $\sigma = 0.52$, $n = [-(N_c-1), \ldots, (N_c-1)]$	$f_0 \in [0, 0.5]$
Chirplet function	$\psi_{\omega,\alpha,\sigma}(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2} e^{-j\left[\frac{\alpha}{2}t^2+\omega t\right]}$	$\psi_{f,\alpha}[n] = e^{-\frac{1}{2}\left(\frac{n}{\sigma N_c}\right)^2} e^{-j2\pi\left[\frac{\alpha}{2}n^2+fn\right]}$, $\sigma = 0.52$, $n = [-(N_c-1), \ldots, (N_c-1)]$	$f \in [0, 0.5]$, $\lvert\alpha\rvert < 0.1/N_c$
Morlet wavelet	$\psi_s(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t}{s}\right)$, $\Psi(t) = Ae^{-\beta\frac{t^2}{2}} e^{j\omega t}$	$\psi_s[n] = \frac{1}{\sqrt{s}}\Psi\left(\frac{n}{s}\right)$, $\Psi(n) = e^{-\frac{1}{2}\left(\frac{n}{\sigma N_c}\right)^2} e^{j2\pi f_0 n}$, $\sigma = 0.6, f_0 = 0.2$ $n = [-10(N_c-1), \ldots, 10(N_c-1)]$	$s \in [0.4, 10]$

TFT methods can project the non-stationary signals from time domain into time-frequency domain to obtain their time-frequency distribution. As shown in Fig. 1, the input is a simulated signal of rolling bearing with inner fault, and its time-frequency distribution demonstrates both the vibration frequency and the dual-impulse property. Due to their powerful analytical abilities, TFT methods play an important role in the feature extraction of mechanical fault diagnosis.

2.2. CNN

The structure of a convolutional neural network (CNN) can be roughly divided into two parts: the convolutional part and the classification part. The convolutional part consists of a series of convolutional layers, BN layers, activation layers, and pooling layers, while the classification part consists of several fully connected layers. The input samples usually are 1D vibration signals, so the CNN models used for mechanical fault diagnosis are 1D accordingly.

The convolutional layer is the core part of the CNN and the process of 1D convolution is shown in Fig. 2. Each randomly initialized convolutional kernel is convolved along the 1D input signal, and the results of multiple convolutional kernels are concatenated to obtain the feature map. The feature map of the lth layer on the kth channel can be expressed as

$$h_k^l = w_k^l * x^l + b_k^l \tag{3}$$

where, x^l denotes the input of the lth convolutional layer, and w_k^l and b_k^l denote the weight and bias of the kth kernel in the lth convolutional layer respectively. The symbol $*$ denotes the convolutional operator.

The BN layer normalizes inputs to a specific distribution, thereby increasing the training speed and alleviating gradient explosion or disappearance. The output value of the kth channel in the lth BN layer can be given as

$$\hat{h}_k^l = \frac{h_k^l - \mu_k^l}{\sqrt{{\sigma_k^l}^2 + \varepsilon}} \tag{4}$$

where μ_k^l and σ_k^l denote the channel-wise mean and variance of the input h_k^l, and ε denotes a small constant that prevents the denominator from being zero.

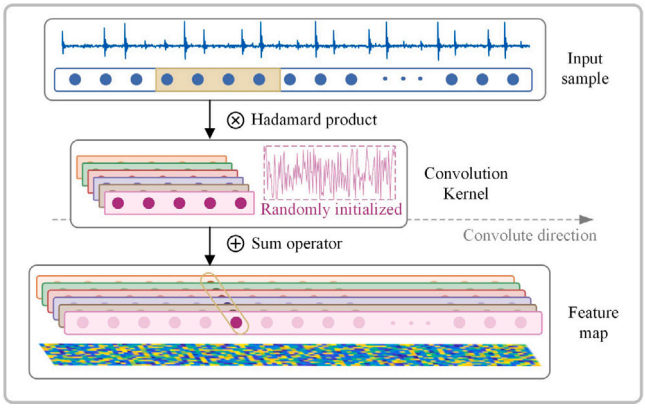


Fig. 2. The process of traditional convolutional layer in CNN.

The activation layer introduces nonlinearity into CNN, thereby enabling nonlinear mapping. The output of the lth activation layer can be denoted as

$$z^l = f\left(\hat{h}^l\right) \tag{5}$$

where $\hat{h}^l$ denotes the input of the activation layer and f denotes the nonlinear activation function, e.g. ReLU, Sigmoid.

With quite many features extracted, the pooling layer aims to compress them to reduce the model parameters while preserve the main information. The output of the lth pooling layer can be denoted as

$$y^l = \text{Downsample}\left(z^l\right) \tag{6}$$

where y^l denotes the input and Downsample denotes downsampling operation, such as maximum or average downsampling.

After the convolutional part, the raw signal has been transformed into high-dimensional features, which are flattened and further classified in the following classification part. Finally, the *Softmax* function is used to obtain the probability of different categories, and classification loss is calculated through cross-entropy, which can be given as

$$L(r, p) = -\sum_i r_i \log p_i \tag{7}$$

where i denotes the number of categories, r and p denotes the ground truth and predicted probability, respectively. The classification loss measures the discrepancy between the true label and the prediction. With the training strategy based on backpropagation (BP) and optimization algorithms like stochastic gradient descent (SGD), the prediction of the model could approach the true label gradually.

3. Methodology

In this section, we introduce the structure and the kernel function of TFconv layer, the interpreting method, and the TFN with its entire fault diagnosis procedure.

3.1. Structure of TFconv layer

The TFT methods are physically interpretable, but they can not extract adaptive time-frequency information based on the characteristics of the mechanical dataset. On the other hand, CNNs can automatically extract high-dimensional features from the original samples and make accurate classifications efficiently, but the logic of CNN's decision-making is not clear enough. To utilize the advantages of both two strategies, we embed the TFT method into the traditional convolutional layer since they can be both regarded as inner products, and this leads to a novel layer named the TFconv layer.

We designed the TFconv layer to simulate the physically interpretable TFT method, and the key point is to use a complex value convolutional kernel instead of the real value kernel used in the current literature [35–37]. Considering that most CNN models use real value variables and in order to get good compatibility, the structure of the TFconv layer is designed as Fig. 3. The TFconv layer consists of a real part kernel and an imaginary part kernel, which convolves with the mechanical vibration signals along the length direction to get real part features and imaginary part features, respectively. After that, the modulus of real and imaginary features is calculated (as traditional TFT method does) to obtain the final time-frequency distribution as the output of the TFconv layer, and each channel is processed independently in the whole process. Moreover, different from the traditional convolutional layer whose

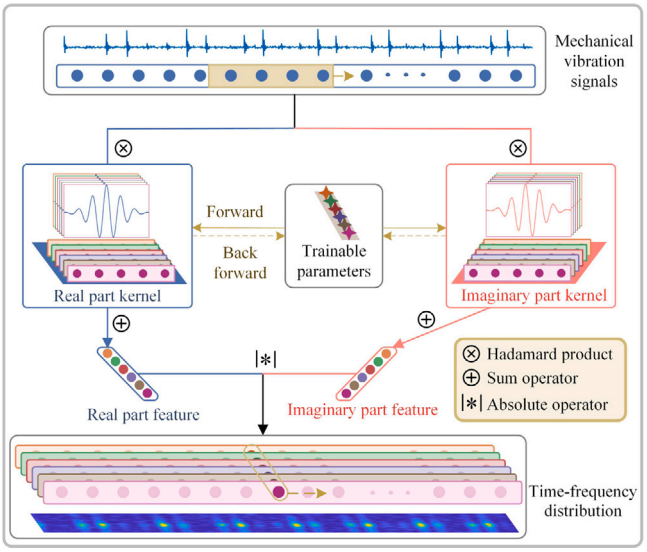


Fig. 3. The process of TFconv layer.

weights are randomly initialized, the weights of the real and imaginary part kernels are initialized and controlled be the kernel function, and their relationship can be expressed as

$$\begin{cases} \psi_\theta \in \mathbb{C}, \quad \psi_{\theta,\text{real}}, \psi_{\theta,\text{imag}} \in \mathbb{R} \\ \psi_{\theta,\text{real}} = \text{real}(\psi_\theta) \\ \psi_{\theta,\text{imag}} = \text{imag}(\psi_\theta) \end{cases} \tag{8}$$

where ψ_θ denotes the kernel function, $\psi_{\theta,\text{real}}$ and $\psi_{\theta,\text{imag}}$ are the real part kernel and the imaginary part kernel respectively, $\text{real}(\cdot)$ and $\text{imag}(\cdot)$ denote the operators to get the real part and the imaginary part of a complex value respectively. The control parameter θ, which adjusts the frequency property of kernel function, is treated as the trainable parameter and updated in the BP process. The kernel function of the TFconv layer is equivalent to the inner product window function in TFT, and the output of the kth channel of the TFconv layer can be denoted as

$$\begin{cases} h_{k,\text{real}} = \psi^k_{\theta,\text{real}} * x \\ h_{k,\text{imag}} = \psi^k_{\theta,\text{imag}} * x \\ h_k = \sqrt{h^2_{k,\text{real}} + h^2_{k,\text{imag}}} \end{cases} \tag{9}$$

where k denotes the kth channel of the convolutional kernel, x denotes the input of the TFconv layer, θ denotes the trainable control parameter of the kernel function, h_{real} and h_{imag} denote the real and imaginary feature map respectively, and h denotes the final output (i.e., time-frequency distribution).

Compared to traditional convolutional layers, the proposed TFconv layer has three novel aspects as follows:

1. Real-imaginary mechanism: TFconv layer has two convolutional processes of the real part kernel and imaginary part kernel, and their outputs are merged through modulo operation.
2. Kernel function: The weight of the convolutional kernel of TFconv layer is determined by the specific kernel function, not randomly initialized.
3. Trainable parameters: The trainable parameters of TFconv layer are the control parameters θ of the kernel function (e.g., frequency factor f in STFT kernel), instead of the convolutional weights.

Since the trainable parameters of the TFconv layer are different from those of the traditional convolutional layer, the BP process of the TFconv layer is different accordingly. In the BP process, the TFconv layer calculates the gradient of trainable parameters and updates them during each training step, which can be expressed as

$$\begin{cases} \delta_\theta = \dfrac{\partial L}{\partial h}\left(\dfrac{\partial h}{\partial \psi_{p,\text{real}}}\dfrac{\partial \psi_{p,\text{real}}}{\partial \theta} + \dfrac{\partial h}{\partial \psi_{p,\text{imag}}}\dfrac{\partial \psi_{p,\text{imag}}}{\partial \theta}\right) \\ \theta \leftarrow \text{optimizer}\left(\theta, \delta_\theta, \eta\right) \end{cases} \tag{10}$$

where θ denotes the trainable parameters of the kernel function, δ_θ denotes the gradient of θ, ∂ denotes the partial derivative operator, L denotes the classification loss, h denotes the output of the TFconv layer, $\psi_{p,\text{real}}$ and $\psi_{p,\text{imag}}$ denote the weights of the real and imaginary parts, η denotes the leraning rate of the optimizer. After the gradient is obtained by the chain rule, the trainable parameters could be updated by optimization algorithms like SGD, Adam or RMSprop.

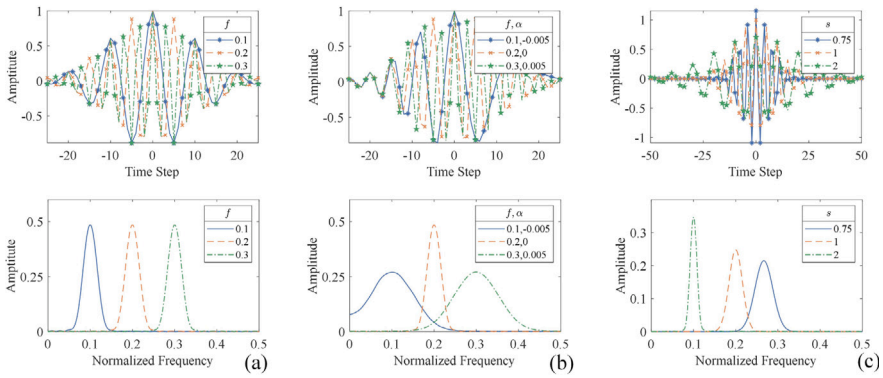


Fig. 4. The time-domain and frequency-domain diagrams of three kernel functions of TFconv layer. (a) STTF. (b) Chirplet function. (c) Morlet wavelet.

3.2. Kernel function of TFconv layer

The kernel function of the TFconv layer derives from the inner product window function of TFT through necessary discretization and modification, which is the key step to embed the TFT method into the TFconv layer. With the TFT method embedded, physical constraints of TFT are also introduced in the TFconv layer, that is, there is a certain limit on the trainable parameters of the kernel function. Taking STFT as an example, due to the *nyquist* sampling theorem, the meaningful normalized frequency is $[0, 0.5]$, so the frequency factor f of the corresponding kernel function should be limited to $[0, 0.5]$ as well.

According to the above analysis, three typical TFT methods (i.e., STFT, CT, and Morlet WT) are considered to formulate TFconv kernel functions. The inner product window functions of TFT, the corresponding kernel functions of the TFconv layer and trainable parameters with their limits are shown in Table 2. Different from the short-time trigonometric function (STTF) and chirplet function, the morlet wavelet kernel can stretch and contract in the time domain by scale factor s. To avoid time-domain truncation, the kernel length of the morlet wavelet kernel is designed to be much longer than that of the STTF and chirplet kernel.

To illustrate the properties of the above three kernel functions, the time-domain and frequency-domain diagrams of them are shown in Fig. 4, and all the kernel functions can be regarded as bandpass filters. STTF kernel function has a fixed frequency bandwidth, and the center frequency is adjusted by the frequency factor f. Chirplet kernel function introduces a linear frequency modulation factor α to the STTF kernel function and can dynamically change its filtering bandwidth. Morlet wavelet kernel function can scale their mother wavelet by scaling factor s to adjust their frequency properties, and with the increase of scaling factor, it has a higher center frequency and wider bandwidth.

3.3. Interpretability of TFconv layer

With the TFT method embedded, the TFconv layer can not only extract time-frequency information from the mechanical vibration signals, but also become explainable through the interpreting method proposed in this subsection. The interpreting method is to perform the frequency response analysis on the trained TFconv layer to obtain its amplitude–frequency response (FR), which indicates the attention that CNN paid to different frequencies. The frequencies with higher attention could easily pass through the TFconv layer and are deeply involved in the prediction-making process and responsible for the results of CNN.

The frequency response analysis of the convolutional layer derives from the filter theory in the signal processing field. The convolutional process in the convolutional layer is exactly equivalent to the filtering process of the finite impulse response (FIR) filter [42]. Therefore, the convolutional kernel can be regarded as a FIR filter, and the input is the mechanical vibration signals to be filtered. In the frequency response analysis, the convolutional kernel is processed by fast Fourier transform (FFT) to obtain the FR of the convolutional layer [43]. Considering multiple channels contained in the convolutional layer, we calculate the channel-wise amplitude–frequency response (C-FR) first, and average them to obtain the overall amplitude–frequency response (O-FR). The calculation process could be denoted as

$$\begin{aligned} H_i(f) &= \left|\mathrm{FFT}(w_i)\right| = \left|\sum_{n=0}^{N} w_i[n] e^{-j\frac{2\pi f n}{N}}\right| \\ H(f) &= \frac{1}{n_c}\sum_{i}^{n_c} H_i(f) \end{aligned} \tag{11}$$

where, w_i denotes the convolutional kernel of the ith channel, $H_i(f)$ and $H(f)$ denote the C-FR of the ith channel and the O-FR, respectively. Although the above frequency response analysis is based on the traditional convolutional layer, it is also applicable to the TFconv layer because the real part kernel and imaginary part kernel in the TFconv layer are equivalent to a complex value kernel.

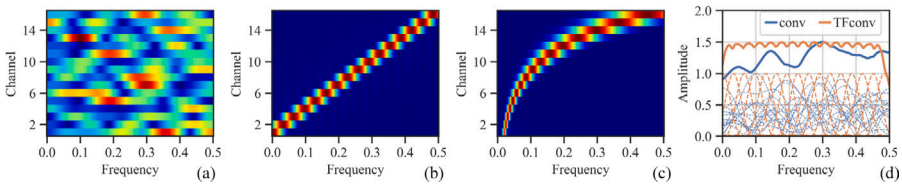


Fig. 5. The comparison between C-FR and O-FR of initialized traditional convolutional layer and that of initialized TFconv layers. (a) C-FR of traditional convolutional layer. (b) C-FR of TFconv layer with STTF kernel. (c) C-FR of TFconv layer with Morlet wavelet kernel. (d) O-FR of traditional convolutional layer and TFconv layer with STFT kernel (solid line represents O-FR while dashed line represents C-FR).

Table 3
The Architecture of the analyzed TFN.

Part	No. Unit	Basic Unit	Output Size
TFconv	–	input	1*1024
	1	TFconv(n_c@N*1)	n_c*1024
Backbone	2	Conv(16@15*1)-BN-ReLU	16*1010
	3	Conv(32@3*1)-BN-ReLU-MaxPool(2)	32*504
	4	Conv(64@3*1)-BN-ReLU	64*502
	5	Conv(128@3*1)-BN-ReLU-AdaptivePool(4)	128*4
	6	Flatten	512
	7	Dense(256)-ReLU-Dense(64)-ReLU-Dense(n_p)	n_p

Both the traditional convolutional layer and the TFconv layer can be conducted by frequency response analysis, yet there still exist some differences in their results. The C-FR of the traditional convolutional layer, shown in Fig. 5(a), presents a random distribution, which is difficult to identify the focusing frequency area (i.e., the passband frequency of the FIR filter). While the TFconv layer, shown in Fig. 5(b) and (c), is well controlled by the kernel function and has a clear frequency preference. The O-FR of the traditional convolutional layer and TFconv layer are shown in Fig. 5(d). Compared with the traditional convolutional layer, the focusing frequency area of the TFconv layer can be identified more easily and used for interpreting the frequency foundation of CNN models.

Moreover, it is necessary to further elaborate on the interpreting logic of the TFconv layer. TFconv layer can be regarded as a series of trainable bandpass FIR filters, without considering the modulo operation. When the TFconv layer is combined with a backbone CNN as a preprocessing layer, the TFconv layer becomes the data portal of the backbone CNN. This means that only the frequency with amplitude in the amplitude–frequency response (O-FR) can pass the TFconv layer and be used by the subsequent backbone CNN for fault diagnosis tasks. Therefore, we claim that our TFN models have interpretability because the overall amplitude–frequency response (O-FR) of the TFconv layer could be used to explain the attention that the CNN model paid to different frequencies. To validate this, we propose a hypothesis that, the information frequency bands (which carry most information of the mechanical dataset) are crucial for fault diagnosis, and TFN models will pay more attention (i.e., amplitude peaks in O-FR) to these information frequency bands during the training process to achieve better diagnostic performance. We will verify the above hypothesis in the following experiment section, and if the O-FRs of trained TFN models have a good correspondence with the information frequency bands of the dataset frequency spectrum, the correctness of the interpreting logic of proposed TFN models would be effectively confirmed.

3.4. Fault diagnosis using TFN

The interpretable TFconv layer is used as the preprocessing layer to combine with a backbone CNN, and the resulting novel network is named the time-frequency network (TFN). With TFN, we can extract fault-related time-frequency information from raw vibration signals and diagnose the fault states of mechanical equipment efficiently. The entire framework of applying TFN to intelligent mechanical fault diagnosis is summarized in Fig. 6.

Firstly, the vibration signals are collected by the accelerometers installed on the mechanical equipment. Secondly, the vibration signal is truncated with a sliding window to generate a series of samples as the input of TFN. Thirdly, the kernel function is formulated from TFT methods and embedded into the TFconv layer. Then, an existing CNN model is selected as the backbone, and the TFconv layer is combined as the preprocessing layer with the backbone to obtain TFN. After that, the obtained TFN is trained and verified by training samples and test samples in fault diagnosis tasks. Finally, the interpreting method is conducted to reveal the focusing frequency area of TFN.

In order to verify the effectiveness of the TFconv layer, a relatively shallow CNN is selected as the backbone, and the architecture of the obtained TFN is shown in Table 3. n_c, N and n_p are the channel number of the preprocessing layer, the length of the TFconv layer, and the number of categories for classification, respectively. n_c and N are determined in the design process of the TFconv layer, and n_p is determined by the diagnostic task.

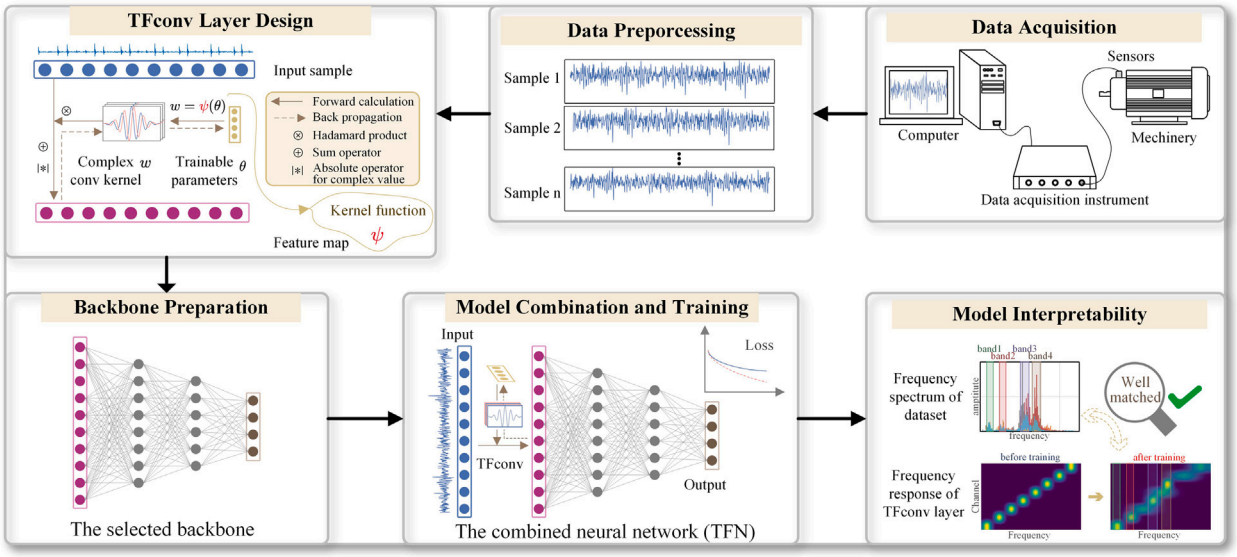


Fig. 6. The entire framework of applying TFN to intelligent mechanical fault diagnosis.

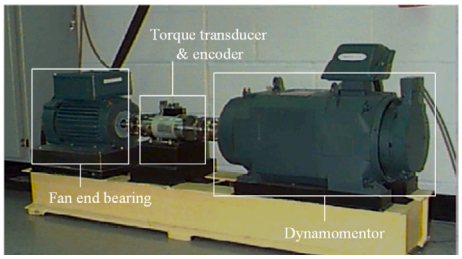


Fig. 7. CWRU bearing experimental system.

4. Experiment

In this section, three experimental datasets are used to verify the diagnostic performance and the interpretability of TFNs. In the diagnostic performance part, considering the importance of the channel numbers, we compare TFN models with other models mentioned in Table 1 under different channel numbers. In the interpretability part, the consistency between the frequency spectrum of the dataset and the O-FRs of trained TFconv layers with different kernels are analyzed to verify the interpretability of TFNs.

4.1. CWRU public bearing dataset

In order to better spread our work, we choose this open-source dataset as a benchmark to test the diagnostic performance of TFNs. The CWRU bearing dataset [44], as shown in Fig. 7, is one of the most popular open-source datasets for mechanical fault diagnosis. The accelerometer is installed on the motor casing of the drive end, and the vibration signals are collected under four loading conditions (load 0–3 HP) at two sampling frequencies of 12 kHz and 48 kHz.

In addition to normal condition (N), three different bearing fault types are contained in this dataset: inner raceway fault (I), rolling element fault (B), and outer raceway fault (O). For each fault type, different fault sizes were considered respectively, i.e., 0.007, 0.014 and 0.021 inches. Therefore, this dataset has nine fault states and a normal state, ten categories in total. The fault diagnosis of CWRU rolling bearing can be regarded as a 10-class classification.

The diagnostic difficulty of the CWRU dataset is relatively low, and the diagnostic accuracy of 12 kHz signals of most models is close to 100%, which is hard to distinguish the diagnostic performance of different models. Therefore, 48 kHz vibration signals are chosen for the following diagnosis task for its higher diagnostic difficulty. The vibration signal is truncated into samples at the length of 1024 and each state has 450 samples, for a total of 4500 samples. 60% of samples in each category are used for training, and the remaining samples are used for testing. The loss function of classification is cross-entropy, the training optimizer is Adam, the momentum is set to 0.9, the initial learning rate is 0.001, the decay ratio of the learning rate is 0.99 per epoch, and the training epoch is 50. Each model is repeated 10 times to eliminate randomness and verify the stability.

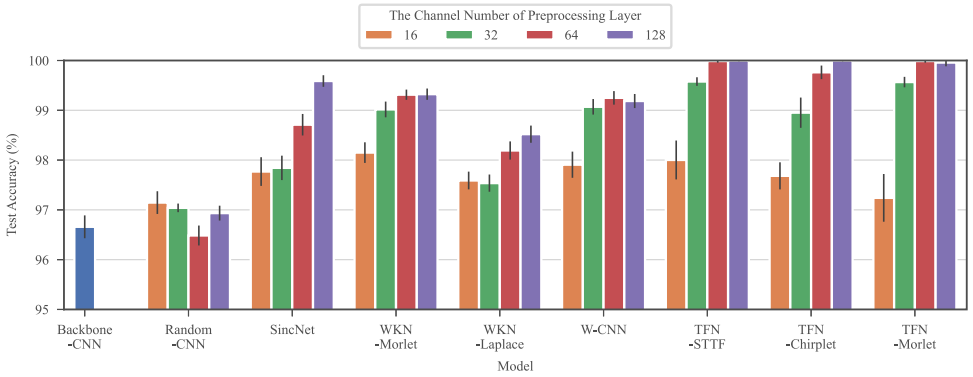


Fig. 8. Test accuracy on the CWRU bearing dataset.

To test the diagnostic performance of the TFconv layer, this experiment contains three types of models. 1) Backbone models: the backbone CNN (denoted as Backbone-CNN), the backbone CNN with traditional randomly-initialized kernel convolutional layer as the preprocessing layer (denoted as Random-CNN). 2) Contrast models: SincNet [36], WKN with morlet kernel and laplace kernel(denoted as WKN-Morlet and WKN-Laplace) [35], W-CNN [37]. 3) TFN models: TFNs with STTF, chirplet function, and morlet wavelet function as the kernel function (denoted as TFN-STTF, TFN-Chirplet, and TFN-Morlet, respectively). Except for the backbone CNN, each type of other models contains 4 different channel numbers of preprocessing layer: 16, 32, 64 and 128.

The fault diagnosis results on the CWRU bearing dataset are shown in Fig. 8, and we can draw the following conclusions.

1. The diagnostic accuracies of Backbone-CNN and Random-CNN are the worst. SincNet, WKN-Morlet, WKN-Laplace, and W-CNN have made great improvements compared to Backbone-CNN. Furthermore, TFN-STTF, TFN-Chirplet and TFN-Morlet reach the best diagnostic performance overall, and their diagnostic accuracy under 64 or 128 channel numbers is close to 100%, demonstrating the effectiveness of TFN in fault diagnosis. The TFconv layer of TFN can transform the raw vibration signal into fault-related time-frequency features, leading to the distinguished diagnostic performance of TFNs.
2. The more channels generally led to the higher diagnostic accuracy and this phenomenon is more obvious in TFN models. The diagnostic accuracy of TFN models and other models are similar when the channel number is 16, but with the increase of the channel number, the diagnostic accuracies of TFN models are significantly higher than that of other methods. TFconv layer with more channels could extract more time-frequency information and lead to higher diagnostic accuracy. However, the diagnostic accuracy of TFN with 64 channels is similar to that with 128 channels. This indicates that when the TFconv layer has enough channels to extract time-frequency information, increasing the number of channels does not bring a corresponding increase in diagnostic accuracy.
3. As for the kernel functions of TFN, all three kernel functions have similar diagnostic performance, and the STTF kernel is slightly better than the chirplet kernel and morlet kernel in general.

After introducing the diagnostic performance of TFN, here comes the interpretability analysis part. To have a suitable dataset frequency spectrum, the vibration signals at 12 kHz sampling frequency under 3 HP loading condition in the CWRU dataset are used as the input samples, and TFNs with different kernel functions are trained by them. In order to obtain a clear interpretable observation, the channel numbers of preprocessing layer are set to 8 and other experimental settings are the same as those in the previous diagnostic experiments.

According to frequency response analysis shown in Eq. (11), the O-FR of the preprocessing layer of different well-trained models could be obtained. The frequency spectrum of CWRU dataset and the O-FR results are shown in Fig. 9. As shown in Fig. 9(a), the frequency amplitude of the dataset mainly exists in four information frequency bands, i.e., bands 1-2-3-4, which carry most information of the dataset and have been marked out by different colors. These information bands are responsible for fault diagnosis and a good CNN model should focus on them to achieve a satisfying diagnostic accuracy. From the O-FR results shown in Fig. 9(b)–(i), the following conclusions could be obtained.

1. The O-FRs of trained Backbone-CNN and Random-CNN have an amplitude increase only at band 2 or band 4, indicating that the information is insufficiently extracted for fault diagnosis.
2. The O-FRs of contrast models (i.e., SincNet, WKN-Morlet and W-CNN) are shown in Fig. 9(d)–(f). The O-FRs of SincNet and WKN-Morlet do not change much after training, and their results do not correspond exactly to the frequency spectrum of the CWRU dataset shown in Fig. 9(a). Unlike the previous two methods, W-CNN is only initialized by kernel functions rather than parameterized, so its O-FR changes significantly and matches well with the dataset frequency spectrum. However, because it is not constrained by kernel function during training and is only a real kernel, its O-FR is relatively chaotic and there exists a redundant peak in the high-frequency area, so W-CNN is not a very ideal way to explain what is the fault-related frequencies in model's view.

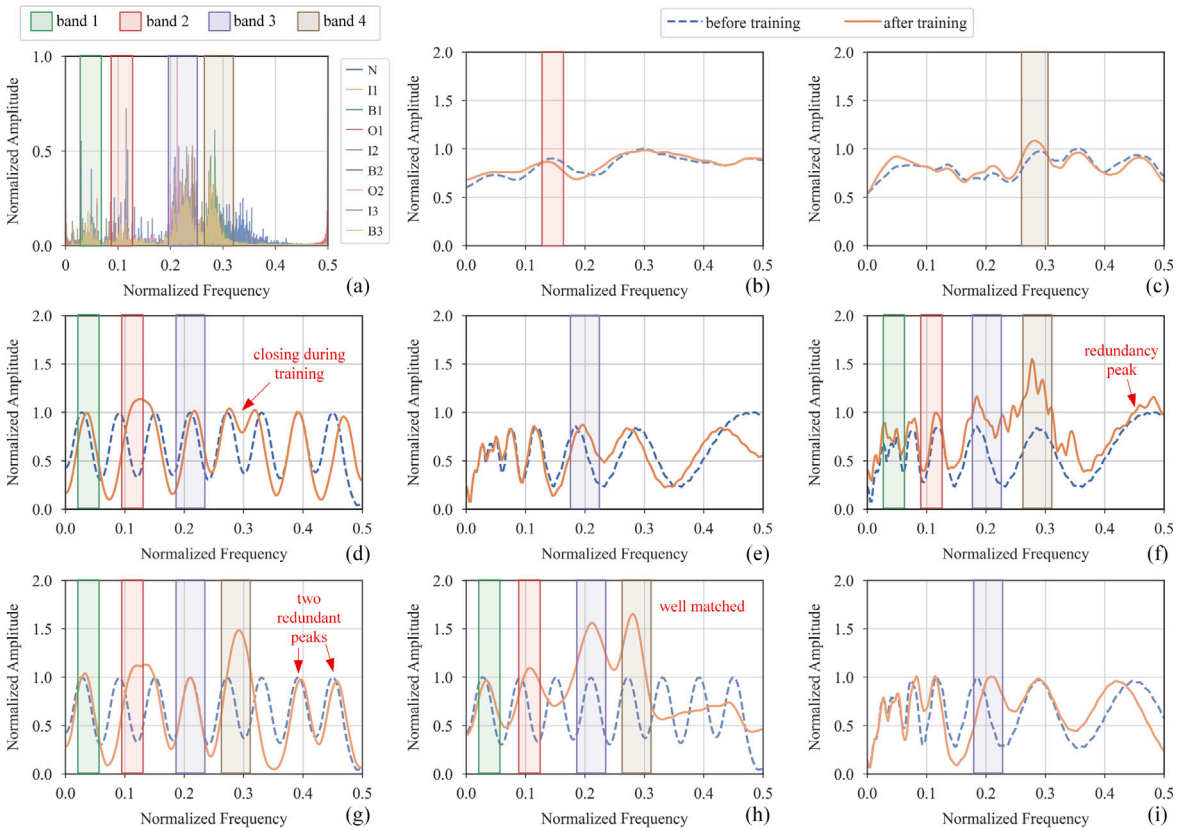


Fig. 9. Frequency spectrum of the CWRU bearing dataset and O-FRs of different models. (a) Frequency spectrum of CWRU dataset. (b) O-FR of the first convolutional layer of Backbone-CNN. (c) O-FR of the first convolutional layer of Random-CNN. (d) O-FR of SincNet. (e) O-FR of WKN-Morlet. (f) O-FR of W-CNN. (g) O-FR of the TFconv layer of TFN-STTF. (h) O-FR of the TFconv layer of TFN-Chirplet. (i) O-FR of the TFconv layer of TFN-Morlet.

3. The O-FR of trained TFN-STTF has amplitude peaks at all the information frequency bands. It means that TFN-STTF pays correct attention to these information bands, which is consistent with its outstanding performance in fault diagnosis. However, since the parameter f of the STTF kernel function can only change the centering frequency, not the bandwidth, there still exist some amplitude peaks outside the information bands, i.e., two redundant peaks marked out in Fig. 9(g).
4. The O-FR of trained TFN-Chirplet has amplitude peaks within all the information frequency bands as well. However, different from the STTF kernel function, the chirplet kernel function has an additional linear frequency modulation factor α to change its filtering bandwidth, and it would increase its bandwidth to search a wider frequency band when this channel cannot get any useful information, so there exists no amplitude peak outside the information bands in the OF-R of TFN-Chirplet. The O-FR of TFN-Chirplet is completely consistent with the frequency spectrum of the CWRU dataset, so TFN-Chirplet achieves the best physical interpretability than other models.
5. The O-FR of TFN-Morlet changes little after training, where only one amplitude peak (corresponding to band 3) can be barely identified. Although the adaptive frequency bandwidth of the wavelet kernel function can help to extract fault-related information, it also blurs the focusing frequency bands and thus prevents them from being identified. This leads to the inferior interpretability of TFN-Morlet compared with TFN-STTF and TFN-Chirplet.

By analyzing the O-FR of the trained TFconv layer, we confirm the excellent performance of the proposed TFNs in terms of interpretability. The O-FR of trained TFN-Chirplet is well matched with the frequency spectrum of the CWRU dataset, verifying our hypothesis that the O-FR of TFconv layer reflects the attention of models to different frequencies and models tend to pay more attention to fault-related frequencies after the training process.

4.2. Planetary gearbox dataset

As shown in Fig. 10(a), this planetary gearbox system includes an electric motor, a transmission shaft, a torque transducer, a planetary gearbox, a magnetic powder brake and a series of sensors. The vibration signals are collected by the accelerometer located at the shell of the planetary gearbox and transmitted to the signal acquisition card at the sampling frequency of 10.2 kHz.

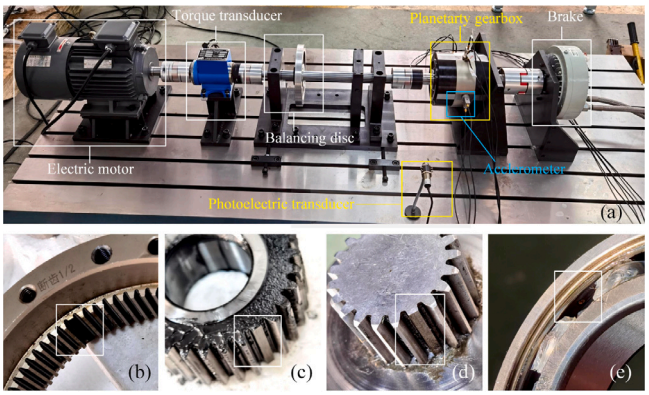


Fig. 10. The planetary gearbox system and the defective components. (a) The planetary gearbox system. (b) Tooth fracture of the ring gear. (c) Tooth fracture of the sun wheel. (d) Tooth fracture of the planetary gear. (e) Outer race pitting of the rolling bearing.

Table 4
The working conditions of gearbox system.

Label	Failure Component	Training/Testing Sample
N	None	264/176
S	Ring Gear	264/176
D	Ring Gear and Sun Wheel	264/176
T	Ring Gear, Sun Wheel and Planetary Gear	264/176
C	Ring Gear and Rolling Bearing	264/176

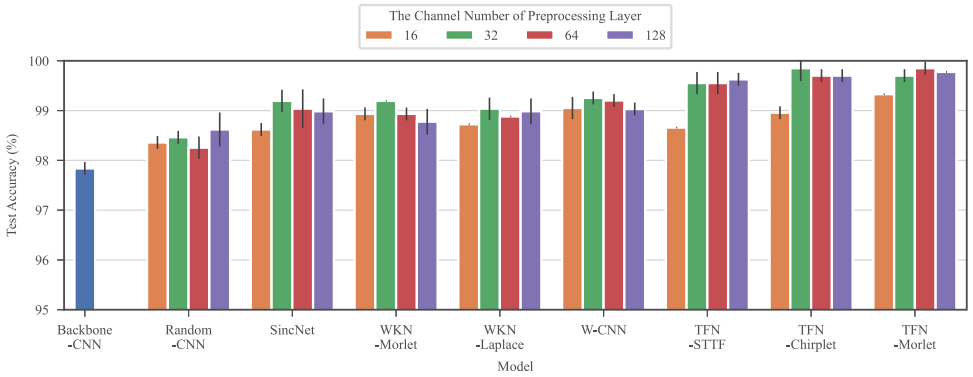


Fig. 11. Test accuracy on the planetary gearbox dataset.

Four types of component failures are considered in this dataset, including tooth fractures of three different gears and the outer race pitting of the rolling bearing as shown in Fig. 10(b)–(e). Based on these faults, the experiments are conducted under five health conditions listed in Table 4, specifically normal state (N), single-point fault (S), double-point fault (D), three-point fault (T) and compound fault (C). The fault diagnosis of the planetary gearbox can be regarded as a 5-class classification task.

In data preparation, the raw vibration signal is truncated without overlap through a sliding window to obtain input samples. Each category contains 440 samples, and the length of each sample is set to 1024. After that, 60% of the samples are randomly divided as the training set, and the rest samples are used as the test set. In addition, Gaussian white noise with signal-to-noise ratio (SNR) of 0 is added to the original signal to increase the difficulty of diagnosis. The rest of the experimental settings are consistent with that of the diagnostic experiment on the CWRU dataset.

The diagnostic results are shown in Fig. 11. The diagnostic difficulty of this dataset is relatively low, and the diagnostic performance gap of different models is not as obvious as in the previous CWRU diagnostic experiment. The diagnostic accuracy of Backbone-CNN is 97.8%, and Random-TFN performs better than the backbone CNN. SincNet, WKN-Morlet, WKN-Laplace, and W-CNN perform slightly better than the previous two models. TFN-STTF, TFN-Chirplet and TFN-Morlet achieve the best overall diagnostic performance, demonstrating the effectiveness of the proposed method. Furthermore, TFNs with 32 channels perform much better than that with 16 channels, but TFNs with more channels (i.e., 64, 128) do not have a corresponding increase in accuracy. It indicates that 32 channels are enough for TFN to extract sufficient Time-Frequency information on this planetary gearbox dataset.

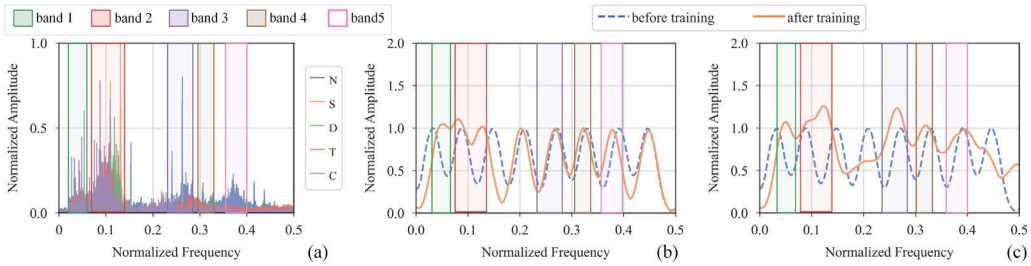


Fig. 12. Frequency spectrum of the planetary gearbox dataset and O-FRs of different models. (a) Frequency spectrum. (b) O-FR of the TFconv layer of TFN-STTF. (c) O-FR of the TFconv layer of TFN-Chirplet.

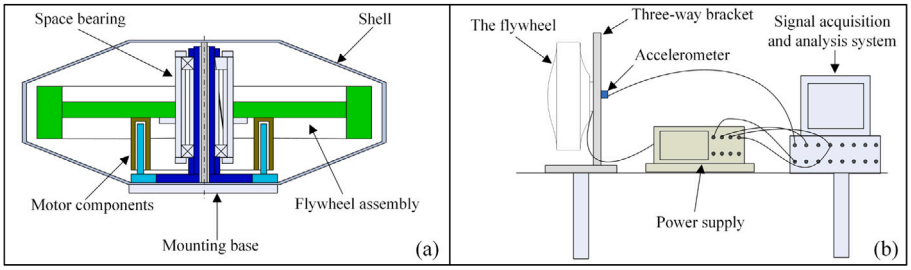


Fig. 13. The configuration of aerospace bearing experimental system. (a) Structure of the flywheel. (b) Schematic of the flywheel vibration acquisition process.

As for the interpretability analysis on the planetary gearbox dataset, considering the inferior interpretability performance of backbone models and TFN-Morlet, we only present the interpretability results of TFN-STTF and TFN-Chirplet for brevity. The channel numbers of the preprocessing layer are set to 8 and other experimental settings are the same as those in the previous diagnostic experiment on the planetary gearbox dataset. The frequency spectrum of the planetary gearbox dataset and the O-FRs of TFN-STTF and TFN-Morlet are shown in Fig. 12.

As marked in Fig. 12(a), the frequency amplitude mainly exists in five information frequency bands that contain most information of the planetary gearbox dataset, i.e., bands 1-2-3-4-5. As shown in Fig. 12(b)–(c), both the O-FRs of trained TFN-STTF and trained TFN-Chirplet have amplitude peaks at these five information frequency bands, showing that TFN-STTF and TFN-Chirplet pay correct attention to the information frequency bands of planetary gearbox dataset. However, the chirplet kernel function has an additional linear frequency modulation factor α for adjusting its filtering bandwidth compared to the STTF kernel function, so the O-FR of TFN-STTF has two amplitude peaks outside the information frequency band (one is nearly 0.1 in frequency and the other is nearly 0.45 in frequency), while the O-FR of TFN-Chirplet has no unrelated amplitude peaks and is more consistent with the frequency spectrum of the planetary gearbox dataset. In conclusion, this phenomenon in the planetary gearbox dataset is the same as that of the CWRU bearing dataset, demonstrating the outstanding interpretability of TFN-Chirplet to reveal the focusing frequency area of CNN models.

4.3. Aerospace bearing dataset

The former two datasets are collected in laboratory scenarios, while this aerospace bearing dataset is derived from the industrial scenario. As shown in Fig. 13, the aerospace bearing is the core component of the flywheel test rig, which consists of motor components, an aerospace bearing, a flywheel assembly, a shell, and a mounting base as shown in Fig. 13(a). The flywheel is driven by an electric motor and then puts the aerospace bearing to work. The data acquisition equipment is shown in Fig. 13(b), including an accelerometer, power supply, a signal acquisition and analysis system. The flywheel is fixed on an upright bracket, where the acceleration sensor collects the three-way vibration signals of the aerospace bearing at the sampling frequency of 25.6 kHz.

The aerospace bearing dataset has five types of health conditions: normal state (N), leading surface scratching (S), cage fault (C), ball fault (B), and inner ring fault (I). For each condition, the radial vibration signal is truncated into samples, and each category contains 1000 samples, for a total of 5000 samples. Then 60% of the samples in each category are used for training, and the remaining samples are used for testing. The fault diagnosis of the aerospace bearing dataset can be regarded as a 5-class classification task. Like the processing of the planetary gearbox dataset, the Gaussian noise of SNR = 0 is added to the original signal to increase the difficulty of diagnosis. The rest settings for this experiment are consistent with that of the previous planetary gearbox experiment.

The experimental results are shown in Fig. 14. The performance of the Backbone CNN (87.3%) is the worst and Random-TFN (around 93%) performs much better. This may be caused by the increased model depth of the newly added convolutional layer, even it is randomly initialized and updated through BP process. The contrast type models (including SincNet, WKN-Morlet, WKN-Laplace, W-CNN) is close or slightly better than Random-CNN, and the diagnostic accuracy of SincNet with 128 channels is 96.7%. As for

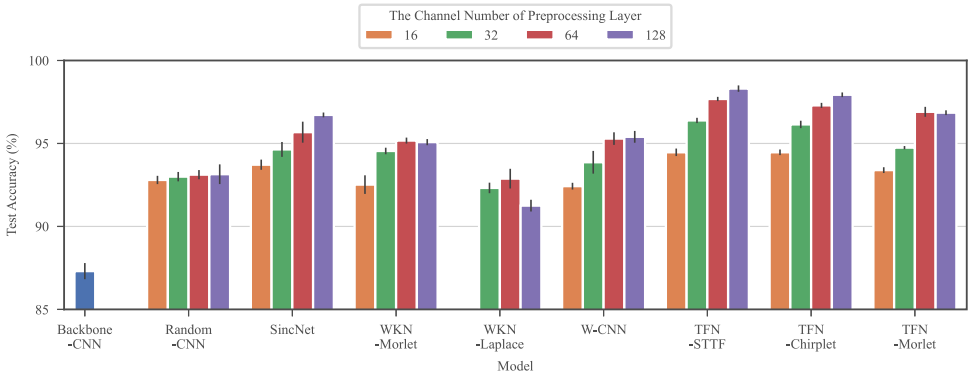


Fig. 14. Test accuracy on the aerospace bearing dataset.

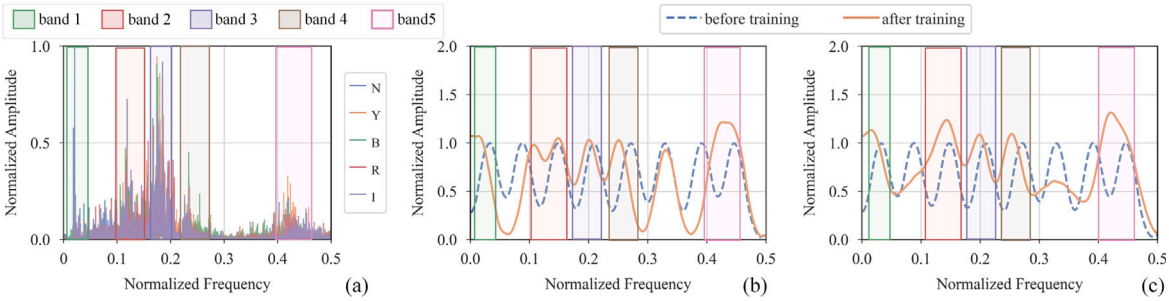


Fig. 15. Frequency spectrum of the aerospace bearing dataset and O-FRs of different models. (a) Frequency spectrum. (b) O-FR of the TFconv layer of TFN-STTF. (c) O-FR of the TFconv layer of TFN-Chirplet.

TFNs, TFN-STTF, TFN-Chirplet, and TFN-Morlet have best performances on diagnostic accuracy, where STTF-TFN with 128 channels achieves the highest average accuracy of 98.3%. TFconv layer with more channels could extract more time-frequency information and lead to a higher diagnostic accuracy despite the expense of training time. In conclusion, proposed TFNs have a much better performance than both backbone models and contrast models, and the diagnostic accuracy of TFNs increase significantly with the increase of the number of channels, demonstrating the importance of the number of channels.

As for the interpretability analysis, the process on the aerospace bearing dataset is the same as the previous two datasets. Specifically, TFN-STTF and TFN-Morlet with 8 preprocessing layer channels are chosen to be trained on the aerospace bearing dataset, and the experiment settings are the same as the diagnostic experiment before. The frequency spectrum of the aerospace bearing dataset and the O-FRs of TFN-STTF and TFN-Morlet are shown in Fig. 15.

As marked in Fig. 15(a), the frequency amplitude mainly exists in five information frequency bands that contain most information of aerospace bearing dataset, i.e., band 1-2-3-4-5. As shown in Fig. 15(b)–(c), the results are consistent with the interpretability analysis of the previous two datasets. Both two models pay correct attention to the information frequency bands of the aerospace bearing dataset, but the O-FR of TFN-STTF has one amplitude peak (nearly 0.35 in frequency) outside the information frequency bands, while that of TFN-Chirplet is pretty flat due to its capability to adjust filtering bandwidth. In conclusion, the O-FR of trained TFN-Chirplet has a good correspondence with the frequency spectrum of aerospace bearing dataset, demonstrating the outstanding interpretability of TFN-Chirplet again.

Based on the above experiments of diagnostic performance and interpretability, we can make an overall evaluation of these three kernel functions. In conclusion, although the morlet kernel has good diagnostic performance, it performs relatively poorly in interpretability. Therefore, we recommend using the STTF kernel or chirplet kernel in practice, where the STTF kernel has the advantage in diagnostic accuracy while the chirplet kernel is more suitable for interpretability.

5. Discussion

To further analyze the properties of TFN, the comparison with contrast models, the analysis of the training process and training time, the few-shot analysis, and the generalizability analysis are conducted on the CWRU open-source dataset. The comparison part shows the essential differences between TFN and contrast models and explains the reason for the superior diagnostic performance of TFNs. The analysis of the training process and training time discusses the advantage of TFN in convergence speed in the training process and the time cost of TFN. The few-shot analysis explores the relationship between the diagnostic accuracy of TFN and the number of training samples, demonstrating the excellent few-shot ability of TFN. The generalizability analysis proves the feasibility of generalizing the TFconv layer to other CNN models.

5.1. Comparison with contrast models

The motivation of contrast models (i.e., SincNet [36], WKN [35], and W-CNN [37]) shown in Table 1 is to parameterize or initialize traditional convolutional kernel by a specific kernel function, but these models only consider real value convolutional kernels and are equivalent to a series of bandpass FIR filters, whose outputs are filtered sub-signals. On the contrary, the TFconv layer uses real-imaginary mechanism to simulate complex value convolutional kernels, thus the output of the TFconv layer is time-frequency distribution (the energy distribution of a signal in the time-frequency domain), which is the same as traditional time-frequency transform methods do, except the TFconv layer is trainable and adaptive based on the specific dataset.

We demonstrate the differences between TFNs and contrast models by the formula derivation and the output presentation of these models. As for the formula derivation part, we would show the difference between the output of "complex kernel" (used by TFNs) and that of "real kernel only" (used by contrast models). We take STFT, a basic time-frequency transform method, for example. Given an input signal $x(t)$, the process of "complex kernel" could be denoted as

$$X(\tau,\omega)=\int x(t)w(t-\tau)e^{-i\omega t}\,\mathrm{d}t \tag{12}$$

where, $X(\tau,\omega)$ is the output of the process of "complex kernel", τ and ω are time shift parameter and frequency shift parameter, respectively. $w(t)$ is the window function of STFT, whose length is denoted as T. The process of "real kernel only" could be denoted as

$$\widehat{X(\tau,\omega)}=\int x(t)w(t-\tau)\cos\omega t\,\mathrm{d}t \tag{13}$$

According to Fourier expansion, $x(t)w(t-\tau)$ could be expanded as

$$x(t)w(t-\tau)=\frac{a_0(\tau)}{2}+\sum_{n=1}^{\infty}a_n(\tau)\sin\left[n\omega_0 t+\phi_n(\tau)\right] \tag{14}$$

where, $a_n(\tau)$ denotes the amplitude of the input signal $x(t)$ at time τ and frequency $n\omega_0$, and $\omega_0=2\pi/\,T$ is the basic frequency.

Therefore, bringing Eq. (14) back to Eq. (12) with modulus operator and considering the orthogonality of the sine function, we could get

$$\begin{aligned}|X(\tau,\omega)|&=\left|\int x(t)w(t-\tau)e^{-i\omega t}\,\mathrm{d}t\right|\\&=\left|\int\left\{\frac{a_0(\tau)}{2}+\sum_{n=1}^{\infty}a_n(\tau)\sin\left[n\omega_0 t+\phi_n(\tau)\right]\right\}e^{-i\omega t}\,\mathrm{d}t\right|\\&=\left|\frac{T}{2}\cdot\left[a_\omega(\tau)\sin\left(\phi_\omega(\tau)\right)+a_\omega(\tau)\cos\left(\phi_\omega(\tau)\right)\cdot i\right]\right|\\&=\frac{T}{2}a_\omega(\tau)\end{aligned} \tag{15}$$

Above is the output of the process of "complex kernel", and $a_\omega(\tau)$ is the amplitude of the input signal x(t) at time τ and frequency ω, i.e., the time-frequency distribution.

To get the output of the process of "real kernel only", it is the same to bring Eq. (14) back to Eq. (13) and we will get

$$\begin{aligned}\widehat{X(\tau,\omega)}&=\int x(t)w(t-\tau)\cos(\omega t)dt\\&=\int\left\{\frac{a_0(\tau)}{2}+\sum_{n=1}^{\infty}a_n(\tau)\sin\left[n\omega_0 t+\phi_n(\tau)\right]\right\}\cos(\omega t)dt\\&=\frac{T}{2}a_\omega(\tau)\cdot\sin\phi_\omega(\tau)\end{aligned} \tag{16}$$

The output of "real kernel only" has two parts: the former part $a_\omega(\tau)$ is time-frequency distribution and the later part $\sin\phi_\omega(\tau)$ is the phrase information subject to time τ and frequency ω. This is the reason that the output of "real kernel only" is filtered sub-signals, not the time-frequency distribution as processed by the "complex kernel".

After the formula derivation part, we use the output presentation to further introduce the differences between TFNs and contrast models (i.e., SincNet, WKN-Morlet, WKN-Laplace, W-CNN). For all the models, the channel numbers of the preprocessing layer are set to 64, and the experiment settings are the same as the interpretability analysis on the CWRU dataset. The illustration of the processing process of different models is shown in Fig. 16, and the simulated input signal with its time-frequency distribution (processed by STFT) is shown in the left column of the figure. The 16th kernel of different trained preprocessing layers are shown in the middle column to show the model differences in kernel shape. The preprocessing layer outputs of different trained models are shown in the right column to show the model differences in output.

As we can see from the middle column, the kernel of Rondom-CNN is randomly initialized which is the way that the traditional convolutional layer goes, and the contrast models use real value kernels and parameterize (or initialize) them to a specific frequency which are equivalent to FIR filters, while TFNs (i.e., TFN-STTF, TFN-Chirplet, and TFN-Morlet) use complex value kernels by parameterizing them, which are equivalent to the time-frequency transform (TFT).

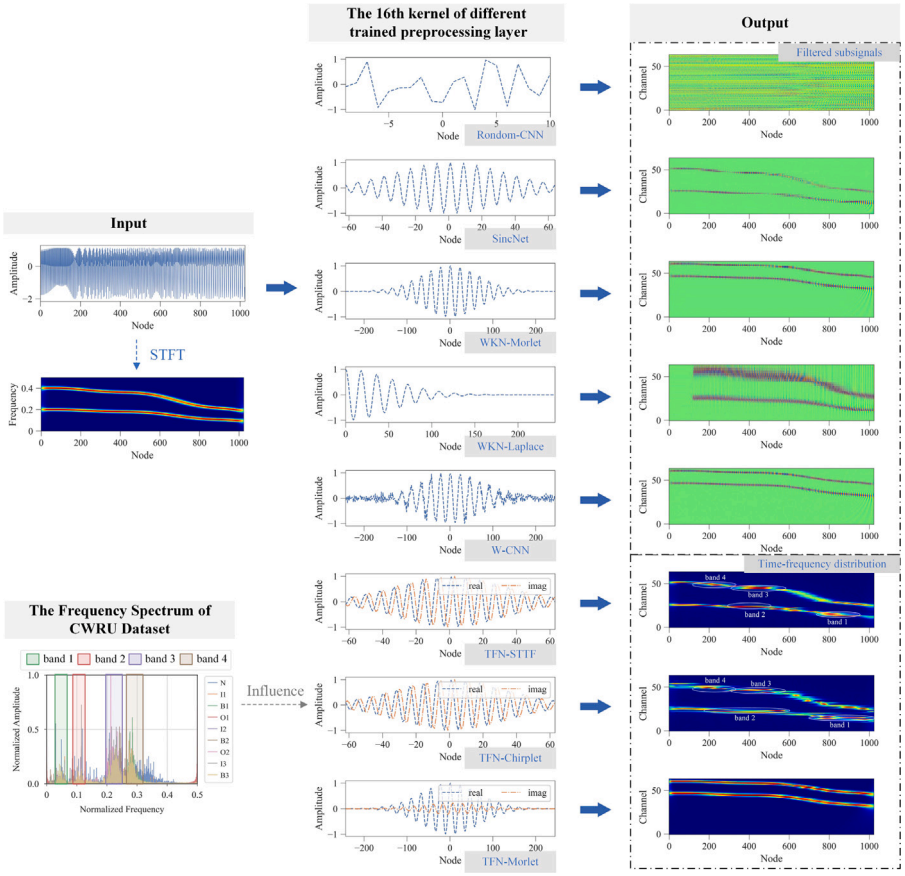


Fig. 16. The illustation of the processing process of different models.

As we can see from the right column, the output of Rondom-CNN is hard to catch any useful information. With kernel functions embedded, the outputs of contrast models have a clear correspondence to the time-frequency distribution of the input signal, meaning these models can extract time-frequency information to some extent. However, these models only consider real value kernels and their outputs contain the phase information of the input signal as explained in Eq. (16), making their ouputs blurry and still obviously different from the time-frequency distribution of the input signal processed by STFT. On the contrary, TFNs use real-imaginary mechanism to take complex value kernels into consideration, and the outputs of TFNs are exactly corresponding to the time-frequency distribution of the input signal, except for some distortions on some specific frequency bands. It is easy to conclude that these distortions are caused by the training process of the TFconv layers. As discussed in interpretability analysis (Fig. 9), TFNs will change their frequency response (FR) during the training process, and these specific frequency bands within distortions of TFN-STTF and TFN-Chirplet are exactly corresponding to the fault-related frequency bands shown in the frequency spectrum of CWRU dataset (left bottom in Fig. 16), which is also consistent with the phenomenon we discussed in the interpretability analysis on the CWRU dataset before.

In conclusion, the outputs of contrast models are a series of filtered sub-signals, that have a certain correspondence to the time-frequency distribution of the input signal but still have distinct differences becaused the phase information is contained. The outputs of TFNs are the time-frequency distribution like the result of STFT, but with more focus on the fault-related frequency bands of the training dataset, and this explains the superior diagnostic performance of TFNs becasuse fault-related features are extracted by the TFconv layer to facilitate the following fault classification.

5.2. Training process and traing time

To further analyze our TFNs models, we compare the convergence speed of different models on the CWRU dataset. Backbone models (i.e., Backbone-CNN and Rondom-CNN), contrast models (i.e., SincNet, WKN-Morlet, WKN-Laplace, and W-CNN) and TFNs (i.e., TFN-STTF, TFN-Chirplet, and TFN-Morlet) are chosen for comparison. The channel numbers of the preprocessing layer are all set to 64, the training epoch is set to 80 to get complete records, and the rest experimental settings are the same as the diagnostic experiment on the CWRU dataset.

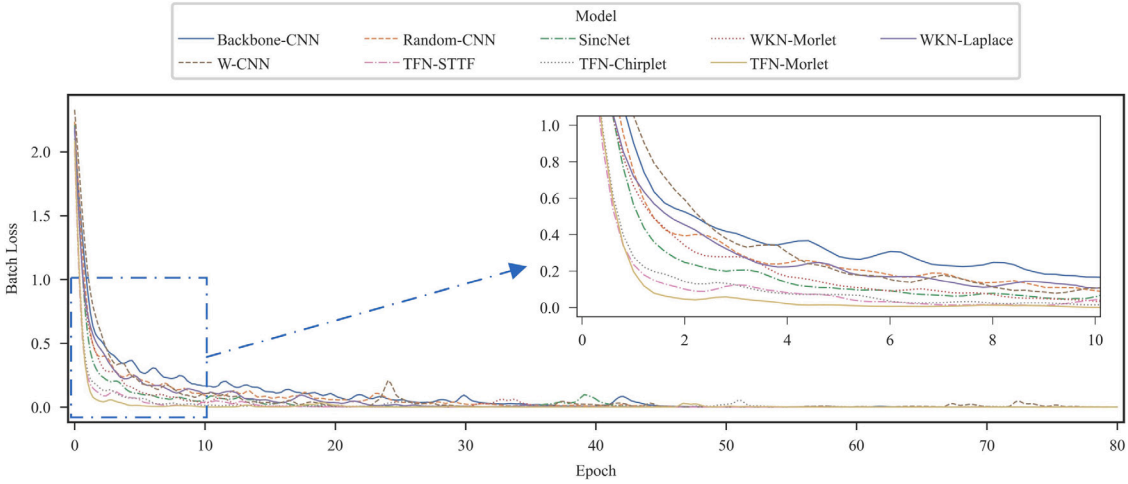


Fig. 17. The training process of different models on the CWRU dataset.

The training process of different models is shown in Fig. 17, and Backbone-CNN gets the worst performance. Random-CNN, W-CNN, and WKN-Laplace perform much better than Backbone-CNN. SincNet and WKN-Morlet belong to the second tier, and they perform slightly better than the previous three models. TFNs have a fast convergence speed than all other models, and TFN-Morlet performs best among them. The convergence speed is consistent with the diagnostic performance shown in Fig. 8, and the fast convergence speed of TFNs is attributed to their ability to extract time-frequency features.

Parameterizing convolutional kernels can significantly improve the diagnostic ability of CNN models, but the diagnostic improvement is at the cost of training time, which is barely discussed in the current literature [35–37]. To quantify the time cost of parameterizing convolutional kernel, we recorded the training time of backbone models, contrast models and TFNs with different channel numbers in the diagnostic experiment on CWRU dataset, which is shown in Fig. 18.

1. The training time of Backbone-CNN and Rondom-CNN is nearly 22 seconds per training. W-CNN just initializes the convolutional kernel, not parameterizes, so the training time of W-CNN is nearly 30 seconds per training, which is close to Backbone models.
2. SincNet, WKN-Morlet and WKN-Laplace parameterize the real value convolutional kernels, so their training times increase significantly than Backbone-CNN and that with 128 channels are close to 350 seconds, which is almost 16 times more than backbone models. Within these three models, the training times of WKN-Morlet and WKN-Laplace are more than SincNet due to their longer kernel length shown in Table 2.
3. TFN-STTF, TFN-Chirplet and TFN-Morlet parameterize the complex value convolutional kernels, and their training times with 128 channels are close to 400 seconds, which is almost 18 times more than Backbone models. Within these three models, TFN-Chirplet requires more training time than TFN-STTF because chirplet kernel function has an additional control parameter (i.e., the linear frequency modulation factor α) to be trained, and TFN-Morlet requires the most training time because morlet kernel function has the longest kernel length than other kernel functions.
4. As the number of channels increases, the training time of these models that parameterize convolutional kernel increases significantly, so the channel number of such models needs to be carefully considered to strike a balance between diagnostic accuracy and training time.

In conclusion, parameterizing convolutional kernel is extremely time-consuming, and such models require numerous training time than backbone models, while parameterizing complex value kernel (used by TFNs) is only slightly more time-consuming than parameterizing real value kernel (used by contrast models [35,36]), which makes our TFNs still competitive compared to contrast models.

5.3. Few-shot analysis

To further analyze the few-shot ability of our TFN models, we conduct a few-shot experiment on CWRU dataset. The channel numbers of the preprocessing layer are all set to 64, and the experimental settings are the same as the diagnostic performance experiment on CWRU dataset where each class has 450 samples, for a total of 4500 samples. To test the few-shot ability of TFNs, we pick a certain number (i.e., 5, 10, 20, 50, 100, 150, 200, 250, and 300) of samples in each class as training data, and the remaining samples are used for testing. Considering the difference in the number of training samples, the number of training epochs

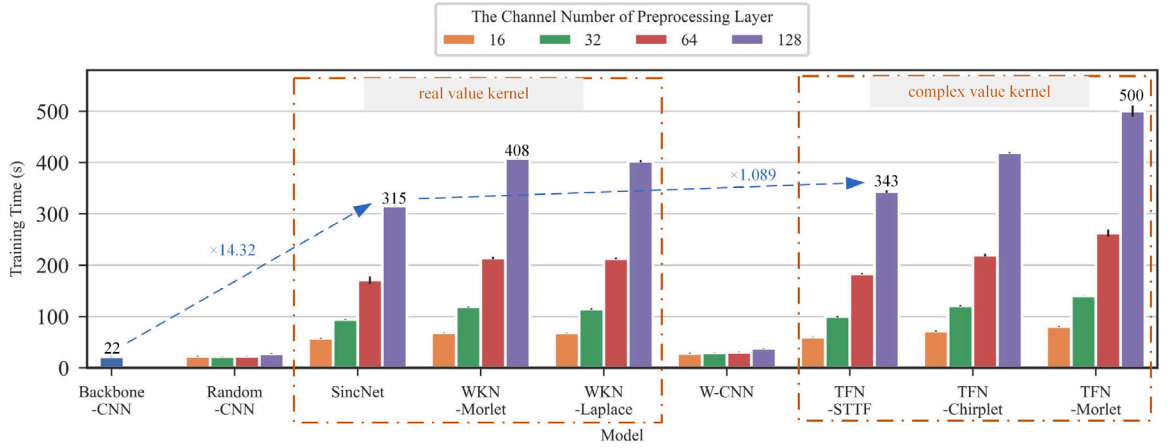


Fig. 18. The training time of different models with different preprocessing layer channels on the CWRU dataset.

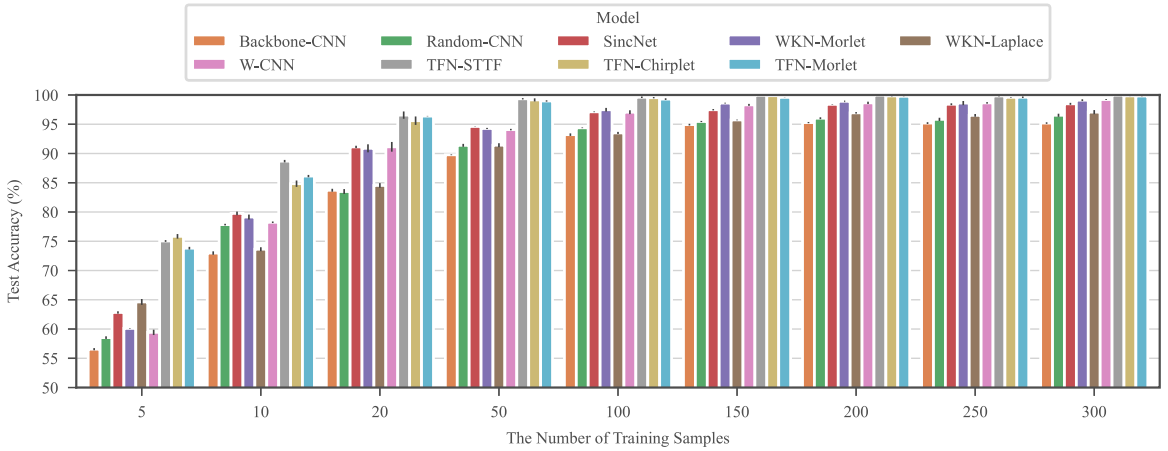


Fig. 19. The diagnostic accuracy of different models with different numbers of training samples on the CWRU dataset.

changes accordingly to ensure that the number of training batches is as equal as possible but no more than 300, which is designed as follows

$$N_{\text{epoch}} = \min\left(\frac{50 \times 300}{N_{\text{training\_sample}}}, 300\right) \tag{17}$$

where, N_{epoch} denotes the number of training epochs and $N_{\text{training\_sample}}$ denotes the number of training samples each class.

The result of the few-shot experiment is shown in Fig. 19. When the number of training samples is 5, the diagnostic accuracies of the models are all below 80%. But our TFNs (close to 75%) perform more outstandingly than other models (below 65%), and the gap in diagnostic accuracy exceeds 10%. When the number of training samples increases to 50, the diagnostic accuracy of TFNs is close to 100%, and the diagnostic accuracy gap between TFNs and other models has been narrowed to about 5%. After that, with the increase in the number of training samples, the performance of TFNs remains at about 100%, and the diagnostic accuracy of other models gradually increases, and finally, the accuracy gap is about 1~2%. In conclusion, our proposed TFNs have much better performances than other models in few-shot diagnostic tasks, and these outstanding performances are due to their ability to extract time-frequency features demonstrated in Fig. 17.

In addition, we also perform the few-shot analysis on the other two datasets. The experimental results are included in the supplementary materials, where the diagnostic accuracies of TFNs are also much higher than contrast models, effectively demonstrating the superiority of our model in few-shot scenarios.

5.4. Generalizability

In order to verify the generalizability of the TFconv layer, three typical CNNs with different depths are selected as the backbone models to obtain different TFNs. The vibration signal under 3 HP load condition of the CWRU dataset is used as the input samples,

Table 5
The diagnostic results of generalizing TFconv layers to other CNN architectures.

Backbone	Combined with	Accuracy	Variance
LeNet	None	90.02	0.177
	TFconv-STTF	95.71	0.355
	TFconv-Chirplet	94.25	0.339
	TFconv-Morlet	**98.96**	0.346
AlexNet	None	97.32	0.476
	TFconv-STTF	98.27	0.413
	TFconv-Chirplet	97.84	0.339
	TFconv-Morlet	**99.89**	0.129
ResNet	None	97.74	0.243
	TFconv-STTF	99.58	0.183
	TFconv-Chirplet	98.79	0.464
	TFconv-Morlet	**99.96**	0.058

and the channel numbers of the TFconv layer are all set to 128. Other experimental settings are consistent with those in the previous diagnostic experiments on the CWRU dataset, and the diagnosis results are shown in Table 5.

It can be seen from the results that, the TFconv layers can significantly improve the diagnostic accuracy on the basis of different backbone CNNs. The TFconv layer with the morlet kernel always achieves the best diagnostic performance, followed by the STTF kernel. The chirplet kernel is slightly worse than the Morlet kernel and STTF kernel, but still better than the backbone model. This experiment shows that the proposed TFconv layer is a general method and can be applied to other CNNs with different depths to improve their diagnostic performance. Besides, this experiment also illustrates the importance of the backbone model and we recommend researchers adopt a backbone CNN with enough depth to formulate the TFN.

6. Conclusion

In this article, an interpretable time-frequency convolutional (TFconv) layer is proposed to extract fault-related time-frequency information. Taking the TFconv layer as a preprocessing layer, we formulated the time-frequency network (TFN) to achieve higher diagnostic accuracy and explain the focusing frequency area in the prediction-making of CNN models. The diagnostic effectiveness and interpretability of TFN have been verified by three datasets of mechanical fault diagnosis experiments. The conclusions of this article could be summarized as follows: 1) The participation of the TFconv layer can greatly improve the diagnostic performance of the CNN in mechanical fault diagnosis tasks. 2) The TFconv layer could explain the focusing frequency area of TFN to extract features and make predictions. 3) The kernel function and channel number of the TFconv layer have a great influence on the diagnostic performance of TFN, and the TFconv layer with 64-channel STTF kernel can achieve the overall optimum in accuracy and efficiency. 4) The TFconv layer has outstanding performance on convergence speed and few-shot scenarios, and can be generalized to other CNN models with different depths. In future research, we will explore other kernel functions with adaptive frequency bandwidth to interpret focusing frequency more explicitly, and investigate the effectiveness of the TFconv layer for other neural networks other than CNNs (e.g. autoencoder).

Declaration of competing interest

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

Data availability

The authors do not have permission to share data.

Acknowledgments

This project was supported by the National Natural Science Foundation of China under Grant No. 12272219, and No. 12121002.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ymssp.2023.110952.

References

[1] J. Li, X. Wang, H. Wu, Rolling bearing fault detection based on improved piecewise unsaturated bistable stochastic resonance method, IEEE Trans. Instrum. Meas. 70 (2021) 1–9, http://dx.doi.org/10.1109/TIM.2020.3024038.

[2] Y. Li, M. Xu, Y. Wei, W. Huang, An improvement EMD method based on the optimized rational Hermite interpolation approach and its application to gear fault diagnosis, Measurement 63 (2015) 330–345, http://dx.doi.org/10.1016/j.measurement.2014.12.021.

[3] F. Tao, Q. Qi, A. Liu, A. Kusiak, Data-driven smart manufacturing, J. Manuf. Syst. 48 (2018) 157–169, http://dx.doi.org/10.1016/j.jmsy.2018.01.006.

[4] H. Chen, B. Jiang, S.X. Ding, B. Huang, Data-driven fault diagnosis for traction systems in high-speed trains : A survey, challenges, and perspectives, IEEE Trans. Intell. Transp. Syst. 23 (3) (2022) 1700–1716, http://dx.doi.org/10.1109/TITS.2020.3029946.

[5] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A.K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mech. Syst. Signal Process. 138 (2020) http://dx.doi.org/10.1016/j.ymssp.2019.106587.

[6] R. Li, D. He, Rotational machine health monitoring and fault detection using EMD-based acoustic emission feature quantification, IEEE Trans. Instrum. Meas. 61 (4) (2012) 990–1001, http://dx.doi.org/10.1109/TIM.2011.2179819.

[7] Y. Si, Y. Wang, D. Zhou, Key-performance-indicator-related process monitoring based on improved kernel partial least squares, IEEE Trans. Ind. Electron. 68 (3) (2021) 2626–2636, http://dx.doi.org/10.1109/TIE.2020.2972472.

[8] Z. Peng, F. Chu, Application of the wavelet transform in machine condition monitoring and fault diagnostics: A review with bibliography, Mech. Syst. Signal Process. 18 (2) (2004) 199–221, http://dx.doi.org/10.1016/S0888-3270(03)00075-X.

[9] Y. Li, M. Xu, X. Liang, W. Huang, Application of bandwidth EMD and adaptive multiscale morphology analysis for incipient fault diagnosis of rolling bearings, IEEE Trans. Ind. Electron. 64 (8) (2017) 6506–6517, http://dx.doi.org/10.1109/TIE.2017.2650873.

[10] A. Brunetti, D. Buongiorno, G.F. Trotta, V. Bevilacqua, Computer vision and deep learning techniques for pedestrian detection and tracking: A survey, Neurocomputing 300 (2018) 17–33, http://dx.doi.org/10.1016/j.neucom.2018.01.092.

[11] H.M. Fayek, M. Lech, L. Cavedon, Evaluating deep learning architectures for speech emotion recognition, Neural Netw. 92 (2017) 60–68, http://dx.doi.org/10.1016/j.neunet.2017.02.013.

[12] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, D. Hassabis, Mastering the game of go without human knowledge, Nature 550 (7676) (2017) 354–359, http://dx.doi.org/10.1038/nature24270.

[13] H. Shao, H. Jiang, H. Zhang, T. Liang, Electric locomotive bearing fault diagnosis using a novel convolutional deep belief network, IEEE Trans. Ind. Electron. 65 (3) (2018) 2727–2736, http://dx.doi.org/10.1109/TIE.2017.2745473.

[14] Y. Wang, R. Liu, D. Lin, D. Chen, P. Li, Q. Hu, C.L.P. Chen, Coarse-to-fine : Progressive knowledge transfer-based multitask convolutional neural network for intelligent large-scale fault diagnosis, IEEE Trans. Neural Netw. Learn. Syst. (2021) http://dx.doi.org/10.1109/TNNLS.2021.3100928.

[15] X. Zhao, J. Yao, W. Deng, P. Ding, Y. Ding, M. Jia, Z. Liu, Intelligent fault diagnosis of gearbox under variable working conditions with adaptive intraclass and interclass convolutional neural network, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–15, http://dx.doi.org/10.1109/TNNLS.2021.3135877.

[16] D. Peng, H. Wang, Z. Liu, W. Zhang, M.J. Zuo, J. Chen, Multibranch and multiscale CNN for fault diagnosis of wheelset bearings under strong noise and variable load condition, IEEE Trans. Ind. Inform. 16 (7) (2020) 4949–4960, http://dx.doi.org/10.1109/TII.2020.2967557.

[17] X. Nie, G. Xie, A novel normalized recurrent neural network for fault diagnosis with noisy labels, J. Intell. Manuf. 32 (5) (2021) 1271–1288, http://dx.doi.org/10.1007/s10845-020-01608-8.

[18] Z. Zhao, T. Li, J. Wu, C. Sun, S. Wang, R. Yan, X. Chen, Deep learning algorithms for rotating machinery intelligent diagnosis: an open source benchmark study, ISA Trans. 107 (2020) 224–255, http://dx.doi.org/10.1016/j.isatra.2020.08.010.

[19] Q.-s. Zhang, S.-c. Zhu, Visual interpretability for deep learning: A survey, Front. Inf. Technol. Electron. Eng. 19 (1) (2018) 27–39, http://dx.doi.org/10.1631/FITEE.1700808.

[20] P.-P. Xi, Y.-P. Zhao, P.-X. Wang, Z.-Q. Li, Y.-T. Pan, F.-Q. Song, Least squares support vector machine for class imbalance learning and their applications to fault detection of aircraft engine, Aerosp. Sci. Technol. 84 (2019) 56–74, http://dx.doi.org/10.1016/j.ast.2018.08.042.

[21] M. Ivanovs, R. Kadikis, K. Ozols, Perturbation-based methods for explaining deep neural networks: A survey, Pattern Recognit. Lett. 150 (2021) 228–234, http://dx.doi.org/10.1016/j.patrec.2021.06.030.

[22] F.-L. Fan, J. Xiong, M. Li, G. Wang, On interpretability of artificial neural networks : A Survey, IEEE Trans. Radiat. Plasma Med. Sci. 5 (6) (2021) 741–760, http://dx.doi.org/10.1109/TRPMS.2021.3066428.

[23] Y. Zhang, P. Tino, A. Leonardis, K. Tang, A survey on neural network interpretability, IEEE Trans. Emerg. Top. Comput. Intell. 5 (5) (2021) 726–742, http://dx.doi.org/10.1109/TETCI.2021.3100641.

[24] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, P. Das, Explanations based on the missing : Towards contrastive explanations with pertinent negatives, in: Proc. Adv. Neural Inf. Process. Syst. Vol. 31, NeurIPS, Curran Associates, Inc., 2018.

[25] Y. Wang, H. Su, B. Zhang, X. Hu, Interpret neural networks by Identifying Critical Data Routing Paths, in: Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, IEEE, Salt Lake City, UT, 2018, pp. 8906–8914, http://dx.doi.org/10.1109/CVPR.2018.00928.

[26] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection : quantifying interpretability of deep visual representations, in: Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, IEEE, Honolulu, HI, 2017, pp. 3319–3327, http://dx.doi.org/10.1109/CVPR.2017.354.

[27] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks : Visualising image classification models and saliency maps, 2014, arXiv:1312.6034.

[28] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM : Visual explanations from deep networks via gradient-based localization, in: Proc. IEEE Int. Conf. Comput. Vis., ICCV, IEEE, Venice, 2017, pp. 618–626, http://dx.doi.org/10.1109/ICCV.2017.74.

[29] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 3319–3328.

[30] O. Li, H. Liu, C. Chen, C. Rudin, Deep learning for case-based reasoning through prototypes : A neural network that explains its predictions, Proc. AAAI Conf. Artif. Intell. 32 (1) (2018) http://dx.doi.org/10.1609/aaai.v32i1.11771.

[31] J. Li, Y. Wang, Y. Zi, Z. Zhang, Whitening-Net : A generalized network to diagnose the faults among different machines and conditions, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–14, http://dx.doi.org/10.1109/TNNLS.2021.3071564.

[32] X. Wu, Y. Zhang, C. Cheng, Z. Peng, A hybrid classification autoencoder for semi-supervised fault diagnosis in rotating machinery, Mech. Syst. Signal Process. 149 (2021) 107327, http://dx.doi.org/10.1016/j.ymssp.2020.107327.

[33] D. Wang, Y. Chen, C. Shen, J. Zhong, Z. Peng, C. Li, Fully interpretable neural network for locating resonance frequency bands for machine condition monitoring, Mech. Syst. Signal Process. 168 (2022) 108673, http://dx.doi.org/10.1016/j.ymssp.2021.108673.

[34] B. Zhao, C. Cheng, G. Tu, Z. Peng, Q. He, G. Meng, An interpretable denoising layer for neural networks based on reproducing kernel Hilbert space and its application in machine fault diagnosis, Chin. J. Mech. Eng. 34 (1) (2021) 44, http://dx.doi.org/10.1186/s10033-021-00564-5.

[35] T. Li, Z. Zhao, C. Sun, L. Cheng, X. Chen, R. Yan, R.X. Gao, WaveletKernelNet : An interpretable deep neural network for industrial intelligent diagnosis, IEEE Trans. Syst. Man Cybern. Syst. 52 (4) (2022) 2302–2312, http://dx.doi.org/10.1109/TSMC.2020.3048950.

[36] M. Ravanelli, Y. Bengio, Interpretable convolutional filters with SincNet , 2019, arXiv:1811.09725.

[37] B. Ganguly, S. Chaudhury, S. Biswas, D. Dey, S. Munshi, B. Chatterjee, S. Dalai, S. Chakravorti, Wavelet kernel based convolutional neural network for localization of partial discharge sources within a power apparatus, IEEE Trans. Ind. Inform. (2020) 1, http://dx.doi.org/10.1109/TII.2020.2991686.
[38] G. Michau, G. Frusque, O. Fink, Fully learnable deep wavelet transform for unsupervised monitoring of high-frequency time series, Proc. Natl. Acad. Sci. 119 (8) (2022) e2106598119, http://dx.doi.org/10.1073/pnas.2106598119.
[39] Y. Yang, W. Zhang, Z. Peng, G. Meng, Multicomponent signal analysis based on polynomial chirplet transform, IEEE Trans. Ind. Electron. 60 (9) (2013) 3948–3956, http://dx.doi.org/10.1109/TIE.2012.2206331.
[40] G. Tu, X. Dong, S. Chen, B. Zhao, L. Hu, Z. Peng, Iterative nonlinear chirp mode decomposition: A Hilbert-Huang transform-like method in capturing intra-wave modulations of nonlinear responses, J. Sound Vib. 485 (2020) 115571, http://dx.doi.org/10.1016/j.jsv.2020.115571.
[41] M.X. Cohen, A better way to define and describe morlet wavelets for time-frequency analysis, NeuroImage 199 (2019) 81–86, http://dx.doi.org/10.1016/j.neuroimage.2019.05.048.
[42] V. Andrearczyk, P.F. Whelan, Using filter banks in convolutional neural networks for texture classification, Pattern Recognit. Lett. 84 (2016) 63–69.
[43] A.V. Oppenheim, A.S. Willsky, S.H. Nawab, G.M. Hernández, et al., Signals & Systems, Pearson Educación, 1997.
[44] W.A. Smith, R.B. Randall, Rolling element bearing diagnostics using the case western reserve university data: A benchmark study, Mech. Syst. Signal Process. 64–65 (2015) 100–131, http://dx.doi.org/10.1016/j.ymssp.2015.04.021.

Generalization Characteristics of Complex-Valued Feedforward Neural Networks in Relation to Signal Coherence

Akira Hirose, *Senior Member, IEEE*, and Shotaro Yoshida

***Abstract*— Applications of complex-valued neural networks (CVNNs) have expanded widely in recent years—in particular in radar and coherent imaging systems. In general, the most important merit of neural networks lies in their generalization ability. This paper compares the generalization characteristics of complex-valued and real-valued feedforward neural networks in terms of the coherence of the signals to be dealt with. We assume a task of function approximation such as interpolation of temporal signals. Simulation and real-world experiments demonstrate that CVNNs with amplitude-phase-type activation function show smaller generalization error than real-valued networks, such as bivariate and dual-univariate real-valued neural networks. Based on the results, we discuss how the generalization characteristics are influenced by the coherence of the signals depending on the degree of freedom in the learning and on the circularity in neural dynamics.**

***Index Terms*— Complex-valued neural network, function approximation, generalization, supervised learning.**

I. Introduction

COMPLEX-VALUED neural networks (CVNNs) have extended their applications into various fields, such as metal-defect imagers [1], radars (including ground penetrating radars to visualize plastic landmines) [2], [3], blur-compensation image processing [4], blind separation based on principal component analysis in sonar [6] and voice processing [7], filtering and other time-sequential signal processing [8], frequency-domain multiplexed microwave signal processing [9] and pulse beamforming in ultra-wideband communications [10], frequency-domain multiplexed neural networks and learning logic circuits using lightwave [11]–[13], theory and design of quantum computation based on superconductive devices [14], and developmental motion learning [15]. Many of these applications deal with wave-related signals, or coherent signals, in the time/space domain or Fourier domain.

Research on general CVNNs has revealed various aspects of their dynamics. It is true that a complex number is represented by a pair of real numbers, namely, real and imaginary parts, or amplitude and phase. A variety of useful neural dynamics are actually obtained by paying attention to the real and imaginary parts [16]–[18] or amplitude and phase [19], [20]. This fact sometimes leads to an assumption that a CVNN is almost equivalent to a double-dimensional real-valued neural network (RVNN). However, it has been discussed intuitively that CVNN has only a smaller degree of freedom at the synaptic weighting, because of the specific complex multiplication in the four arithmetic operations, and that this fact suggests the realization of learning with the presumption to obey phase-rotational dynamics [21].

In this paper, we compare complex- and real-valued neural networks by focusing on their generalization characteristics. Generalization is one of the features most useful and specific to neural networks. We investigate the generalization ability of feedforward CVNNs and RVNNs for function approximation, such as filtering and interpolation. We observe the characteristics by feeding signals that have various degrees of coherence.

We examine four types of neural networks. As a basic complex-valued network, we adopt a CVNN having an amplitude-phase-type activation function because of its suitability to circular (isotropic in the complex plane) signals, which are often observed in coherent imaging and other electronics. We also examine a CVNN having a real-imaginary separate-type activation function (RI-CVNN), which is another widely used CVNN. Regarding real-valued networks, basically we construct a double-input-terminal double-output-neuron RVNN with full connections between layers, which processes real and imaginary signals in a bivariate manner. We also prepare a dual-univariate RVNN (dual-RVNN) with the same numbers of input terminals and output neurons but processing the real part of the signals separately from the imaginary part.

Computer simulations as well as experiments for real-world signal data demonstrate that the generalization characteristics of the four types of the neural networks are different depending on the signal coherence. We discuss the origin of the difference in the generalization abilities based on simulation and experiment results.

This paper is organized as follows. Section II presents the construction of experiments, network structures, and learning dynamics. In Section III, we conduct computer simulations as well as electronics experiments. We find that the generalization characteristics are different among the four networks. In Section IV, we discuss how the signal properties influence the generalization characteristics. Section V concludes this paper.

Manuscript received May 8, 2011; revised January 5, 2011; accepted January 5, 2011. Date of publication January 23, 2012; date of current version March 6, 2012.

The authors are with the Department of Electrical Engineering and Information Systems, University of Tokyo, Tokyo 113-8656, Japan (e-mail: ahirose@ee.t.u-tokyo.ac.jp; yoshida@eis.t.u-tokyo.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2183613

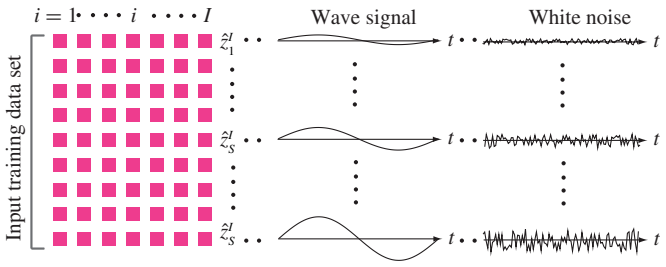


Fig. 1. Set of teacher signals.

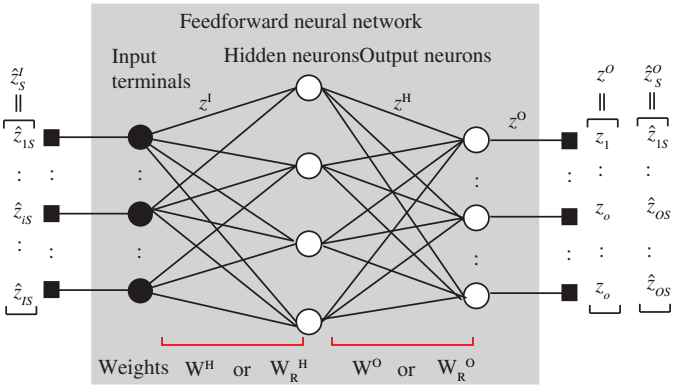


Fig. 2. Basic construction of the complex- and real-valued feedforward neural networks.

II. Construction of Experiments and Learning Dynamics

A. Coherence

We investigate numerically and experimentally the difference in the generalization characteristics of CVNNs and RVNNs. We also focus on the dependence of the generalization on coherence (degree of wave nature) of the signals. In the following computer simulation and experiments, we observe the generalization characteristics by changing the coherence of the signals with the following definition.

Fig. 1 illustrates the generation of signals having various coherence degrees. We take a sinusoidal time-sequential signal $v(t) = a\exp(\jmath\omega t)$ of time t as a completely coherent signal where a and ω are the amplitude and angular frequency, respectively. We memorize the serial data discretely, and feed it to the neural input terminals in parallel, just like in the case of a finite impulse response filter. We may add white Gaussian noise (WGN) to the sinusoid to reduce the coherence. WGN is completely incoherent. By changing the ratio of the sinusoid to the white noise, we prepare signals having various coherence degrees. Then the signal-to-noise ratio $SNR \equiv P_s/P_n$, the ratio of the sinusoidal wave power P_s to the WGN power P_n, shows the coherence degree quantitatively. When $SNR = \infty$, the signal is completely coherent, while when $SNR = 0$ (or $-\infty$dB), it is completely incoherent.

Section II-B describes the construction of the computer simulations and experiments shown in this paper. We choose a typical situation in which we can observe the dependence of generalization on the coherence. We implicitly suppose a neural network that processes signals related to lightwave, electromagnetic wave, or ultrasonic wave in, for example, a sonar imaging system.

Regarding the word "coherence," we have several closely related concepts. One of them is bandwidth, which expresses signal characteristics in the spectral domain. A high coherence signal is a narrowband signal. In the spectrum, the amount of disturbance that reduces the coherence is determined by the product of two factors, namely, bandwidth and spectral power density. Here we adopt a widely applicable WGN as the disturbance, which has infinitely wide bandwidth. In the experiment, however, we deal with discrete-time signals and, therefore, we have a limited bandwidth determined automatically by the sampling process. Then, the above definition of coherence by the signal to noise ratio (SNR) corresponds to the level of the flat WGN power density under a constant bandwidth condition. In this sense, the SNR is directly related to the coherence under the constant bandwidth condition. Another concept is circularity, which mostly means probabilistic isotropy of the signal in the complex plane. This is an important concept in, for example, prediction of wind force and direction for wind power generation. In a sinusoidal wave, circularity is related to distortion. However, the circularity does not include temporal continuity characteristics explicitly or sampling interval influence for discretization of continuous signals. Circularity and the widely linear processing will be discussed in Section IV.

B. Organization of the Experiment

The outline of our experiments is as follows.

1) *Input signals*: weighted summation of the following.
   a) Sinusoid: completely coherent signal.
   b) WGN: completely incoherent data having random amplitude and phase (or real and imaginary parts).
2) *Task to learn*: identity mapping, which is expected to show the learning characteristics most clearly for the above signals with various degrees of coherence.
3) *Evaluation of generalization*: observation of the generalization error when the input signals shift in time, or when the amplitude is changed.

C. Forward Processing and Learning Dynamics

1) CVNN: Fig. 2 shows the general construction of the neural network to be considered here. It is a layered feedforward network having input terminals, hidden neurons, and output neurons. In a CVNN, we first employ a phase–amplitude-type sigmoid activation function and the teacher–signal–backpropagation learning process, [19], [22] with the following notations:

$$z^{\mathrm{I}} = [z_1, \ldots, z_i, \ldots, z_I, z_{I+1}]^T \quad \text{(Input signal vector)} \tag{1}$$

$$z^{\mathrm{H}} = [z_1, \ldots, z_h, \ldots, z_H, z_{H+1}]^T \quad \text{(Hidden} - \text{layer output signal vector)} \tag{2}$$

$$z^{\mathrm{O}} = [z_1, \ldots, z_o, \ldots, z_O]^T \quad \text{(Output} - \text{layer signal vector)} \tag{3}$$

$$\mathbf{W}^{\mathrm{H}} = [w_{hi}] \quad \text{(Hidden neuron weight matrix)} \tag{4}$$

$$\mathbf{W}^{\mathrm{O}} = [w_{oh}] \quad \text{(Output neuron weight matrix)} \tag{5}$$

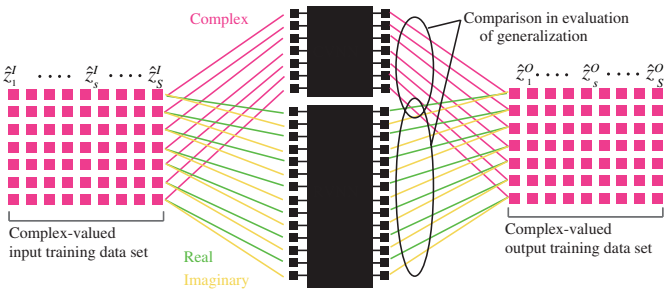


Fig. 3. Schematic diagram of the learning process for pairs of input–output teachers.

where $[\cdot]^T$ means transpose. In (4) and (5), the weight matrices include additional weights $w_{h\ I+1}$ and $w_{o\ H+1}$, equivalent to neural thresholds, where we add formal constant inputs $z_{I+1} = 1$ and $z_{H+1} = 1$ in (1) and (2), respectively. The respective signal vectors and synaptic weights are connected with one another through an activation function $f(z)$ as

$$z^{\mathrm{H}} = f\left(\mathbf{W}^{\mathrm{H}} z^{\mathrm{I}}\right) \tag{6}$$

$$z^{\mathrm{O}} = f\left(\mathbf{W}^{\mathrm{O}} z^{\mathrm{H}}\right) \tag{7}$$

where $f(z)$ is a function of each vector element z ($\in \boldsymbol{C}$) defined as

$$f(z) = \tanh(|z|)\exp(\jmath \arg z). \tag{8}$$

Fig. 3 is a diagram to explain the supervised learning process. We prepare a set of teacher signals at the input $\hat{z}_s^{\mathrm{I}} = [\hat{z}_{1s}, \ldots, \hat{z}_{is}, \ldots, \hat{z}_{Is}, \hat{z}_{I+1\ s}]^T$ and the output $\hat{z}_s^{\mathrm{O}} = [\hat{z}_{1s}, \ldots, \hat{z}_{os}, \ldots, \hat{z}_{Os}]^T$ $(s = 1, \ldots, s, \ldots S)$ for which we employ the teacher–signal backpropagation learning. We define an error function E to obtain the dynamics by referring to [9], [19], and [22] as

$$E \equiv \frac{1}{2}\sum_{s=1}^{S}\sum_{o=1}^{O}\left|z_o(\hat{z}_s^{\mathrm{I}}) - \hat{z}_{os}\right|^2 \tag{9}$$

$$\left|w_{oh}^{\mathrm{new}}\right| = \left|w_{oh}^{\mathrm{old}}\right| - K\frac{\partial E}{\partial |w_{oh}|} \tag{10}$$

$$\arg w_{oh}^{\mathrm{new}} = \arg w_{oh}^{\mathrm{old}} - K\frac{1}{|w_{oh}|}\frac{\partial E}{\partial(\arg w_{oh})} \tag{11}$$

$$\begin{aligned}\frac{\partial E}{\partial |w_{oh}|} = {} & \left(1 - |z_o|^2\right)\left(|z_o| - \left|\hat{z}_o\right|\cos\left(\arg z_o - \arg \hat{z}_o\right)\right)|z_h| \\ & \cdot \cos\left(\arg z_o - \arg \hat{z}_o - \arg w_{oh}\right) \\ & - |z_o|\left|\hat{z}_o\right|\sin\left(\arg z_o - \arg \hat{z}_o\right)\frac{|z_h|}{\tanh^{-1}|z_o|} \\ & \cdot \sin\left(\arg z_o - \arg \hat{z}_o - \arg w_{oh}\right)\end{aligned} \tag{12}$$

$$\begin{aligned}\frac{1}{|w_{oh}|}\frac{\partial E}{\partial(\arg w_{oh})} = {} & \left(1 - |z_o|^2\right)\left(|z_o| - \left|\hat{z}_o\right|\cos\left(\arg z_o - \arg \hat{z}_o\right)\right)|z_h| \\ & \cdot \sin\left(\arg z_o - \arg \hat{z}_o - \arg w_{oh}\right) \\ & + |z_o|\left|\hat{z}_o\right|\sin\left(\arg z_o - \arg \hat{z}_o\right)\frac{|z_h|}{\tanh^{-1}|z_o|} \\ & \cdot \cos\left(\arg z_o - \arg \hat{z}_o - \arg w_{oh}\right)\end{aligned} \tag{13}$$

where $(\cdot)^{\mathrm{new}}$ and $(\cdot)^{\mathrm{old}}$ indicates the update of the weights from $(\cdot)^{\mathrm{old}}$ to $(\cdot)^{\mathrm{new}}$, and K is a learning constant. The teacher signals at the hidden layer $\hat{z}^{\mathrm{H}} = [\hat{z}_1, \ldots, \hat{z}_h, \ldots, \hat{z}_H]^T$ is obtained by making the output teacher vector itself $\hat{z}^{\mathrm{O}}$ propagate backward as

$$\hat{z}^{\mathrm{H}} = \left(f\left(\left(\hat{z}^{\mathrm{O}}\right)^* \hat{\mathbf{W}}^{\mathrm{O}}\right)\right)^* \tag{14}$$

where $(\cdot)^*$ denotes the Hermite conjugate. Using $\hat{z}^{\mathrm{H}}$, the hidden layer neurons change their weights by following (10)–(13) with replacement of the suffixes o,h with h, i [11], [22].

2) CVNN Having Real–Imaginary Separate-Type Activation Function (RI-CVNN): We also investigate the characteristics of CVNNs having real–imaginary separate-type activation function. Instead of (8), a neuron has an activation function expressed as

$$f(z) = \tanh(\mathbf{Re}[z]) + \jmath \tanh(\mathbf{Im}[z]). \tag{15}$$

The structure and the dynamics of feedforward processing and backpropagation learning are those described in, for example, [18].

3) RVNN Having Double Input Terminals and Output Neurons for Bivariate Procesing (RVNN): Similarly, the forward processing and learning of a RVNN having double input terminals and output neurons are explained as follows. Fig. 3 includes also this case. We represent a complex number as a pair of real numbers as $z_i = x_{2i-1} + \jmath x_{2i}$. Then we have a double number of terminals for real and imaginary parts of input signals $z_{\mathrm{R}}^{\mathrm{I}}$ and a double number of output neurons to generate real and imaginary parts of output signals $z_{\mathrm{R}}^{\mathrm{O}}$. We also prepare a double number of hidden neurons for hidden layer signals $z_{\mathrm{R}}^{\mathrm{H}}$ so that the equivalent number of neurons is the same as that of the above CVNN.

Forward signal processing connects the signal vectors as well as hidden neuron weights $\mathbf{W}_{\mathrm{R}}^{\mathrm{H}}$ and output neuron weights $\mathbf{W}_{\mathrm{R}}^{\mathrm{O}}$ through a real-valued activation function f_{R} as

$$\begin{aligned} z_{\mathrm{R}}^{\mathrm{I}} = [& \overbrace{x_1, \quad x_2}^{\text{real\&imaginary}}, \ldots, x_{2i-1}, x_{2i}, \ldots, \\ & x_{2I-1}, x_{2I}, x_{2I+1}, x_{2I+2}]^T \\ & \left(= z^{\mathrm{I}}\right) \quad \text{(Input signal vector)} \end{aligned} \tag{16}$$

$$\begin{aligned} z_{\mathrm{R}}^{\mathrm{H}} = {} & [x_1, x_2, \ldots, x_{2h-1}, x_{2h}, \ldots, \\ & x_{2H-1}, x_{2H}, x_{2H+1}, x_{2H+2}]^T \\ & \text{(Hidden} - \text{layer output signal vector)} \end{aligned} \tag{17}$$

$$\begin{aligned} z_{\mathrm{R}}^{\mathrm{O}} = {} & [x_1, x_2, \ldots, x_{2o-1}, x_{2o}, \ldots, x_{2O-1}, x_{2O}]^T \\ & \text{(Output} - \text{layer signal vector)} \end{aligned} \tag{18}$$

$$\mathbf{W}_{\mathrm{R}}^{\mathrm{H}} = [w_{\mathrm{R}hi}] \quad \text{(Hidden neuron weight matrix)} \tag{19}$$

$$\mathbf{W}_{\mathrm{R}}^{\mathrm{O}} = [w_{\mathrm{R}oh}] \quad \text{(Output neuron weight mateix)} \tag{20}$$

$$z_{\mathrm{R}}^{\mathrm{H}} = f_{\mathrm{R}}\left(\mathbf{W}_{\mathrm{R}}^{\mathrm{H}} z_{\mathrm{R}}^{\mathrm{I}}\right) \tag{21}$$

$$z_{\mathrm{R}}^{\mathrm{O}} = f_{\mathrm{R}}\left(\mathbf{W}_{\mathrm{R}}^{\mathrm{O}} z_{\mathrm{R}}^{\mathrm{H}}\right) \tag{22}$$

$$f_{\mathrm{R}}(x) = \tanh(x) \tag{23}$$

where the thresholds are $w_{\mathrm{R}\ h\ 2I+1}$, $w_{\mathrm{R}\ h\ 2I+2}$, $w_{\mathrm{R}\ h\ 2H+1}$, and $w_{\mathrm{R}\ h\ 2H+2}$ with formal additional inputs $x_{2H+1} = 1$,

TABLE I
PARAMETERS IN THE NEURAL NETWORKS

	CVNN or RI-CVNN	RVNN or dual-RVNN
Number of input neurons	$I = 16$	$2I = 32$
Number of hidden neurons	$H = 25$	$2H = 50$
Number of output neurons	$O = 16$	$2O = 32$
Learning constant	$K = 0.01$	$K = 0.01$

$x_{2H+2} = 1$, $x_{2H+1} = 1$, and $x_{2H+2} = 1$. We employ the conventional error backpropagation learning. That is, we define an error function E_{R} for a set of input and output teacher signals $(\hat{z}_s^{\mathrm{I}}, \hat{z}_s^{\mathrm{O}})$ to obtain the learning dynamics as

$$E_{\mathrm{R}} \equiv \frac{1}{2}\sum_{s=1}^{S}\sum_{o=1}^{2O}\left|x_o(\hat{z}_{\mathrm{R}s}^{\mathrm{I}}) - \hat{x}_{os}\right|^2 \quad (= E) \tag{24}$$

$$w_{\mathrm{R}oh}^{\mathrm{new}} = w_{\mathrm{R}oh}^{\mathrm{old}} - K\frac{\partial E_{\mathrm{R}}}{\partial w_{\mathrm{R}oh}} \tag{25}$$

$$w_{\mathrm{R}ji}^{\mathrm{new}} = w_{\mathrm{R}hi}^{\mathrm{old}} - K\frac{\partial E_{\mathrm{R}}}{\partial w_{\mathrm{R}hi}} \tag{26}$$

$$\frac{\partial E_{\mathrm{R}}}{\partial w_{\mathrm{R}oh}} = \left(x_o - \hat{x}_o\right)\left(1 - x_o^2\right)x_h \tag{27}$$

$$\frac{\partial E_{\mathrm{R}}}{\partial w_{\mathrm{R}hi}} = \left(\sum_{o}^{2O}\left(x_o - \hat{x}_o\right)\left(1 - x_o^2\right)w_{oh}\right)\left(1 - x_h^2\right)x_i. \tag{28}$$

4) Dual Real-Valued Neural Networks for Real-Imaginary Separate Processing (Dual-RVNN): We consider another type of RVNN in which the real and imaginary parts of input signals are processed separately. It is an extension of dual univariate RVNN having single-layer structure. We may have a variety of ways of mixing and separating the real and imaginary variables in multiple-layer networks. With this network, we examine a completely separate case where the neurons in the real-part network have no connections to those in the imaginary-part network. The learning and processing dynamics are identical to those of the above RVNN except that the numbers of input terminals and output neurons are the same as the CVNNs for the respective real and imaginary networks.

III. EXPERIMENTS

A. Experimental Setup

1) Simulation Setup: Fig. 4 shows schematically how to observe the generalization characteristics of the networks. We conducted the learning process as follows. We chose the identity mapping as the task to be learned to show the network characteristics most clearly. That is, we take a set of input and output teacher signals as $\hat{z}_s^{\mathrm{I}} = \hat{z}_s^{\mathrm{O}}$ $(s = 1, 2, \ldots, S)$ with the following conditions. For a signal set showing high coherence, we choose its wavelength in such a manner that a unit wave spans just over the neural input terminals $i = 1, \ldots, I$, and discrete I points are fed to the network evenly with a constant interval in the unit wave. In more detail, we choose multiple amplitude values between 0 and 1 evenly for

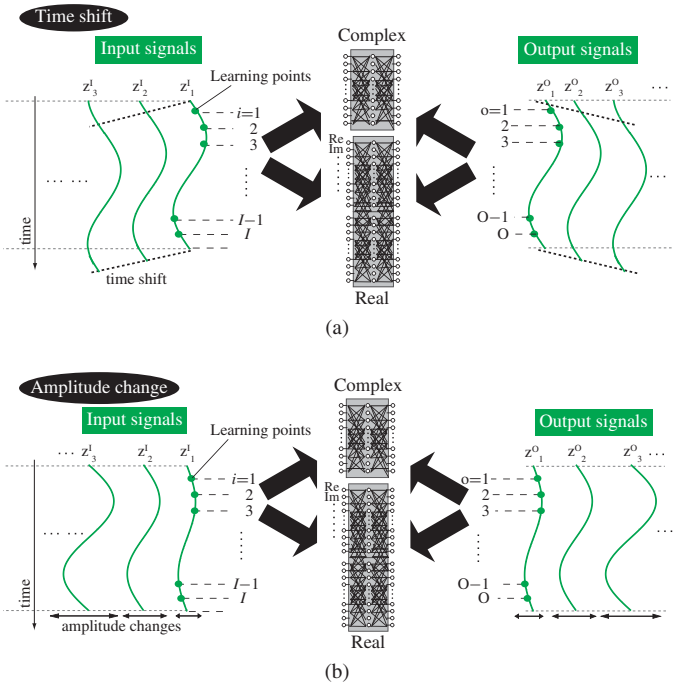


Fig. 4. Schematic diagrams showing how to feed signals to observe (a) time shift and (b) amplitude change generalizations.

$s_{\mathrm{A}} = 1, \ldots, S_{\mathrm{A}}$ teacher signals as well as multiple time shift amounts between 0 and half-wave duration (phase shift in a sinusoidal case between 0 to π) evenly for $s_{\mathrm{t}} = 1, \ldots, S_{\mathrm{t}}$ teachers. Consequently, we generate $S = S_{\mathrm{A}} \times S_{\mathrm{t}}$ points of discrete teacher–signal sets $\hat{z}_{is}$ $(s = 1, 2, \ldots, S)$ as

$$z_{is} \equiv \frac{s_{\mathrm{A}}}{S_{\mathrm{A}} + 1}\exp\left(\jmath\left(\frac{s_{\mathrm{t}}}{2S_{\mathrm{t}}} + \frac{i}{I}\right)2\pi\right). \tag{29}$$

Note that the wavelength and, thus, the signal frequency are unchanged. Fig. 1, referred to previously, includes the manner of the amplitude variation. We add WGN to the sinusoidal wave with various weightings. The noise power is adjusted depending on the signal power and the expected SNR, which is determined in each learning trial.

The dots on the continuous signals in Fig. 4 indicate the discrete teacher signal points $\hat{z}_{is}$. We observe the generalization characteristics by inputting signals other than the teachers and evaluate the output errors. Fig. 4(a) illustrates the observation of outputs when the input signal is shifted in time. The continuous time signal was generated by Lagrange interpolation. Fig. 4(b) shows the observation when the amplitude is changed. We combine the time shift and the amplitude change to evaluate the generalization. In the experiment below, $S_{\mathrm{A}} = 4$, $S_{\mathrm{t}} = 4$, and the neural network parameters are listed in Table I. The learning iteration is 3000.

2) Heterodyne Signal Experiment: We process a heterodyne signal observed in a sonar imaging system. The signal has a carrier of 100 kHz with thermal noise. It is converted into 100-Hz in-phase and quadrature-phase (IQ) intermediate-frequency (IF) signals through an IQ mixer. The imbalance of the IQ mixer is less than 0.3 dB in amplitude and 3° in phase,

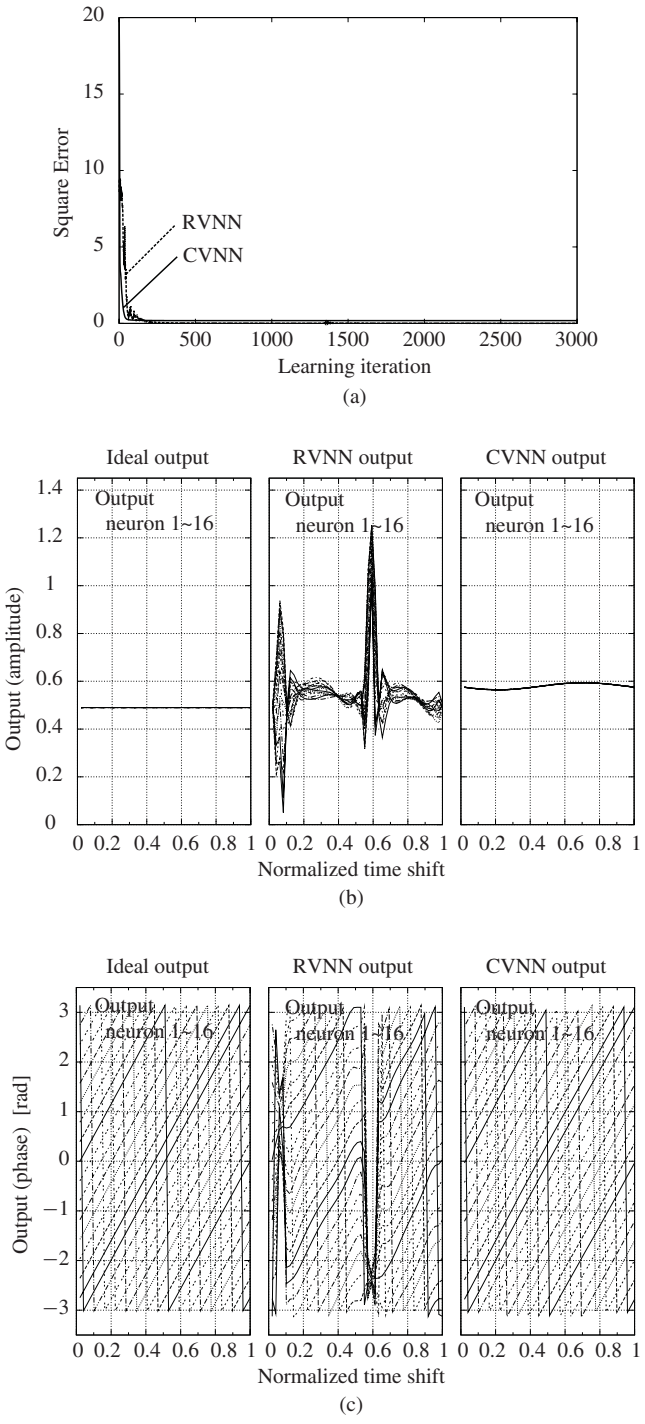


Fig. 5. Example of (a) learning curves and corresponding output (b) amplitude and (c) phase when the input signal gradually sifts in time in the real-valued and complex-valued neural networks (RVNN and CVNN) when no noise is added to sinusoidal signals ($SNR = \infty$).

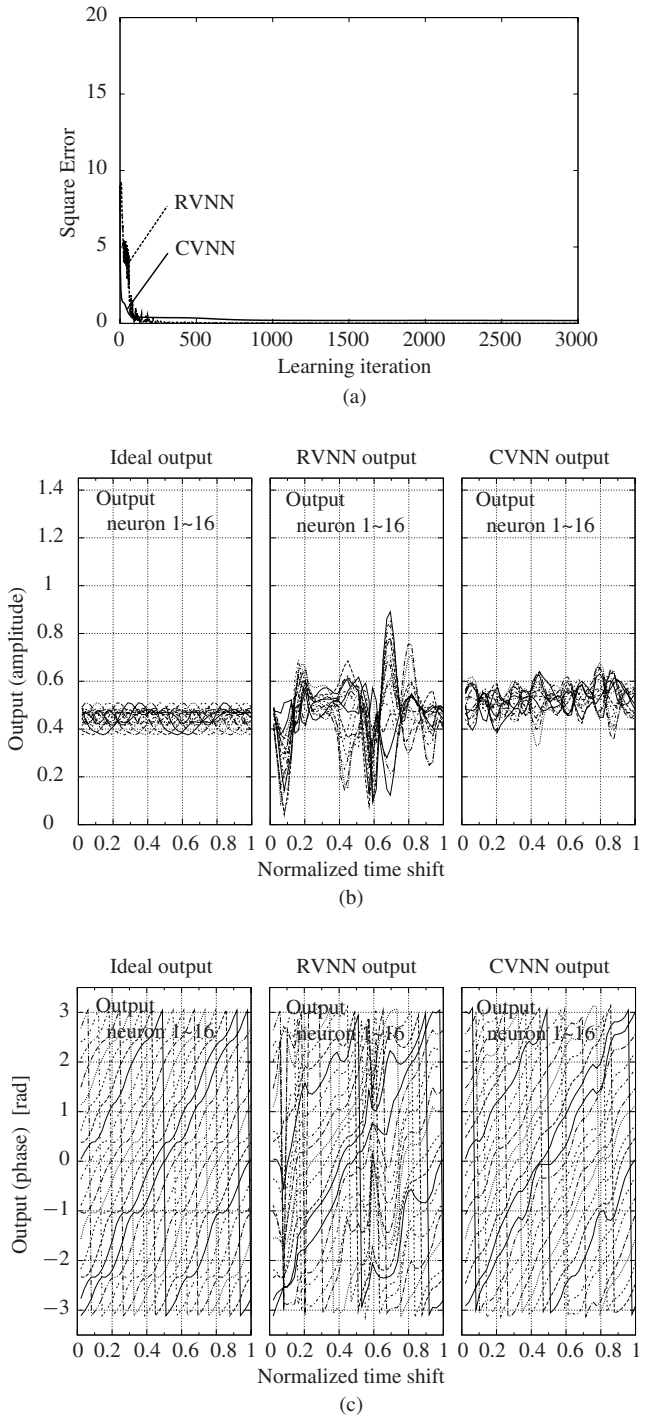


Fig. 6. Example of (a) learning curves and corresponding output (b) amplitude and (c) phase when the input signal gradually sifts in time in the real-valued and complex-valued neural networks (RVNN and CVNN) when white noise is added to sinusoidal signals ($SNR = 20$ dB).

which is common in this type of system. The IF signal is recorded by a personal computer through an analog/digital converter with a sampling frequency of 600 000 samples per second. We aim at appropriate interpolation of the signals in time and/or space domain for postprocessing to generate high-quality time-space images. When the 100-kHz carrier signal power changes, the SNR also changes for a constant noise power.

B. Results

1) Examples of Learning Curves and Output Signals for Inputs Having Various Coherence Degrees: Figs. 5–8 display typical examples of the learning curves and output signals of the CVNN and RVNN for a single learning trial when $SNR = \infty$, 20 dB, 10 dB, and 0 dB, respectively. Fig. 5(a) shows the learning curve when $SNR = \infty$, i.e., the signal is sinusoidal.

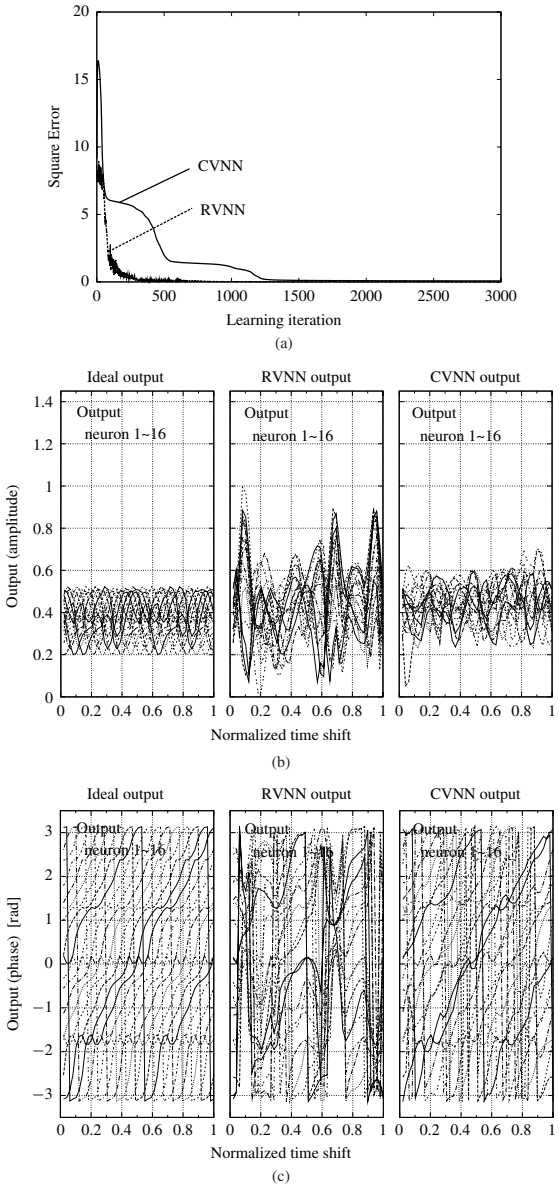


Fig. 7. Example of (a) learning curves and corresponding output (b) amplitude and (c) phase when the input signal gradually sifts in time in the real-valued and complex-valued neural networks (RVNN and CVNN) when white noise is added to sinusoidal signals ($SNR = 10$ dB).

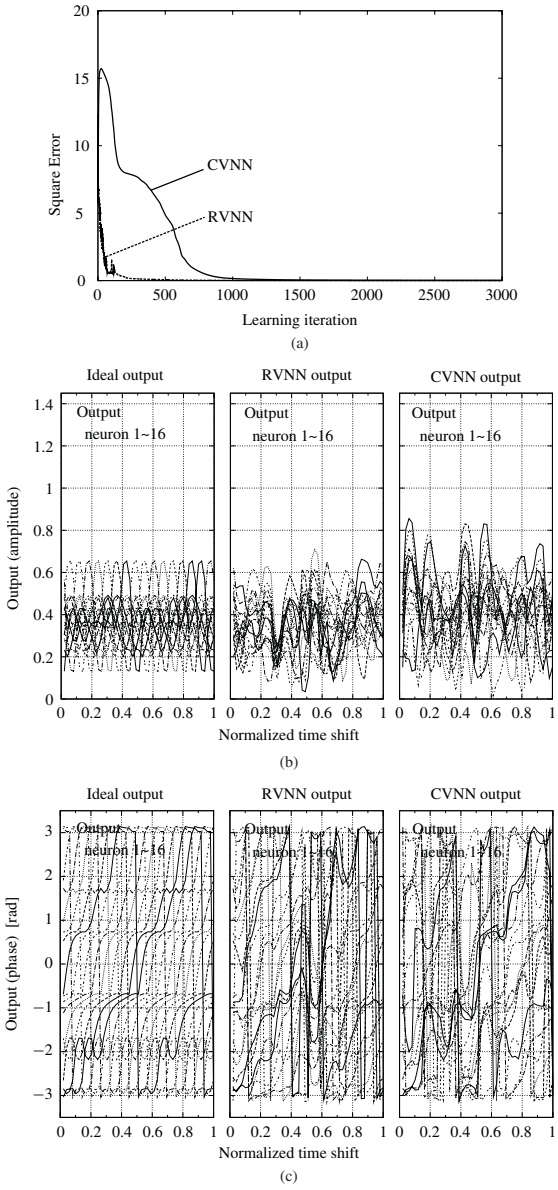


Fig. 8. Example of (a) learning curves and corresponding output (b) amplitude and (c) phase when the input signal gradually sifts in time in the real-valued and complex-valued neural networks (RVNN and CVNN) when white noise is added to sinusoidal signals ($SNR = 0$ dB).

We find that the learning is successfully completed for both CVNN and RVNN. The learning errors converge almost at zero, which means that there is only a slight residual error at the learning teacher points.

After the learning, we use other input signals to investigate the generalization. As mentioned previously, the wavelength is adjusted to span over the 16 neural input terminals. For example, we gradually move the input signal forward in time while keeping the amplitude unchanged at $a = 0.5$. Fig. 5(b) and (c) present the output amplitude and phase, respectively, showing from left-hand side to the right-hand side the ideal output of the identity mapping, the RVNN outputs, and CVNN outputs of the 16 output neurons. The horizontal axes present the time shift t normalized by the unit-wave duration.

In Fig. 5(b), we find that the output signals of the RVNN locally deviate greatly from the ideal ones. The learning points are plotted at $t = 0$ (no time shift), where the output amplitude is almost 0.5 for all the neurons. However, with the time course, the amplitude values fluctuate largely. Contrarily, the CVNN amplitude stays almost constant. At the learning point $t = 0$, the value is slightly larger than 0.5, corresponding to the slight nonzero value of the residual error in the learning curve.

In Fig. 5(c), the ideal output phase values on the left-hand side exhibit linear increase in time. In the RVNN case, though the phase values at $t = 0$ are the same as those of ideal outputs, the values sometimes swing strongly. In contrast, the CVNN output phase values increase in an orderly manner, which is almost identical with the ideal values. In summary,

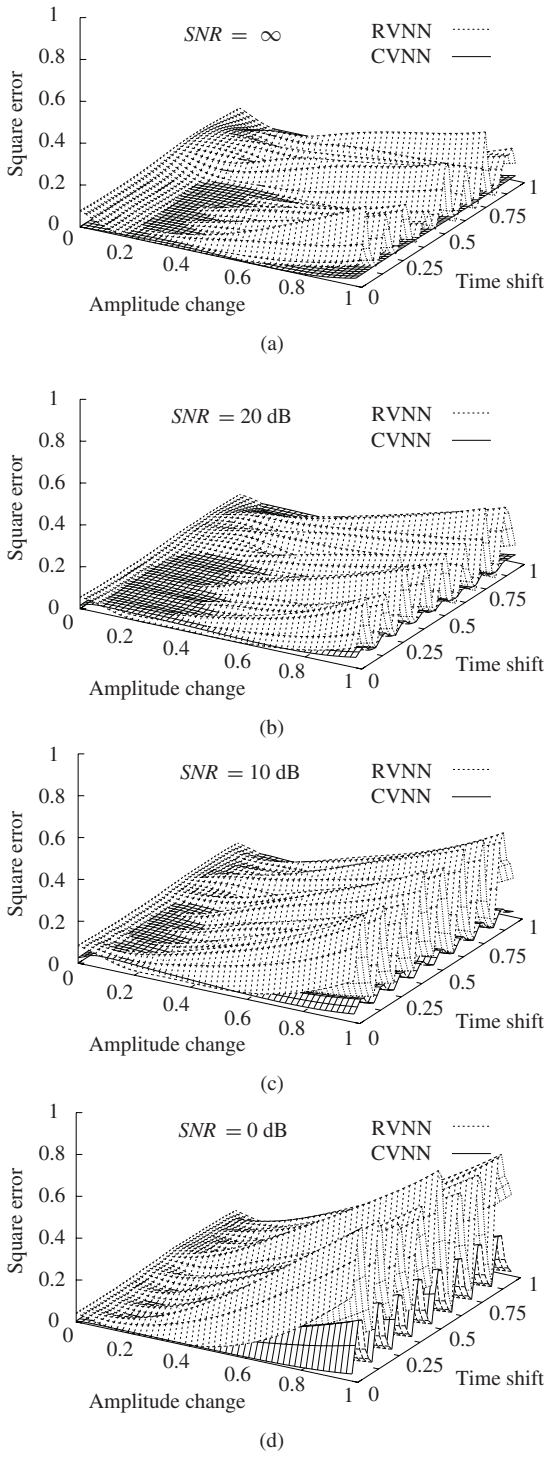


Fig. 9. Squared generalization errors averaged for 100 trials as functions of amplitude change and time shift for SNR = (a) ∞, (b) 20 dB, (c) 10 dB, and (d) 0 dB.

the CVNN presents much better generalization characteristics than the RVNN when the coherence is high, i.e., $SNR = \infty$.

Figs. 6–8 show the data for SNR = 20, 10, and 0 dB, respectively. As the coherence decreases, the generalization error increases. However, in any SNR case, both the amplitude and phase of the CVNN exhibit better generalization than the RVNN.

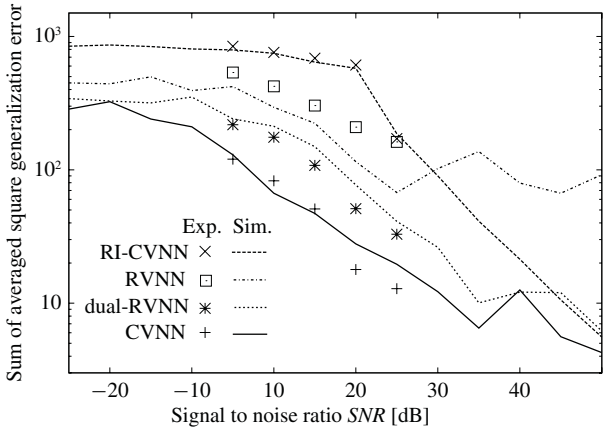


Fig. 10. Squared generalization errors summed up for all the sampling amplitude-time points shown in Fig. 9 versus signal SNR for the real-valued and complex-valued neural networks (CVNN, RI-CVNN, RVNN, and dual-RVNN, curves: simulations, marks: experiments).

The learning curves of the CVNN show that its learning speed depends on the signal coherence. The CVNN learning is completed fast when the coherence is high (about $SNR > 20$ dB), while it takes a longer time when the coherence is low ($SNR < 10$ dB). Contrarily, the learning speed of the RVNN is much less dependent on the coherence. We conducted simulations also for the RI-CVNN and the dual-RVNN. In the case of RI-CVNN, the convergence rate decreases largely in the low coherence region ($SNR < 5$ dB). In the dual-RVNN case, the learning curves are similar to those of RVNN.

2) Generalization Error and its Dependence on the Coherence: Here we present statistic results obtained by repeating the above simulations as well as the real-world experiment dealing with the heterodyne signals explained in Section III-A.

Fig. 9 is 3-D representation of the square errors as the average of 100 learning trials for various coherence degrees, namely, SNR = (a) ∞, (b) 20 dB, (c) 10 dB, and (d) 0 dB, as functions of time shift and amplitude change. The learning points exist at $t = 0$ and amplitude values of $a = 0.2$, 0.4, 0.6, and 0.8. At these points, we can find that the errors are very small, which corresponds to the almost zero residual errors in the learning curves. However, the errors at the teacher points for lower SNRs are obviously positive. This is because the learning error in some trials fails to converge at zero. As a whole, we notice in Fig. 9 that the generalization errors of the RVNN are larger than those of the CVNN, in particular in the cases of higher SNR. When SNR is low ($\sim$0 dB), the error of the CVNN also increases.

Fig. 10 compares quantitatively the generalization errors, summed up for all the sampling amplitude–phase points shown in Fig. 9, for the CVNN, RI-CVNN, RVNN, and dual-RVNN as functions of the coherence degree, i.e., SNR. The four curves show the results of the simulation, while the marks indicate experimental results. In all the neural network cases, the generalization error reduces according to the increase of the coherence (increase of SNR). The CVNN curve shows lower errors than other network curves over a wide range of SNR. The dual-RVNN also shows low errors, though at

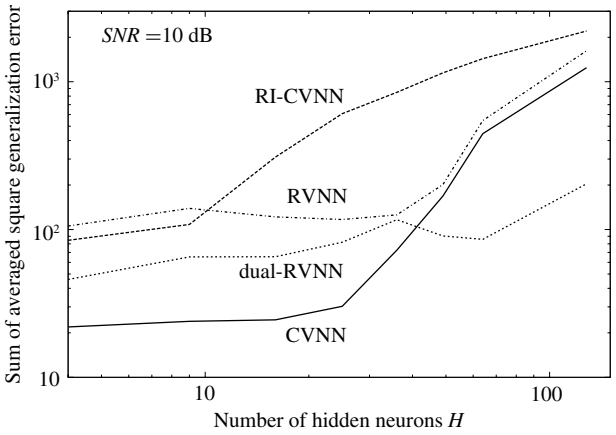


Fig. 11. Squared generalization errors summed up for all the sampling amplitude-time points shown in Fig. 9 versus the number of hidden neurons when $SNR = 10$ dB for the real-valued and complex-valued neural networks (CVNN, RI-CVNN, RVNN, and dual-RVNN).

the middle SNR ($SNR = -5$ to 15 dB) the value is 3–6 dB larger than for CVNN. The error of the simulated RVNN is about 2 dB larger than the dual-RVNN in the low and middle SNR range. The experimental results (marks) of the RVNN are a little larger. It is remarkable that, in the higher coherence region ($SNR > 10$ dB), the RVNN curve holds a floor at a nonnegligible level. The RI-CVNN shows a large generalization error in the low coherence region. This is not only because of the errors at non-teacher points but also because of the errors at teacher points. That is, the learning sometimes fails. In the high coherence region ($SNR > 20$ dB), however, the generalization error decreases and approaches the curves of the CVNN and dual-RVNN. In summary, we found that the four neural networks present different generalization characteristics among them. The experimental results have been found mostly close to the simulation results.

Fig. 11 presents the dependence of the generalization error on the number of hidden neurons H. The SNR is $SNR = 10$ dB where the coherence is high and we generally need only a small number of hidden neurons. The numbers of input terminals and output neurons are fixed as $I = 16$ and $O = 16$. The CVNN shows low generalization error over a wide region of H. However, when H is extremely large ($H > 50$), the error increases. The RVNN has larger error, but the increase tendency is almost the same as that of CVNN. The RI-CVNN has almost the same large error in small H region, and shows an earlier rise versus the increase of H. In contrast, dual-RVNN has only less increase, though the error level is not small.

IV. Discussion

The experimental results shown in Section III revealed the following characteristics of the networks.

1) Learning speed: the CVNN has a high learning speed for high coherence signals, but low speed for low coherence signals. The speed of the RVNN has a smaller dependence on the coherence.
2) Generalization error:
   a) The CVNN has 3–6 dB lower generalization error than dual-RVNN in the middle coherence region (-5 dB $< SNR < 20$ dB).
   b) The RVNN has a high error floor in the high coherence region ($SNR > 15$ dB).
   c) The RI-CVNN has higher error in the low coherence region ($SNR < 10$ dB).

These characteristics are explained by taking into consideration the degree of freedom in the learning process and the circularity of signals, as well as the neural dynamics below.

A. Degree of Freedom in Learning in CVNN, Dual-RVNN, and RVNN

What is the most specific nature of a complex number? As we focus on multiplication out of the four arithmetic operations of complex numbers, we can represent a complex number as a real 2×2 matrix. That is, with every complex number $c = a + \jmath b$, where a and b are real numbers, we associate a $\boldsymbol{C}$-linear transformation T_c of $z = x + \jmath y$ as

$$T_c : \boldsymbol{C} \to \boldsymbol{C}, \qquad z \mapsto cz = ax - by + \jmath(bx + ay). \quad (30)$$

If we identify $\boldsymbol{C}$ with $\boldsymbol{R}^2$ by

$$z = x + \jmath y \;=\; \begin{pmatrix} x \\ y \end{pmatrix} \quad (31)$$

it follows that:

$$T_c \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax - by \\ bx + ay \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \quad (32)$$

In other words, the linear transformation T_c determined by $c = a + \jmath b$ is expressed by a matrix which means phase rotation and magnitude amplification or attenuation as

$$\begin{pmatrix} a & -b \\ b & a \end{pmatrix} = r \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad (33)$$

where $r \equiv \sqrt{a^2 + b^2}$ and $\theta \equiv \arctan b/a$ denote amplification or attenuation of amplitude and rotation angle applied to the complex signal z, respectively.

Let us consider how the above feature of the complex number emerges in neural dynamics [21]. Assume a very simple case shown in Fig. 12(a) where we have a single-layer two-input two-output feedforward neural network in the real number domain. For simplicity, we omit possible nonlinearity at the neurons, i.e., the activation function is the identity function. The task for the network here is to realize a mapping that transforms an input $\boldsymbol{x}^{\mathbf{IN}}$ to an output $\boldsymbol{x}^{\mathbf{OUT}}$ through supervised learning that adjusts the synaptic weights w_{ji}. Simply, we have only a single teacher pair of input and output signals here. Then the general input–output relationship is described by using four real numbers a, b, c, and d as

$$\begin{pmatrix} x_1^{\mathbf{OUT}} \\ x_2^{\mathbf{OUT}} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1^{\mathbf{IN}} \\ x_2^{\mathbf{IN}} \end{pmatrix}. \quad (34)$$

In the present case, we have a variety of possible mappings realized by the learning because the number of parameters to

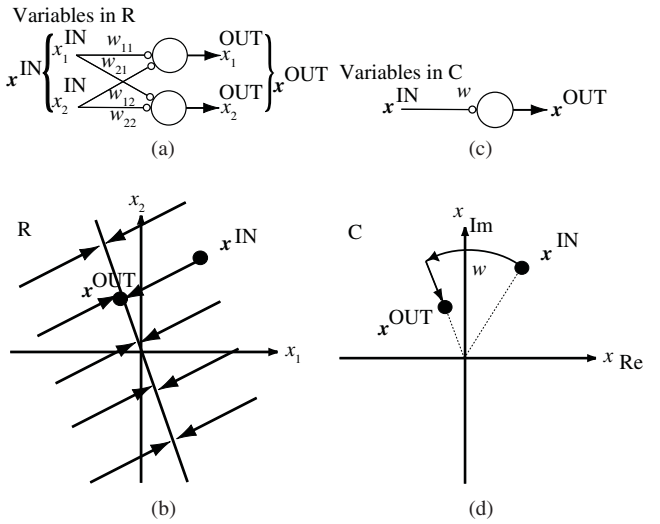


Fig. 12. Simple single-layer neural network to learn a task that maps $\boldsymbol{x}^{\mathbf{IN}}$ to $\boldsymbol{x}^{\mathbf{OUT}}$. (a) Real-valued bivariate network structure in $\boldsymbol{R}^2$. (b) Possible but degenerate solution that is often not useful. (c) Complex-valued network structure in $\boldsymbol{C}$. (d) Solution obtained in this small dgree-of-freedom case [21].

be determined is larger than the number of conditions, i.e., the learning task is an ill-posed problem. The functional difference among the possible mapping emerges as the difference in the generalization characteristics. For example, learning can result in a degenerate mapping shown in Fig. 12(c), which is often not useful in practice. Such degeneracy is suggested in the present simulation in Fig. 5(b) in the RVNN output as the bottleneck-shaped forced consistency of the amplitude curves.

In parallel, let us consider the learning of the mapping in the complex domain, which transforms a complex value $\boldsymbol{x}^{\mathbf{IN}} = (x_{\mathrm{Re}}^{\mathbf{IN}}, x_{\mathrm{Im}}^{\mathbf{IN}})$ to another complex value $\boldsymbol{x}^{\mathbf{OUT}} = (x_{\mathrm{Re}}^{\mathbf{OUT}}, x_{\mathrm{Im}}^{\mathbf{OUT}})$. Fig. 12(c) shows a complex-valued network, where the weight is a single complex value $w = |w| \exp(\jmath\theta)$. The situation is expressed just like in (34) with the constraint (33) as

$$\begin{pmatrix} x_{\mathrm{Re}}^{\mathbf{OUT}} \\ x_{\mathrm{Im}}^{\mathbf{OUT}} \end{pmatrix} = \begin{pmatrix} |w|\cos\theta & -|w|\sin\theta \\ |w|\sin\theta & |w|\cos\theta \end{pmatrix} \begin{pmatrix} x_{\mathrm{Re}}^{\mathbf{IN}} \\ x_{\mathrm{Im}}^{\mathbf{IN}} \end{pmatrix}. \quad (35)$$

The degree of freedom is reduced, and the arbitrariness of the solution is also reduced. Fig. 12(d) illustrates the result of the learning. The mapping is a combination of phase rotation and amplitude attenuation. This property can be a great advantage when we deal with coherent wave information such as electromagnetic waves, lightwaves, sonic waves, and electron waves. This property can be named *circularity* in the neural dynamics in analogy to the signal circularity for high coherence.

B. Circularity

The circularity of the signals to be processed is also an important factor. To deepen the discussion, we refer to the wide sense linear (or widely linear: WL) systems which introduce conjugate signals in addition to direct complex signals [23], [24]. WL systems well learn complex data distributed anisotropically in the complex plane, i.e., noncircular data. For example, it is useful for predicting wind strength and direction by assuming the axes of the complex number plane represent north, south, east, and west, and the distance from the origin expresses the strength. Augmented CVNNs have been proposed in such a context [25]. Wind has high anisotropy in general. The augmented complex-valued networks do not lead to the reduction of the degree of freedom. The degree is the same as that of real-valued networks, resulting in a dynamics similar to that of real-valued ones [26].

Fig. 13 is a conceptual illustration showing the bases of the respective networks. The number of bases of the augmented complex networks turns back to that of the real-valued networks, and its dynamics approaches that of real networks. This situation is analogous to the fact that the combination of positive and negative frequency spectra generates almost real-valued signals. In other words, if we compare the relationship to polarization of lightwaves, we come to the following conclusion. CVNNs deal with only right- or left-handed circular polarized light, which are suitable for circular signal processing. Note that the signal in total can be out of complete circularity, but only each frequency component has circularity. Since any waveform can be synthesized by sinusoidal components through Fourier synthesis, the signals that the complex networks can deal with are not limited to completely coherent signals. In contrast, the augmented complex-valued networks deal with both the right- and left-handed circular polarized light. They are more flexible because of the larger degree of freedom, which is too much for circular signals. Dual univariate networks have the same degree of freedom, though in this case the bases are linear polarization corresponding to the real and imaginary parts instead of the right- and left-handed circular bases in the augmented networks. In this manner, they are similar to each other.

Consequently, CVNNs are suitable for processing analytic signals, which consist of a real component and its consistent imaginary component that has the same amplitude but a phase shifted by 90°. Analytic signals are essentially circular, existing widely in electronics, for example, at the output of heterodyne or homodyne mixers and at the output of digital signal processing using the Hilbert transform. Complex-valued networks have the ability to process such analytic signals appropriately.

Noncircular signals are observed in wind processing, but less in electronics dealing with electromagnetic wave and related time-sequential signals. The reason lies in the fact that phase does not have any meaning in its absolute value, but in its difference from a certain reference. However, a few exceptions may exist. For example, signals in offset quadrature phase shift keying modulation may include nonnegligible noncircularity generated by highly unbalanced electronics.

C. Differences in Learning Speed and Generalization Errors

Based on the considerations on the degree of freedom in learning in Section IV-A and the circularity in signals and network dynamics in discussed in Section IV-B, we can summarize the experimental results as follows.

1) The CVNN shows quick learning and a small generalization error in the middle and high coherence region because of its smaller degree of freedom and the circular dynamics. In the case of an extremely large number

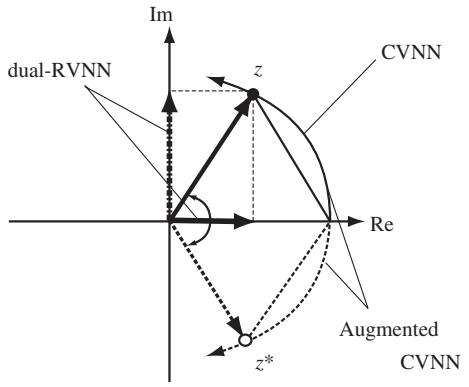


Fig. 13. Conceptual illustration of the relationship between bases in the respective neural networks to deal with complex signal z.

of hidden neurons, however, it cannot maintain the small degree of freedom, resulting in a larger generalization error. This phenomenon originates also from the circularity, i.e., the origin-point symmetry, which is similar to the degeneration of attractors in complex-valued associative memories. Because of the circularity, connections can remain excessively to express coherent signals.

2) The dual-RVNN has a larger degree of freedom and a larger generalization error than the CVNN. In this interpolation task, however, the network can get a solution by looking up the unit-wavelength past signal (or half-wavelength past signal with reversing its sign) when the signal has very high coherence and therefore the time delay directly corresponds to the phase delay. At other points, the neural output has no correlation with the input, for which the neural connections decay. In other words, the signal degeneration does not occur on the real–imaginary bases. Then the generalization error stays almost constant against the increase of the number of hidden neurons.
3) The RVNN has further larger degrees of freedom, resulting in a larger generalization error.
4) The RI-CVNN includes two properties inconsistent with each other, namely, the circular process of the weight multiplication and the anisotropy of activation function, which lead to a large generalization error. The error is enhanced when the signal has smaller relationship between the time delay and the phase shift because of the noise. However, for high coherence signals, the anisotropy works to reduce the harmful excess connections just like in the RVNN case, and makes the generalization error small.

V. Conclusion

This paper investigated numerically the generalization characteristics in the feedforward CVNN and RVNN with their variations. We focused on their dependence on the coherence degree. We compared CVNN (with amplitude–phase-type activation function), RI-CVNN (with real–imaginary-type activation function), RVNN (bivariate), and dual-RVNN (dual univariate) in a case where the networks approximate functions. Computer experiments demonstrated that the CVNN exhibits smaller generalization error in particular for signals having high coherence. This result is attributed to the smaller degree of freedom and the coherence-compatible circularity in the neural dynamics. We discussed the relationship between the generalization characteristics and the dynamics of the various CVNNs and RVNNs in general by focusing on the degree of freedom and the circularity.

Acknowledgment

The authors would like to acknowledge the insightful suggestions of the anonymous reviewers.

References

[1] D. L. Birx and S. J. Pipenberg, "A complex mapping network for phase sensitive classification," *IEEE Trans. Neural Netw.*, vol. 4, no. 1, pp. 127–135, Jan. 1993.
[2] T. Hara and A. Hirose, "Plastic mine detecting radar system using complex-valued self-organizing map that deals with multiple-frequency interferometric images," *Neural Netw.*, vol. 17, nos. 8–9, pp. 1201–1210, 2004.
[3] S. Masuyama and A. Hirose, "Walled LTSA array for rapid, high spatial resolution, and phase-sensitive imaging to visualize plastic landmines," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2536–2543, Aug. 2007.
[4] I. Aizenberg, D. V. Paliy, J. M. Zurada, and J. T. Astola, "Blur identification by multilayer neural network based on multivalued neurons," *IEEE Trans. Neural Netw.*, vol. 19, no. 5, pp. 883–898, May 2008.
[5] G. Tanaka and K. Aihara, "Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1463–1473, Sep. 2009.
[6] Y. Zhang and Y. Ma, "CGHA for principal component extraction in the complex domain," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1031–1036, Sep. 1997.
[7] H. Sawada, R. Mukai, S. Araki, and S. Makino, "Polar coordinate based nonlinear function for frequency-domain blind source separation," *IEICE Trans. Fundam. Electron., Commun., Comput. Sci.*, vol. E86A, no. 3, pp. 590–596, 2003.
[8] S. Goh and D. P. Mandic, "Nonlinear adaptive prediction of complex-valued signals by complex-valued PRNN," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1827–1836, May 2005.
[9] A. Hirose and R. Eckmiller, "Behavior control of coherent-type neural networks by carrier-frequency modulation," *IEEE Trans. Neural Netw.*, vol. 7, no. 4, pp. 1032–1034, Jul. 1996.
[10] A. B. Suksmono and A. Hirose, "Beamforming of ultrawideband pulses by a complex-valued spatio-temporal multilayer neural network," *Int. J. Neural Syst.*, vol. 15, no. 1, pp. 1–7, 2005.
[11] A. Hirose, "Applications of complex-valued neural networks to coherent optical computing using phase-sensitive detection scheme," *Inf. Sci. - Appl.*, vol. 2, no. 2, pp. 103–117, Sep. 1994.
[12] S. Kawata and A. Hirose, "A coherent optical neural network that learns desirable phase values in frequency domain by using multiple optical-path differences," *Opt. Lett.*, vol. 28, no. 24, pp. 2524–2526, 2003.
[13] A. Hirose, T. Higo, and K. Tanizawa, "Efficient generation of holographic movies with frame interpolation using a coherent neural network," *IEICE Electron. Exp.*, vol. 3, no. 19, pp. 417–423, 2006.
[14] Y. Nakamiya, M. Kinjo, O. Takahashi, S. Sato, and K. Nakajima, "Quantum neural network composed of Kane's qubits," *Jpn. J. Appl. Phys.*, vol. 45, no. 10A, pp. 8030–8034, Oct. 2006.
[15] A. Hirose, Y. Asano, and T. Hamano, "Developmental learning with behavioral mode tuning by carrier-frequency modulation in coherent neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1532–1543, Nov. 2006.
[16] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 2101–2104, Sep. 1991.
[17] N. Benvenuto and F. Piazza, "On the complex backpropagation algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 967–969, Apr. 1992.

[18] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Netw.*, vol. 10, no. 8, pp. 1391–1415, Nov. 1997.
[19] A. Hirose, "Continuous complex-valued back-propagation learning," *Electron. Lett.*, vol. 28, no. 20, pp. 1854–1855, 1992.
[20] G. M. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE Trans. Circuits Syst. II*, vol. 39, no. 5, pp. 330–334, May 1992.
[21] A. Hirose, "Nature of complex number and complex-valued neural networks," *Front. Electr. Electron. Eng. China*, vol. 6, no. 1, pp. 171–180, 2011.
[22] A. Hirose, *Complex-Valued Neural Networks*. New York: Springer-Verlag, 2006.
[23] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 2030–2033, Aug. 1995.
[24] D. P. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters – Noncircularity, Widely Linear and Neural Models*. New York: Wiley, Apr. 2009.
[25] Y. Xia, B. Jelfs, M. M. Van Hulle, J. C. Principe, and D. P. Mandic, "An augumented echo state network for nonlinear adaptive filetering of complex noncircular signals," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 74–83, Jan. 2011.
[26] D. P. Mandic, S. Still, and S. C. Douglas, "Duality between widely linear and dual channel adaptive filtering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Taipei, Taiwan, Apr. 2009, pp. 1729–1732.

Akira Hirose (S'85–M'88–SM'08) received the Ph.D. degree in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He joined the Research Center for Advanced Science and Technology (RCAST), University of Tokyo, in 1987, as a Research Associate. In 1991, he became an Instructor at RCAST. From 1993 to 1995, he joined the Institute for Neuroinformatics, University of Bonn, Bonn, Germany. He is currently a Professor with the Department of Electrical Engineering and Information Systems, University of Tokyo. His current research interests include neural networks and wireless electronics.

Prof. Hirose is a senior member of the Society of Information and Communication Engineers (IEICE) and a member of the Japanese Neural Networks Society (JNNS). He served as the Chair of the Neurocomputing Technical Group, Institute of Electronics, IEICE, and is currently the Vice-President of JNNS, a member of the IEEE Computational Intelligence Society (CIS) Neural Networks Technical Committee, a Steering Committee Member of the IEEE CIS Japan Chapter, and a Governing Board Member of the Asia-Pacific Neural Network Assembly. He serves as Editor-in-Chief of the IEICE *Transactions on Electronics*, and as an Associate Editor of journals such as the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE GEOSCIENCE AND REMOTE SENSING NEWSLETTER.

Shotaro Yoshida was born in Oita, Japan, on November 5, 1988. He received the B.S. degree in electronic and information engineering from the University of Tokyo, Tokyo, Japan, in 2011. He is currently pursuing the M.S. degree.

His current research interests include neural networks and bioelectronics.

Complex-valued neural networks for machine learning on non-stationary physical data

Jesper Sören Dramsch[a,*], Mikael Lüthje[a], Anders Nymark Christensen[a]

[a] *Technical University of Denmark, Kongens Lyngby, Denmark*

Abstract

Deep learning has become an area of interest in most scientific areas, including physical sciences. Modern networks apply real-valued transformations on the data. Particularly, convolutions in convolutional neural networks discard phase information entirely. Many deterministic signals, such as seismic data or electrical signals, contain significant information in the phase of the signal. We explore complex-valued deep convolutional networks to leverage non-linear feature maps. Seismic data commonly has a lowcut filter applied, to attenuate noise from ocean waves and similar long wavelength contributions. Discarding the phase information leads to low-frequency aliasing analogous to the Nyquist-Shannon theorem for high frequencies. In non-stationary data, the phase content can stabilize training and improve the generalizability of neural networks. While it has been shown that phase content can be restored in deep neural networks, we show how including phase information in feature maps improves both training and inference from deterministic physical data. Furthermore, we show that the reduction of parameters in a complex network outperforms larger real-valued networks.

Keywords: Machine Learning, Deep Learning, Neural Networks, Physics-based machine learning, Geophysics, Seismic

1. Introduction

Seismic data is high-dimensional physical data. During acquisition, the data is collected over an area on the Earths surface. This images a 3D cube of the subsurface. Due to low reflection coefficients and low signal-to-noise ratio, the measurements are repeated, while moving over the target area. This provides a collection of illumination angles over a subsurface area. The dimensionality of this data has historically been reduced to a stacked 3D cube or 2D sections for interpreters to be able to grasp the information of the seismic data.

With the recent revolution of image classification, segmentation and object detection through deep learning (Krizhevsky et al., 2012), geophysics has regained interest in automatic seismic interpretation (classification), and analysis of seismic signals. Through transfer learning, several initial successes were presented

*Corresponding author
Email address: jesper@dramsch.net (Jesper Sören Dramsch)

in Dramsch and Lüthje (2018a). Nevertheless, seismic data has its caveats due to the complicated nature of bandwidth-limited wave-based imaging. Common problems are cycle-skipping of wavelets and nullspaces in inversion problems (Yilmaz, 2001).

Automatic seismic interpretation is complicated, as the modelling of seismic data is computationally expensive and often proprietary. Seismic field data is often not available and their interpretation is highly subjective and ground truth is not available. The lack of training data has been delaying the adoption of existing methods and hindering the development of specific geophysical deep learning methods. Incorporating domain knowledge into general deep learning models has been successful in other fields (Paganini et al., 2017).

The state-of-the-art method has been a iterative windowed Fourier transform for phase reconstruction (Griffin and Lim, 1984). Modern neural audio synthesis focuses on methods that do not require explicit reconstruction of the phase (Mehri et al., 2016; van den Oord et al., 2016, 2017; Prenger et al., 2018). Mehri et al. (2016) introduced a recurrent neural network formulation, where van den Oord et al. (2016) reformulated the synthesis network in a strided convolutional network. The original WaveNet formulation in van den Oord et al. (2016) is slow due to the autoregressive filter, warranting the parallel formulation in van den Oord et al. (2017).

We explicitly incorporate phase information in a deep convolutional neural network. These have been heavily explored in the digital signal processing community, before the recent renaissance of neural networks and deep learning. Relevant examples to seismic data processing include source separation (Scarpiniti et al., 2008), adaptive noise reduction (Suksmono and Hirose, 2002), and optical flow (Miyauchi et al., 1993) with complex-valued neural networks. Sarroff (2018) gives a comprehensive overview of applications of complex-valued neural networks in signal and image processing.

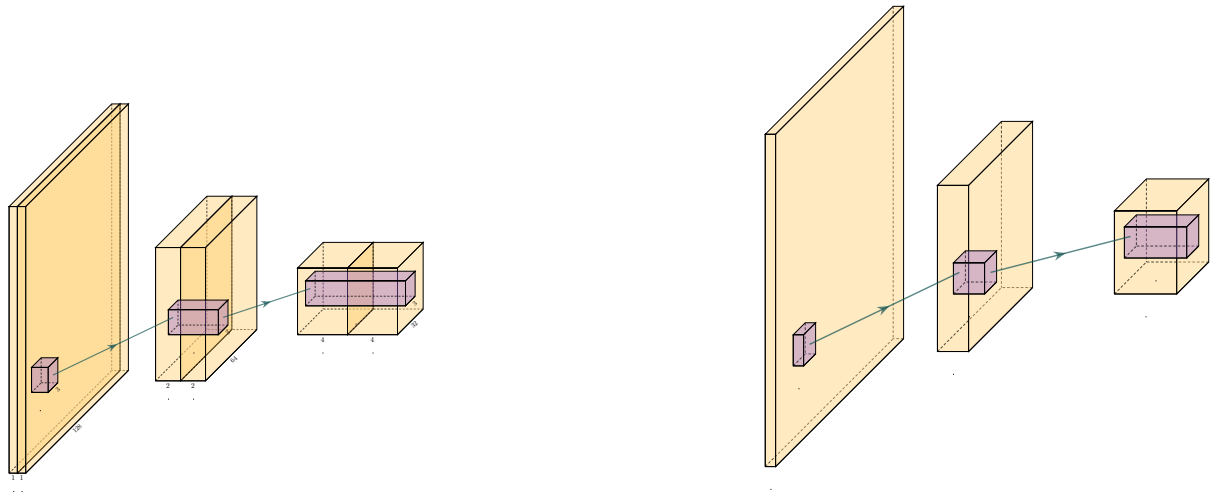
In this work, we calculate the complex-valued seismic trace by applying the Hilbert transform to each trace. Phase information has been shown to be valuable in the processing (Liner, 2002) and interpretation of seismic data (Roden and Sepúlveda, 1999; Mavko et al., 2003). Purves (2014) provides a tutorial that shows the implementation details of Hilbert transforms.

In this paper we give a brief overview of convolutional neural networks and then introduce the extension to complex neural networks and seismic data. We show that including explicit phase information provides superior results to real-valued convolutional neural networks for seismic data. Difficult areas that contain seismic discontinuities due to geologic faulting are resolved better without leakage of seismic horizons. We train and evaluate several complex-valued and real-valued auto-encoders to show and compare these properties. These results can be directly extended to automatic seismic interpretation problems.

2. Complex Convolutional Neural Networks

2.1. Basic principles

Convolutional neural networks (LeCun et al., 1999) use multiple layers of convolution and subsampling to extract relevant information from the data (see Figure 1)



(a) Complex Neural Network

(b) Real Neural Network

Figure 1: Schematic of equivalent complex- and real-valued convolutional neural network

The input image is repeatedly convolved with filters and subsampled. This creates many, but smaller and smaller images. For a classification task, the final step is then a weighting of these very small images leading to a decision about what was in the original image. The filters are learned as part of the training process by exposing the network to training images. The salient point is, that the convolution kernels are learned based on the training. If the goal is - for example - to classify geological facies, the convolutional kernels will learn to extract information from the input, that helps with that task. It is thus a very strong methodology, that can be adapted to many tasks.

2.2. Real- and Complex-valued Convolution

Convolution is an operation on two signals f and g or a signal and a filter that produce a third signal, containing information from both of the inputs. An example is the moving average filter, which smoothes the input, acting as a low-pass filter. Convolution is defined as

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau, \tag{1}$$

at the location τ. While often applied to real value signals, convolution can be used on complex signals. For the integral to exist both f and g must decay when approaching infinity. Convolution is directly generalizable to N-dimensions by multiple integrations and maintains commutativity, distributivity, and associativity. In digital signals this extends to discrete values by replacing the integration with summation.

2.3. Complex Convolutional Neural Networks

Complex convolutional networks provide the benefit of explicitly modelling the phase space of physical systems (Trabelsi et al., 2017). The complex convolution introduced in Section 2.2, can be explicitly implemented as convolutions of the real and complex components of both kernels and the data. A complex-valued data matrix in cartesian notation is defined as $\mathbf{M} = M_{\Re} + iM_{\Im}$ and equally, the complex-valued convolutional

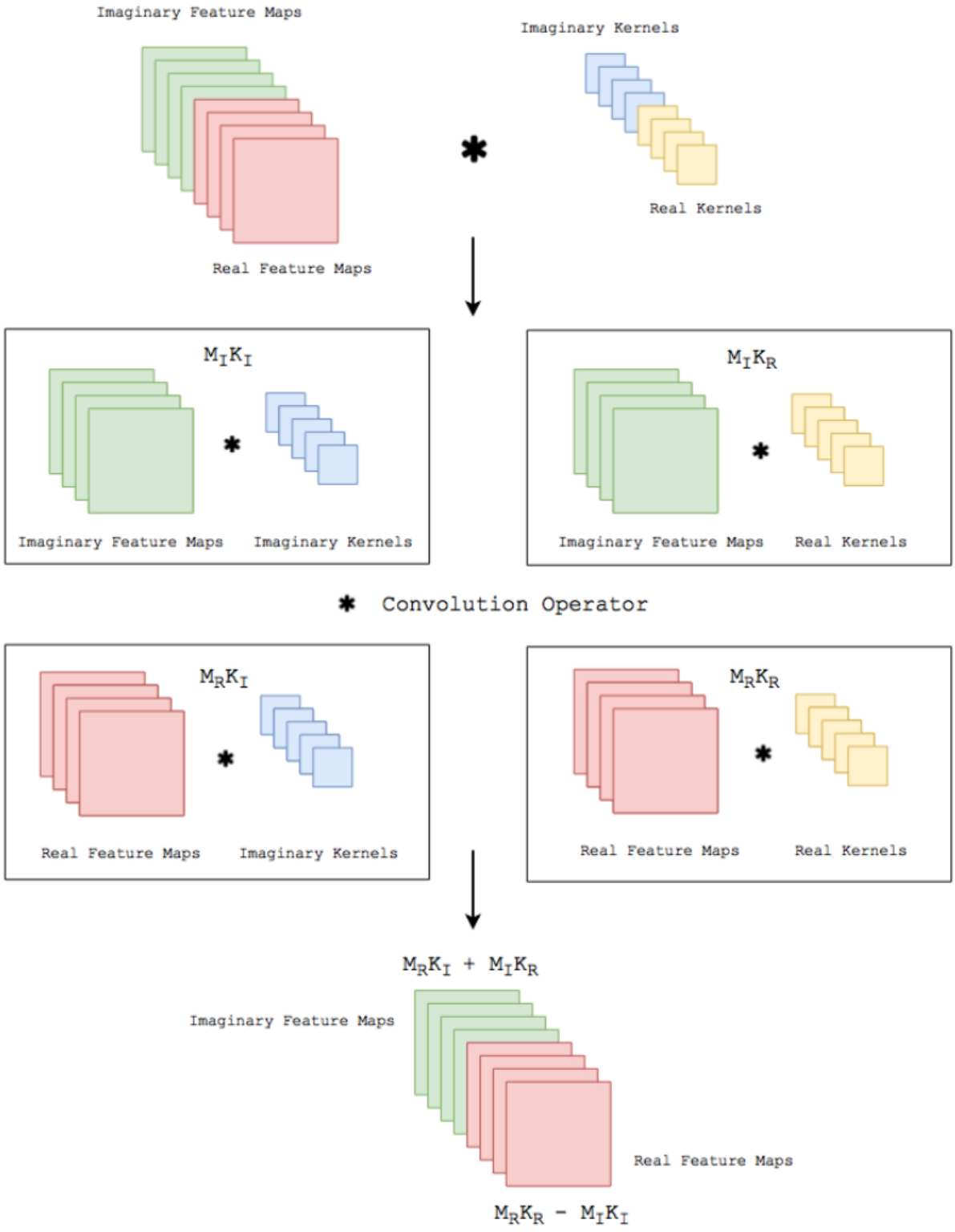


Figure 2: Implementation details of Complex Convolution CC-BY (Trabelski et al. 2017).

kernel is defined as $\mathbf{K} = K_\Re + iK_\Im$. The individual coefficients $(M_\Re, M_\Im, K_\Re, K_\Im)$ are real-valued matrices, considering vectors are special cases of matrices with one of two dimensions being one.

Solving the convolution of

$$M' = K * M = (M_\Re + iM_\Im) * (K_\Re + iK_\Im), \tag{2}$$

we can apply the distributivity of convolutions (cf. section 2.2) to obtain

$$M' = \{M_\Re * K_\Re - M_\Im * K_\Im\} + i\{M_\Re * K_\Im + M_\Im * K_\Re\}, \tag{3}$$

where K is the Kernel and M is a data vector (see Figure 2).

We can reformulate this in algebraic notation

$$\begin{bmatrix} \Re\{M * K\} \\ \Im\{M * K\} \end{bmatrix} = \begin{bmatrix} K_\Re & -K_\Im \\ K_\Im & K_\Re \end{bmatrix} * \begin{bmatrix} M_\Re \\ M_\Im \end{bmatrix} \tag{4}$$

Complex convolutional neural networks learn by back-propagation. Sarroff et al. (2015) state that the activation functions, as well as the loss function must be complex differentiable (holomorphic). Trabelsi et al. (2017) suggest that employing complex losses and activation functions is valid for speed, however, refers that Hirose and Yoshida (2012) show that complex-valued networks can be optimized individually with real-valued loss functions and contain piecewise real-valued activations. We reimplement the code Trabelsi et al. (2017) provides in keras (Chollet et al., 2015) with tensorflow (Abadi et al., 2015), which provides convenience functions implementing a multitude of real-valued loss functions and activations.

While common up- and downsampling functions like MaxPooling, UpSampling, or striding do not suffer from complex-valued neural networks, batch normalization (BN) (Ioffe and Szegedy, 2015) does. Real-valued batch normalization normalizes the data to zero mean and a standard deviation of 1. This does not guarantee normalization in complex values. Trabelsi et al. (2017) suggest implementing a 2D whitening operation as normalization in the following way.

$$\widetilde{x} = V^{-\frac{1}{2}}(x - \mathbb{E}[x]), \tag{5}$$

where x is the data and V is the 2x2 covariance matrix, with the covariance matrix being

$$V = \begin{bmatrix} V_{\Re\Re} & V_{\Re\Im} \\ V_{\Im\Re} & V_{\Im\Im} \end{bmatrix} \tag{6}$$

Effectively, this multiplies the inverse of the square root of the covariance matrix with the zero-centred data. This scales the covariance of the components instead of the variance of the data (Trabelsi et al., 2017).

2.4. Auto-encoders

Auto-encoders (Hinton and Salakhutdinov, 2006) are a special configuration of the encoder-decoder network that map data to a low-level representation and back to the original data. This low-level representation is often called bottleneck or code layer. Auto-encoder networks map $f(x) = x$, where x is the data and f

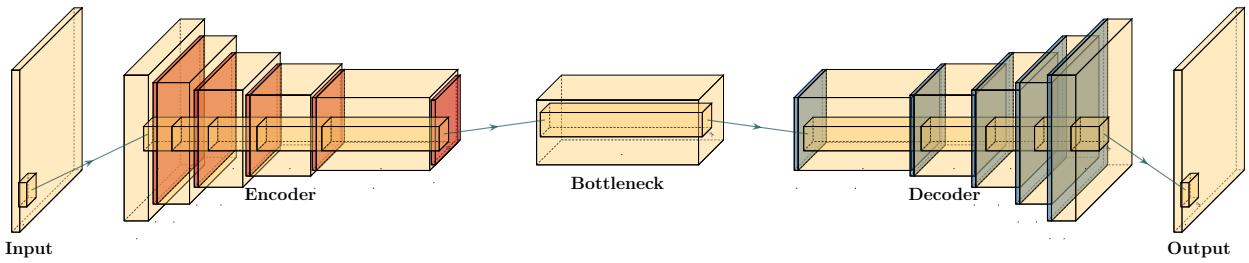


Figure 3: Typical autoencoder architecture. The data is compressed to a low dimensional bottleneck, and then reconstructed.

is an arbitrary network. The architecture of auto-encoders is an example of lossy compression and recovery from the lossy representation. Commonly, recovered data is blurred by this process.

The principle is illustrated in figure 3. The input is transformed to a low-dimensional representation - called a code or latent space - and then reconstructed again from this low dimensional representation. The intuition is, that the network has to extract the most salient parts from the data, to be able to perform a reconstruction. As opposed to other methods for dimensionality reduction - e.g. principal component analysis - an auto-encoder can find a non-linear representation of the data. The low-dimensional representation can then be used for anomaly detection, or classification.

3. Aliasing in Patch-based training

3.1. Mean-Shift in Neural Networks

A single neuron in a neural network can be described by $\sigma(w \cdot x + b)$, where w is the network weights, x is the input data, b is the network bias, and σ is a non-linear activation function. During training, the network weights w and biases b are are adjusted to a value that represents the training minimum. Learning on a mean-shift of q of an arbitrary distribution over x leads to $\sigma(w \cdot (x + q) + b)$, which increases the neuron response by q, weighted by w. During inference, both w and b are fixed, by extension the mean-shift q is fixed as well. The mean-shift over larger inference data disappears, introducing an additional bias of $w \cdot q$ before non-linear activation. This training bias may lead to prediction errors of the neuron and consequently the full neural network.

3.2. Windowed Aliasing

Non-stationary data such as seismic data can contain sections within the data that contain spurious offsets from the mean. Figure 4 shows varying sizes of cutouts, with 101 and 256 samples respectively. In the middle, the full normalised amplitude spectra are presented. On the right, the corresponding phase spectra are presented. On the left, we focus on the frequency content of the amplitude spectra around 0 Hz. The cutouts were Hanning tapered, however, a mean shift appears with decreasing patch size.

These concepts of mean-shift corresponds to a DC offset in spectral data, which can be audio, seismic or electrical data. In images this corresponds to a non-zero alpha channel. While batch normalization can

correct the mean shift in individual mini-batches (Ioffe and Szegedy, 2015), this may shift the entire spectrum by the aliased offset. Additionally, batch normalization may not be feasible in some physical applications pertaining to regression tasks.

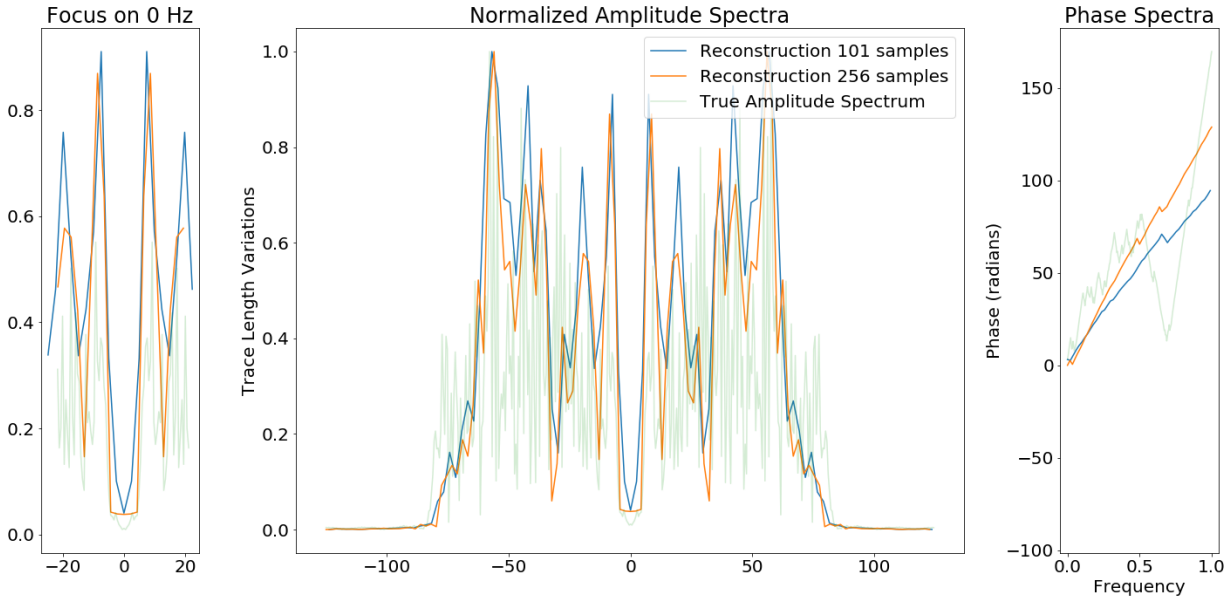


Figure 4: Spectral aliasing dependent on window-size (from Dramsch and Lüthje (2018b))

4. Complex Seismic Data

Complex seismic traces are calculated by applying the Hilbert transform to the real-valued signal. The Hilbert transform applies a convolution with to the signal, which is equivalent to a -90-degree phase rotation. It is essential that the signal does not contain a DC component, as this would not have a phase rotation.

The Hilbert transform is defined as

$$H(u)(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} \, d\tau, \tag{7}$$

of a real-valued time series $u(t)$, where the improper integral has to be interpreted as the Cauchy principal value. In the Fourier domain, the Hilbert transform has a convenient formulation, where frequencies are set zero and the remaining frequencies are multiplied by 2. This can be written as

$$x_a = F^{-1}(F(x)2U) = x + iy \tag{8}$$

where x_a is the analytical signal, x is the real signal, F is the Fourier transform, and U is the step function. The imaginary component y is simultaneously the quadrature of the real-valued trace. This provides locality to explicit phase information, where the Fourier transform itself does not lend itself to the resolution of the phase in the time domain. In conventional seismic trace analysis, the complex data is used

to calculate the instantaneous amplitude and instantaneous frequency. These are beneficial seismic attributes for interpretation (Barnes, 2007).

5. Experiments

5.1. Data

The data is the F3 seismic data, interpreted by Alaudah et al. (2019). They provide a seismic benchmark for machine learning with accessible NumPy format. The interpretation (labels) of the seismic data is relatively coarse compared to conventional seismic interpretation, but the accessibility and pre-defined test case are compelling.

The F3 data set was acquired in the Dutch North Sea in 1987 over an area of 375.31 km^2. The sampling-rates are 4 ms in time and inline/crossline bins of 25 m. The extent being 650 inline traces and 950 crossline traces with a total length of 1.848 s. The data contains faulted reflector packets, of which the lowest one overlays a salt diapir. The data contains some noise that masks lower-amplitude events.

We generate 64x64 2D patches in the inline and crossline direction from the 3D volume to train our network. The fully convolutional architecture can predict on arbitrary sizes after training. The seismic data is normalized to values in the range of [-1, 1]. To obtain complex-valued seismic data we apply a Hilbert transform to every trace of the data and subtract the real-valued seismic from the real component (Taner et al., 1979).

5.2. Architecture

The Auto-encoder architecture uses 2D convolutions with 3x3 kernels. We employ batch normalization to regularize the training and speed up training (Ioffe and Szegedy, 2015). The down and up sampling is achieved by MaxPooling and the UpSampling operation, respectively. We reduce a 64x64 input 4 times by a factor of two to encode a 4x4 encoding layer. The architecture for the complex convolutional network is identical, except for replacing the real-valued 2D convolutions with complex-valued convolutions. The layers used are shown below (see Table 1).

Complex-valued neural networks contain two feature maps for every feature map contained in a real-valued network. Matching real-valued and complex-valued neural networks is quite complicated, as the same filter values yield a vastly different amount of parameters, as can be seen in Table 1. The smaller real-valued network contains as many feature maps for the real-valued seismic as the large complex network, the large complex network contains an additional feature map for every real-valued input for the complex component. We define a complex-valued network that effectively has the same number of filters as the real-valued small network. This network effectively has half the available feature maps for the real-valued seismic input. Moreover, we define a large real-valued network to match the number of filters of the large complex-valued network, this network has twice the feature-maps available for representation of the real-valued seismic data, compared to the large complex-valued network. The parameters are counted on the computational graph compiled by Tensorflow.

Layer (Size)	Spatial X	Spatial Y	$\mathbb{C}$omplex Small	$\mathbb{R}$eal Small	$\mathbb{C}$omplex Large	$\mathbb{R}$eal Large
Input	64	64	2	1	2	1
($\mathbb{C}$)-Conv2D	64	64	8	8	16	16
($\mathbb{C}$)-Conv2D + BN	64	64	8	8	16	16
Pool + ($\mathbb{C}$)-Conv2D + BN	32	32	16	16	32	32
Pool + ($\mathbb{C}$)-Conv2D + BN	16	16	32	32	64	64
Pool + ($\mathbb{C}$)-Conv2D + BN	8	8	64	64	128	128
Pool + ($\mathbb{C}$)-Conv2D	4	4	128	128	256	256
Up + ($\mathbb{C}$)-Conv2D + BN	8	8	64	64	128	128
Up + ($\mathbb{C}$)-Conv2D + BN	16	16	32	32	64	64
Up + ($\mathbb{C}$)-Conv2D + BN	32	32	16	16	32	32
Up + ($\mathbb{C}$)-Conv2D	64	64	8	8	16	16
($\mathbb{C}$)-Conv2D + BN	64	64	8	8	16	16
($\mathbb{C}$)-Conv2D	64	64	2	1	2	1
Parameters on Graph			100,226	198,001	397,442	790,945
Size on Disk [MB]			1.4	2.5	4.8	9.2

Table 1: Layers used in the four auto-encoders and according parameter count on the computational graph and size on disc. Complex-valued convolutions and real-valued convolutions used respectively.

5.3. Training

We train the networks with an Adam optimizer (Kingma and Ba, 2014) and a learning rate of 10^{-3} without decay, for 100 epochs. The loss function is mean squared error, as the seismic data contains values in the range of [-1,1]. All networks reach stable convergence without overfitting, shown in Figure 5.

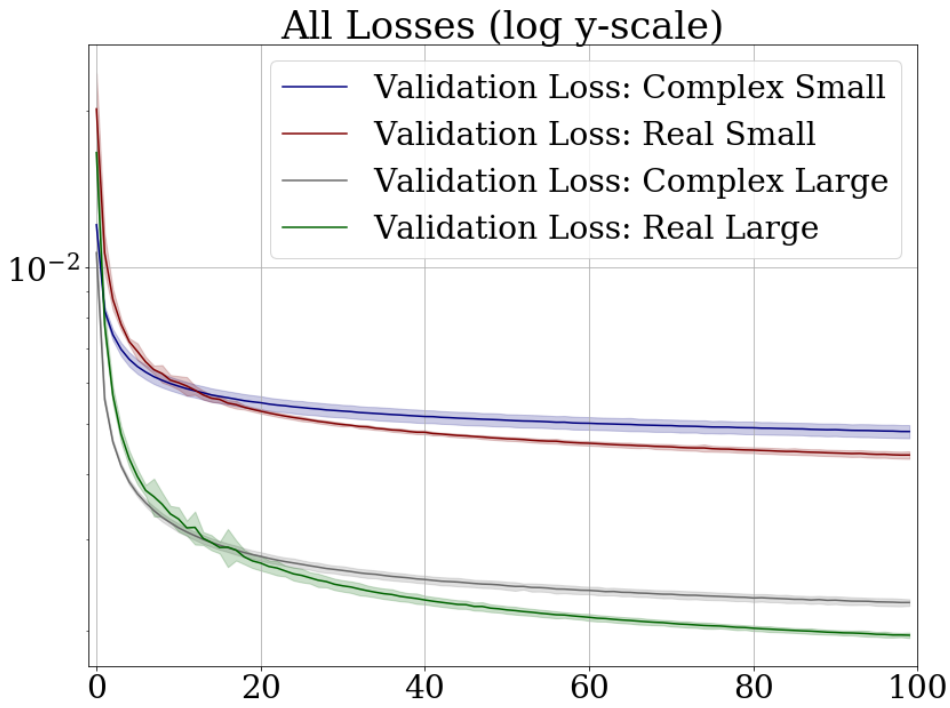


Figure 5: Validation Loss (MSE) on 7 random seeds per network. (Real-valued loss on real-valued seismic and combined complex-valued loss on complex-valued seismic, as the network "sees" it.)

5.4. Evaluation

We compare the complex auto-encoders with the real-valued auto-encoders, through the reconstruction error on unseen test data on 7 individual realizations of the respective four networks and qualitative analysis of reconstructed images. We focus on evaluating the real-valued reconstruction of the seismic data.

6. Results

We trained four neural network auto-encoders with seven random initializations for each network, to allow for error bars on the estimates in Figure 5. The mean squared error and the mean absolute error for each parameter configuration during training is given in Table 2. There is a clear correspondence of the reconstruction error of the auto-encoder to the size of network. The real-valued networks outperform the complex-valued networks in both the mean squared error and mean absolute error, however, we see that a real-valued network needs around twice as many parameters as a complex-valued network to attain the same reconstruction errors.

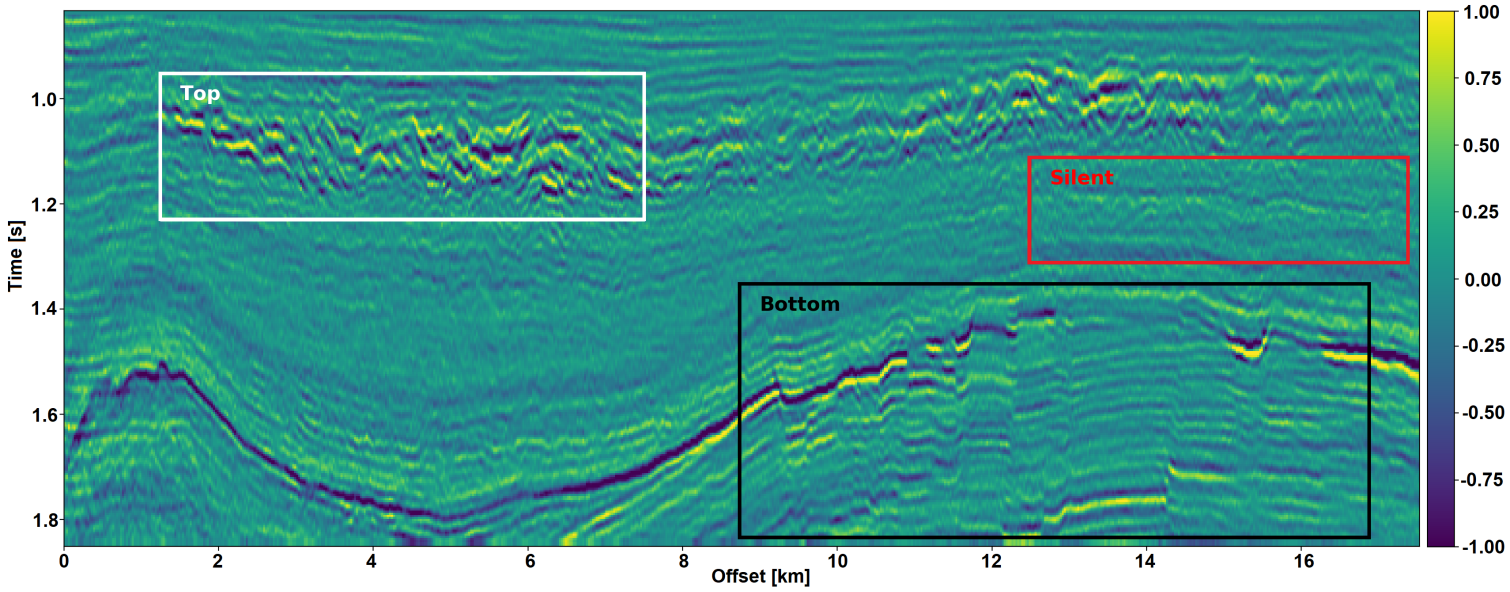
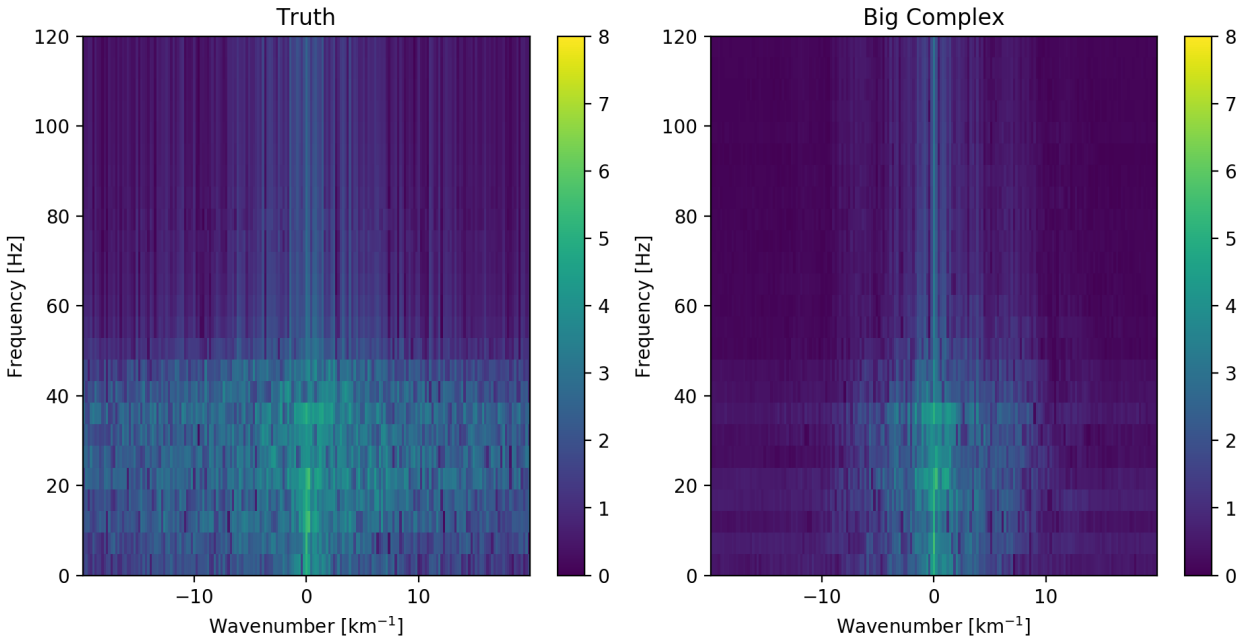


Figure 6: Seismic Test Data with marked section for closer inspection. We chose the "top" section for it's faulted chaotic texture, "bottom" for the faulted blocks, and "silent" for a noisy but geologically uninteresting section.

Network	Runs × epochs	Parameters	MSE [$\times 10^{-2}$]	MAE [$\times 10^{-2}$]
1) $\mathbb{C}_{\text{small}}$	7×100	100,226	0.484 ± 0.013	4.695 ± 0.058
2) $\mathbb{R}_{\text{small}}$	7×100	198,001	0.436 ± 0.006	4.500 ± 0.028
3) $\mathbb{C}_{\text{large}}$	7×100	397,442	0.227 ± 0.003	3.247 ± 0.025
4) $\mathbb{R}_{\text{large}}$	7×100	790,945	0.196 ± 0.002	3.050 ± 0.013

Table 2: Parameters and errors for networks (lower is better). Losses on network validation.

The seismic sections in Figure 6 show the unseen test seismic section. We perform a closer inspection of the regions "top" and "bottom" to focus on geologically relevant sections in the reconstruction process. The noisy segment without strong reflectors is a good baseline to evaluate the noise reduction of the Autoencoder and the behaviour of the different networks on low amplitude data. Overall, all networks denoise the original seismic, with the lowest reconstruction errors being RMS of 0.1187 and MAE of 0.0947 (cf. Table 3). Figure 7 shows the frequency-wavenumber (FK) of the ground truth (7 (a)) and the large complex network reconstruction (7 (b)). These show a decrease in the 0 - 60 Hz band for larger absolute wavenumbers.



(a) Ground Truth

(b) Large Complex Network

Figure 7: Evaluation on Silent Noise Patch in FK Domain. Noise reduction of frequencies below 50 Hz apparent, while reconstruction does not introduce visible aliasing.

	Full		Silent		Top		Bottom	
Network	RMS	MAE	RMS	MAE	RMS	MAE	RMS	MAE
1) $\mathbb{C}_{\text{small}}$	0.1549	0.1145	0.1265	0.1010	0.2315	0.1759	0.1588	0.1200
2) $\mathbb{R}_{\text{small}}$	0.1581	0.1153	0.1247	0.0994	0.2395	0.1810	0.1612	0.1205
3) $\mathbb{C}_{\text{large}}$	0.1508	0.1101	0.1187	0.0947	0.2301	0.1747	0.1514	0.1135
4) $\mathbb{R}_{\text{large}}$	0.1469	0.1072	0.1214	0.0967	0.2222	0.1679	0.1459	0.1088

Table 3: RMS and MAE on real component of Data Patches.

6.1. "Top" seismic section

The "top" segment contains strong reflections that are very faulted with strong reflectors. Figure 8 shows the top segment and the reconstructions of the four networks. All networks display various amounts of smoothing. The quantitative results show that the complex networks perform very similar regardless of size. The large real-valued network outperforms the complex networks by 2.5 % on RMS, while the small real-valued network underperforms by 2.5 % on RMS. The panel in Figure 8c shows a very smooth result. Despite the close score of the complex networks, it appears that the complex-valued network restores more high-frequency content. We can also see less smearing of discontinuities in the larger complex network, particularly visible in the lower part (1.2 s) at 6000 m offset, which is smeared to appear like a diffraction

in the smaller network. The large real-valued network shows good reconstruction with minor smearing with higher amplitude fidelity in areas like 1.1 s at 2000 m, however, some of the steeply dipping artifacts are visible below the reflector packet between 0 m and 2000 m offset.

6.2. "Bottom" seismic section

The data marke as "bottom" in Figure 6 contains a faulted anticline and relatively strong noise levels. The small complex network in Figure 9b reconstructs a denoised image with good reconstruction of the visible discontinuities. Some leakage of the reflector starting at 1.5 s across discontinuities is visible. The real small network in Figure 9c reconstructs a strongly smoothed image, with some ringing below the main reflector, which is not visible in the other reconstructions. The dipping reflector at an offset of 16000 m is well reconstructed, however, it seems like the reconstruction introduced ringing noise over the vertical image. The large real-valued network in Figure 9e performs best quantitatively (cf. Table 3). The complex-valued large network in Figure 9d does a fairly good job at reconstructing the image, similar to the large real-valued network. However, the amplitude reconstruction of high-amplitude events particularly in the main reflector around 1.5 s is showing.

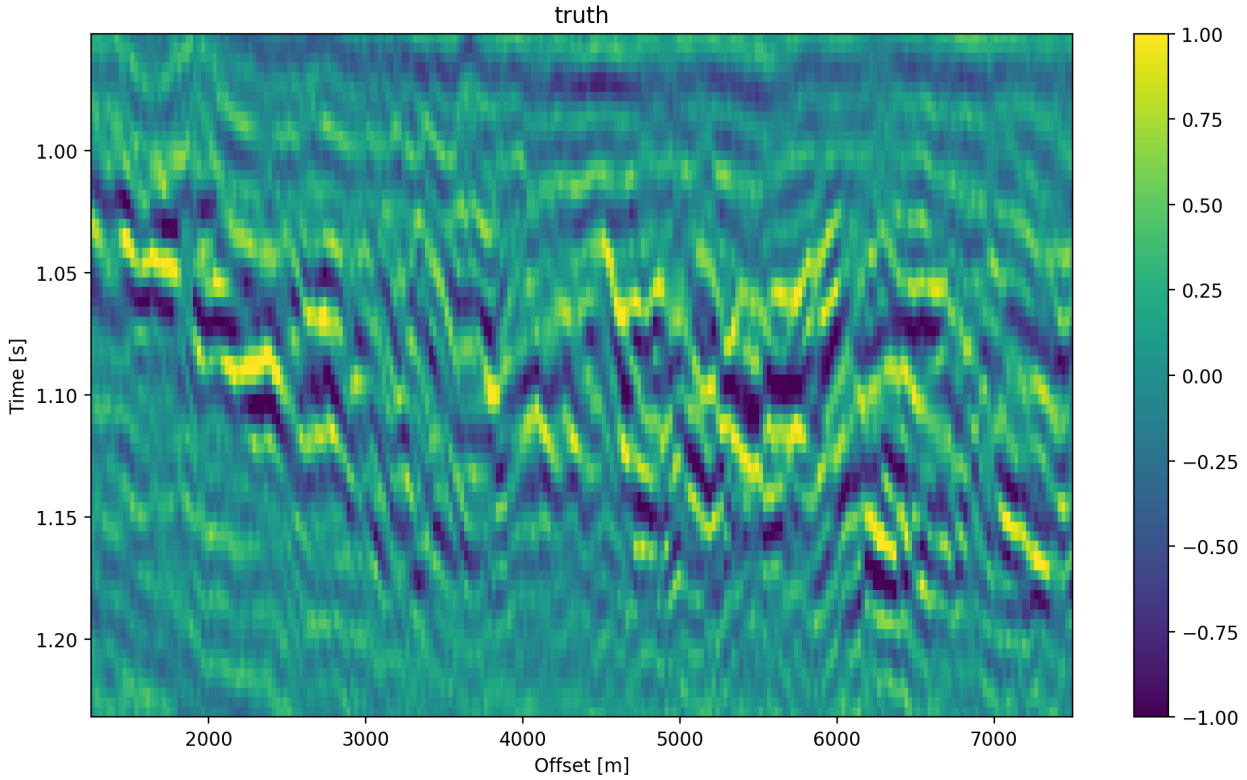
6.3. Full seismic test data

It is evident, that the small real-valued network does not match the performance of the smaller complex-valued network, even less so when compared to the large complex-valued network. We therefore compare the large networks on the full seismic data.

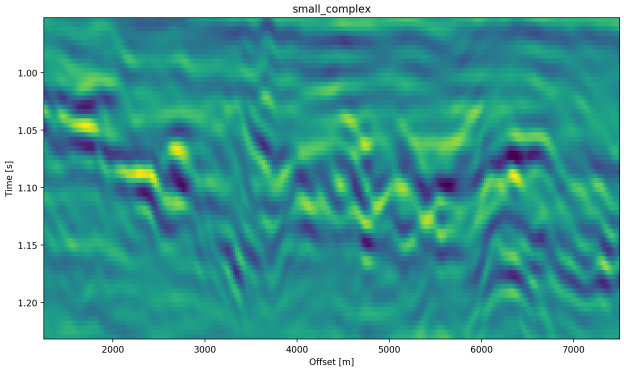
Overall, both networks return a smoothed image. The findings for the strongly faulted sections in the "top" panel hold across the entire faulted area around 1.1 s in Figure 11. The complex-valued network does a better job at reconstructing faults and discontinuities. The real-valued network is better at reconstructing high-amplitude regions that appear dimmer in the complex-valued region. The reconstruction of both networks seems adequately close to the ground truth, with differences in the details. Quantitatively, the real-valued network does the better reconstruction in Table 3 with an improvement of 2.5 % over the large complex-valued network. The FK domain shows a very similar reduction in noise in the sub 50 Hz band in Figure 10. All networks introduce an increase of energy across all frequencies at wave-number $k = 0\ km^{-1}$. Additionally, a dimming of the frequencies around $k = 2.5\ km^{-1}$ appears in all reconstructions, but is more prominent in the large complex-valued network. The ground truth seismic contains some scattered energy in the high-frequency mid-wavenumber region, visible as "diagonal stripes". These were attenuated in the complex-valued network in Figure 10b, but are partially present in the real-valued reconstruction in Figure 10c.

7. Discussion

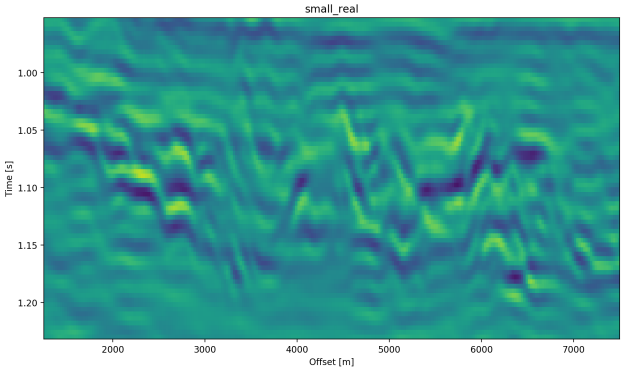
We evaluated the outputs of the real-valued and complex-valued neural networks. All auto-encoder outputs are blurred to different degrees and denoised. The denoising effect of the seismic was most visible in



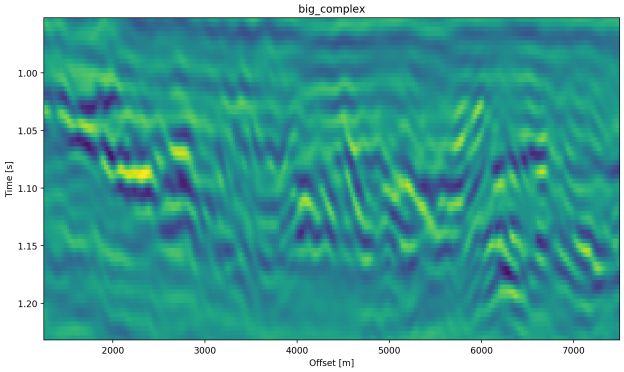
(a) Ground Truth



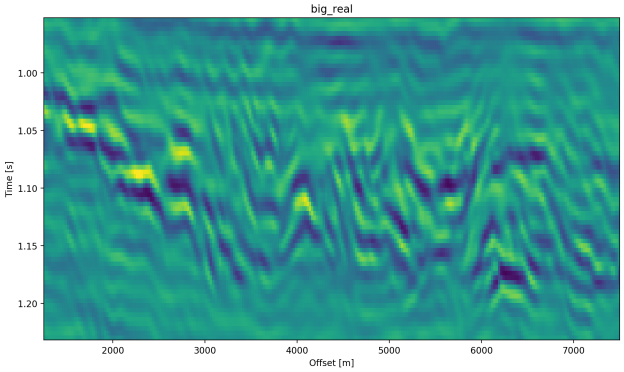
(b) Small Complex Network top Patch



(c) Small Real Network top Patch

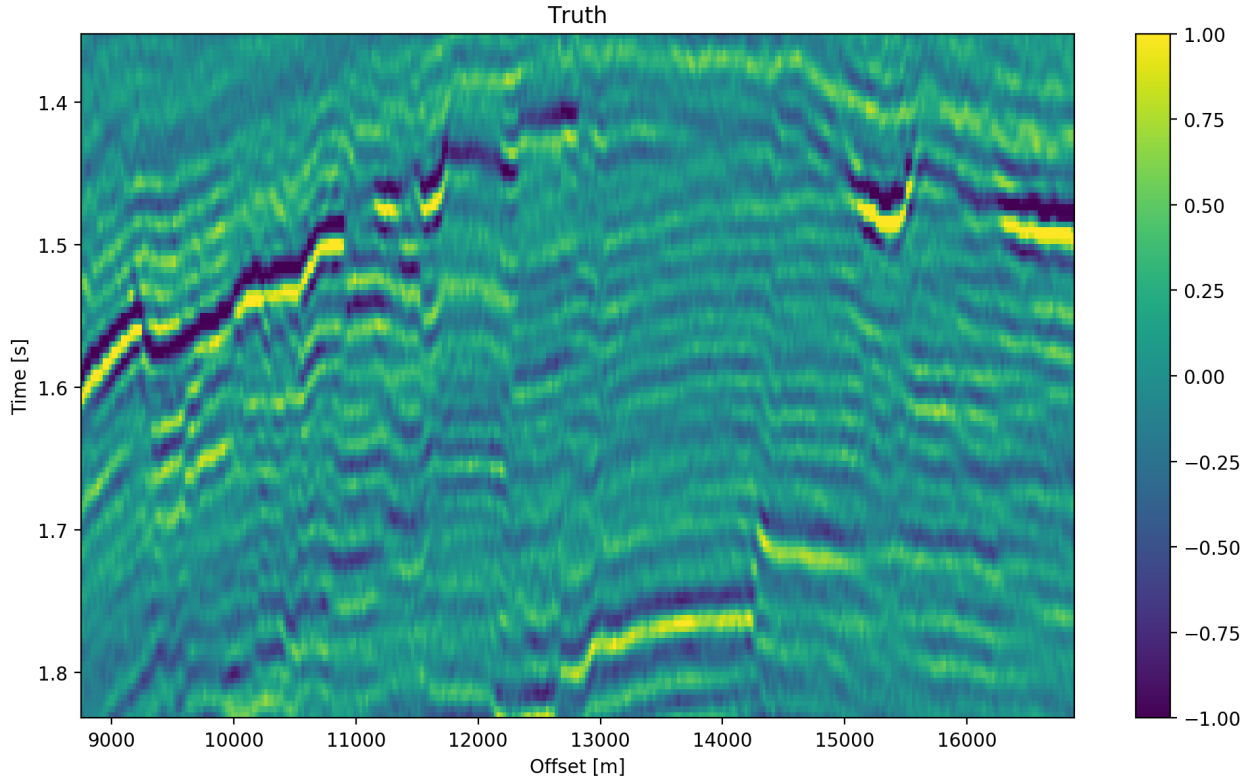


(d) Large Complex Network top Patch

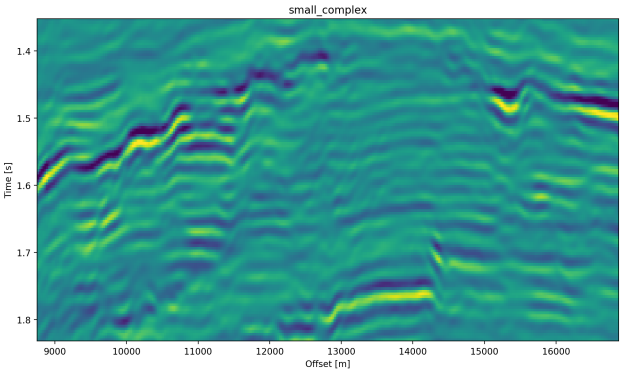


(e) Large Real Network top Patch

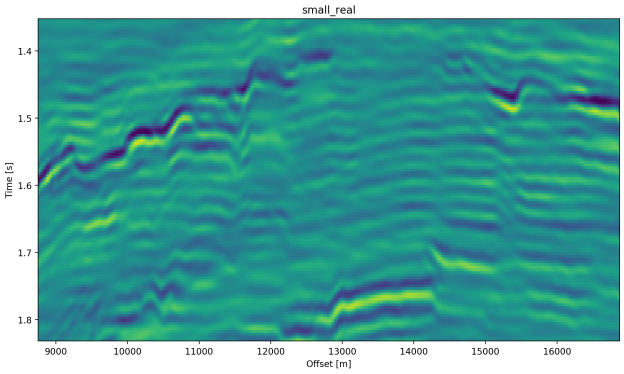
Figure 8: Evaluation on top Noise Patch



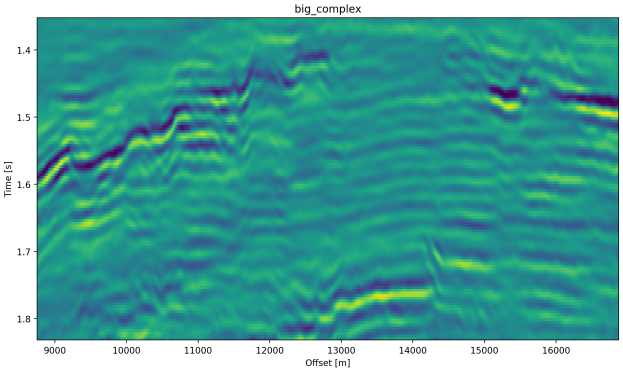
(a) Ground Truth



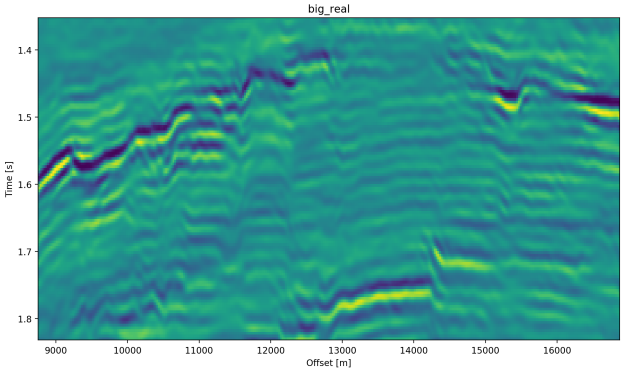
(b) Small Complex Network bottom Patch



(c) Small Real Network bottom Patch

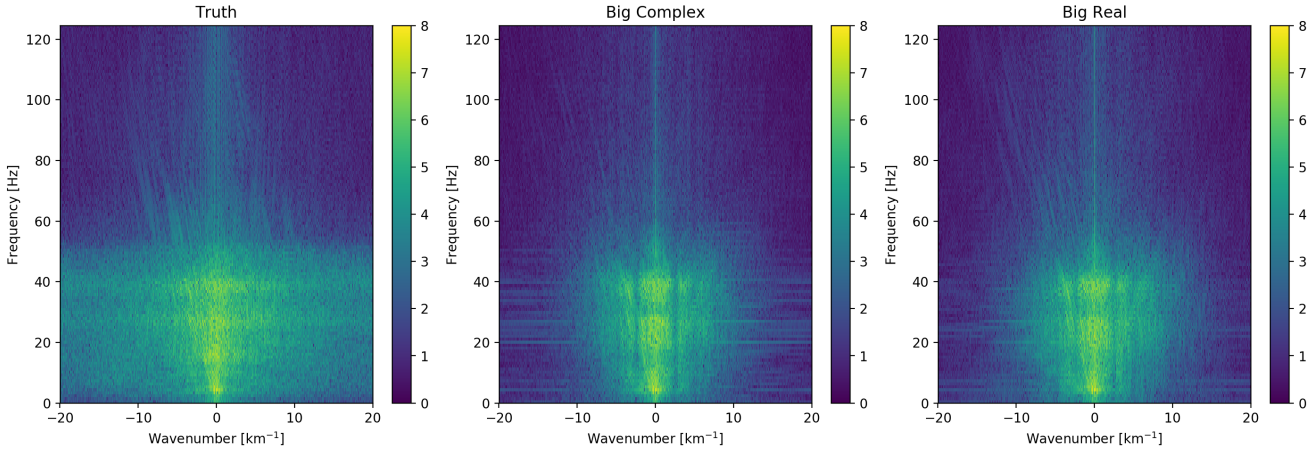


(d) Large Complex Network bottom Patch



(e) Large Real Network bottom Patch

Figure 9: Evaluation on bottom Noise Patch



(a) Ground Truth (b) Large Complex Network (c) Large Real Network

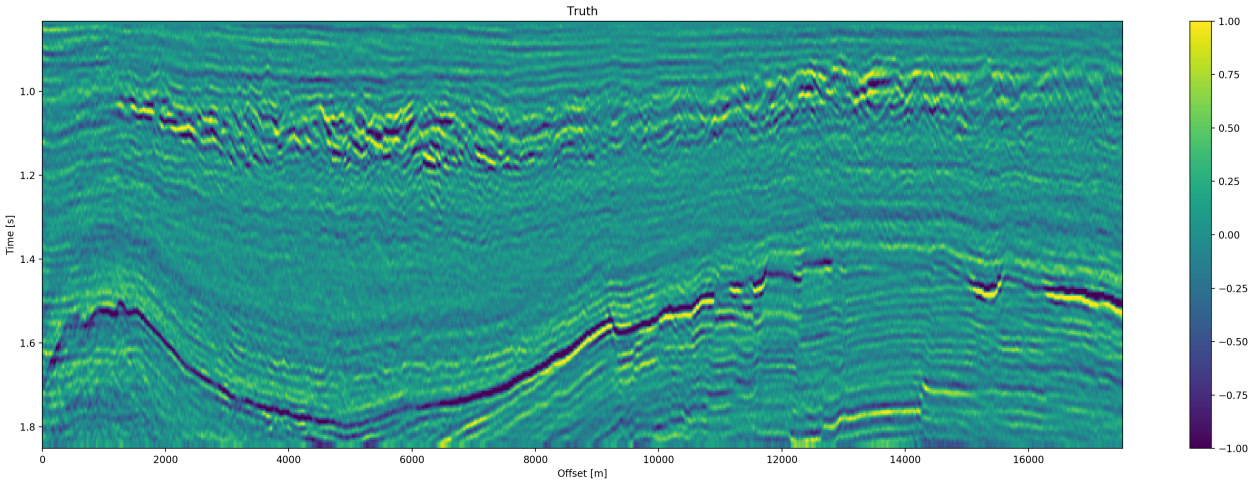
Figure 10: FK domain of full seismic data.

the frequency band below 50 Hz. Additionally, some scattered high-frequency energy was attenuated by the networks.

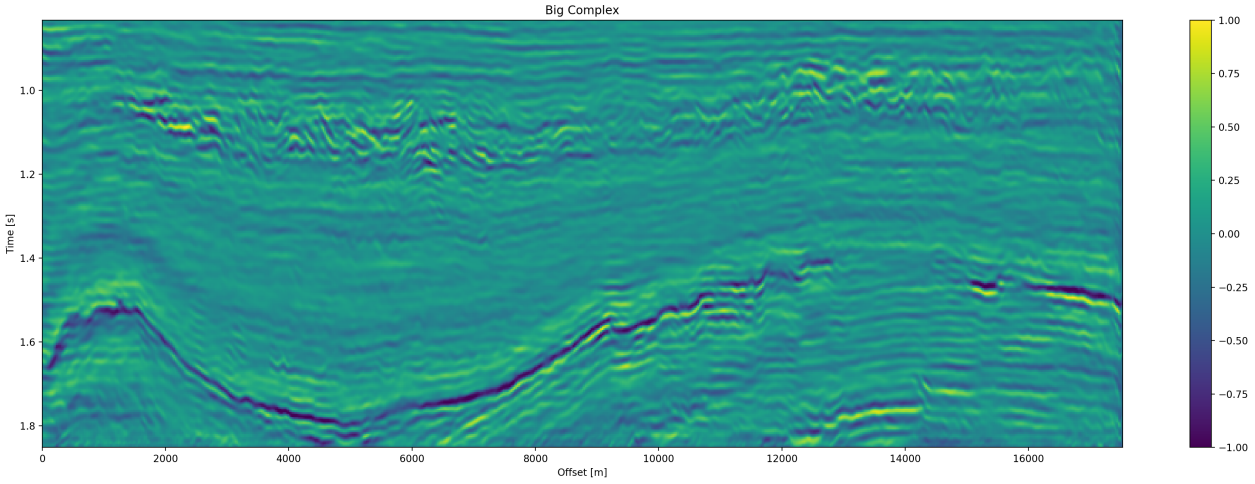
The largest differences of the outputs in real-valued and complex-valued networks can be observed in discontinuous areas. Particularly, the faulted blocks in the top quarter and in the bottom center of the seismic section show inconsistencies. The real-valued network smooths over discontinuities and steep reflectors. Fault lines are imaged better in the complex-valued network output.

In seismic data processing, including phase information stabilizes discontinuities and disambiguates cycle-skipping in horizons. This could be observed in the network performance and reconstruction. The increase in performance of the real-valued networks was significant (7.0 % RMS), while the complex-valued networks already had an acceptable performance on the smaller network architecture (2.6 % RMS). We provide the complex-valued networks with a bias towards learning phase information, by providing the Hilbert transformed analytical trace, while the real-valued network needs to learn this information implicitly from the data itself. Considering, that during the training, the complex network evaluates both the real-valued seismic, which we primarily care about in addition to the complex-valued component, we can see how the losses in Figure 5 differ from the real-valued networks.

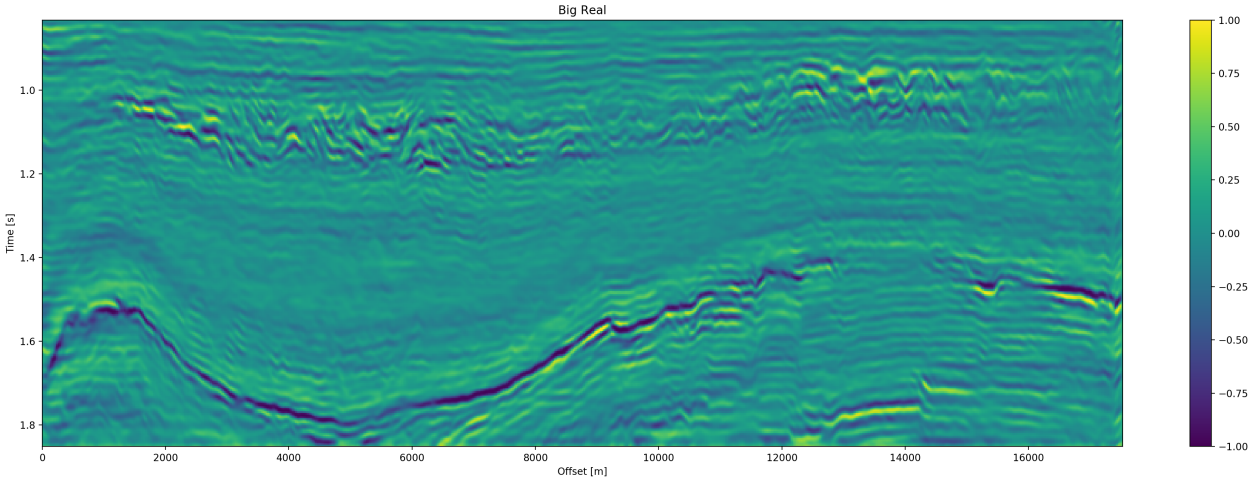
The largest network with 790,945 trainable parameters quantitatively performed the best on the reconstruction of the data. However, analysis of the reconstructed seismic shows, that while the high-amplitude regions are reconstructed to higher fidelity, discontinuous sections may be smeared by the real-valued network. The real-valued network that was matched to contain as many filters for the real-valued component of the seismic as the large complex-valued network, did not perform well. Furthermore, the smaller complex-valued network with 100,226 parameters that contains as many filter maps as the real-valued network in total, and half the trainable parameters, outperformed the smaller real-valued network across all test cases.



(a) Ground Truth



(b) Large Complex Network



(c) Large Real Network

Figure 11: Evaluation on full seismic data.

8. Conclusion

The inclusion of phase-information leads to a better representation of seismic data in convolutional neural networks. Complex-valued networks perform consistently, where real-valued networks have to learn phase-representations through implicit correlation, which requires larger networks. We show that complex trace information in deep neural networks improves the imaging of discontinuities as well as steep reflectors, particularly in chaotic seismic textures that are smoothed by real-valued neural networks of the same size.

We show that convolutional neural networks can perform lossy compression on seismic data, where the reconstruction error is dependent on both network architecture and implementation details, like providing explicit phase information. During this compression, noise and scattered energy get attenuated. The real-valued network is prone to introduce steeply dipping artifacts in the reconstruction.

The stabilization of the reconstruction can be useful in seismic applications. While automatic seismic interpretation may benefit from the inclusion of information on discontinuities, we see the main application to be lossy seismic compression. The open source tool developed to make this research possible, enables further research and development of complex-valued solutions to non-stationary physics problems that benefit from explicit phase information.

The research shows that a change as small as 2.5 % in RMS can change the reconstruction from being acceptable to very smeared to a geoscientist. This touches on the fact that better metrics to evaluate computer vision tasks in geoscience are necessary. Additionally, these tasks have to be noise-robust and while amplitude-preserving be outlier robust too. Moreover, more research in the frequency dimming of bands in the network reconstruction is necessary.

Overall, the computational memory footprint of the complex convolution is higher than real-valued convolutional neural networks comparing singular convolutional operations. A significant increase in depth and width of networks to obtain an acceptable result in real-valued neural network to implicitly learn the phase information is necessary. The complex-valued networks an 8^{th} of the size already performs well, suggesting that expert domains that contain beneficial information in the phase of signals, could benefit from applying complex convolutional networks.

9. Acknowledgments

We thank Andrew Ferlitsch for his valuable insights. The research leading to these results has received funding from the Danish Hydrocarbon Research and Technology Centre under the Advanced Water Flooding program. We thank DTU Compute for access to the GPU Cluster.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser,

L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: http://tensorflow.org/. software available from tensorflow.org.

Alaudah, Y., Michalowicz, P., Alfarraj, M., AlRegib, G., 2019. A machine learning benchmark for facies classification. arXiv preprint arXiv:1901.07659 .

Barnes, A.E., 2007. A tutorial on complex seismic trace analysis. Geophysics 72, W33–W43.

Chollet, F., et al., 2015. Keras. https://keras.io.

Dramsch, J.S., Lüthje, M., 2018a. Deep-learning seismic facies on state-of-the-art cnn architectures, in: SEG Technical Program Expanded Abstracts 2018. Society of Exploration Geophysicists, pp. 2036–2040.

Dramsch, J.S., Lüthje, M., 2018b. Information theory considerations in patch-based training of deep neural networks on seismic time-series, in: First EAGE/PESGB Workshop Machine Learning, EAGE Publications BV. URL: https://doi.org/10.3997/2214-4609.201803020, doi:10.3997/2214-4609.201803020.

Griffin, D., Lim, J., 1984. Institute of Electrical and Electronics Engineers. pp. 236–243. URL: https://doi.org/10.1109/icassp.1983.1172092, doi:10.1109/icassp.1983.1172092.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science 313, 504–507.

Hirose, A., Yoshida, S., 2012. Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. IEEE Transactions on Neural Networks and Learning Systems 23, 541–551. URL: https://doi.org/10.1109/tnnls.2012.2183613, doi:10.1109/tnnls.2012.2183613.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 .

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.

LeCun, Y., Haffner, P., Bottou, L., Bengio, Y., 1999. Object recognition with gradient-based learning, in: Shape, contour and grouping in computer vision. Springer, pp. 319–345.

Liner, C.L., 2002. Phase, phase, phase. The Leading Edge 21, 456–457. URL: https://doi.org/10.1190/1.1885500, doi:10.1190/1.1885500.

Mavko, G., Mukerji, T., Dvorkin, J., 2003. The rock physics handbook.

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A.C., Bengio, Y., 2016. Samplernn: An unconditional end-to-end neural audio generation model. CoRR abs/1612.07837. URL: http://arxiv.org/abs/1612.07837, arXiv:1612.07837.

Miyauchi, M., Seki, M., Watanabe, A., Miyauchi, A., 1993. Interpretation of optical flow through complex neural network, in: New Trends in Neural Computation. Springer Berlin Heidelberg, pp. 645–650. URL: https://doi.org/10.1007/3-540-56798-4_215, doi:10.1007/3-540-56798-4_215.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. CoRR abs/1609.03499. URL: http://arxiv.org/abs/1609.03499, arXiv:1609.03499.

van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L.C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., Hassabis, D., 2017. Parallel wavenet: Fast high-fidelity speech synthesis. CoRR abs/1711.10433. URL: http://arxiv.org/abs/1711.10433, arXiv:1711.10433.

Paganini, M., de Oliveira, L., Nachman, B., 2017. CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks arXiv:1705.02355.

Prenger, R., Valle, R., Catanzaro, B., 2018. Waveglow: A flow-based generative network for speech synthesis. CoRR abs/1811.00002. URL: http://arxiv.org/abs/1811.00002, arXiv:1811.00002.

Purves, S., 2014. Phase and the hilbert transform. The Leading Edge 33, 1164–1166. URL: https://doi.org/10.1190/tle33101164.1, doi:10.1190/tle33101164.1.

Roden, R., Sepúlveda, H., 1999. The significance of phase to the interpreter: Practical guidelines for phase analysis. The Leading Edge 18, 774–777. URL: https://doi.org/10.1190/1.1438375, doi:10.1190/1.1438375.

Sarroff, A.M., 2018. Complex Neural Networks for Audio. Technical Report TR2018-859. Dartmouth College, Computer Science. Hanover, NH. URL: http://www.cs.dartmouth.edu/~trdata/reports/TR2018-859.pdf.

Sarroff, A.M., Shepardson, V., Casey, M.A., 2015. Learning representations using complex-valued nets. CoRR abs/1511.06351. URL: http://arxiv.org/abs/1511.06351, arXiv:1511.06351.

Scarpiniti, M., Vigliano, D., Parisi, R., Uncini, A., 2008. Generalized splitting functions for blind separation of complex signals. Neurocomputing 71, 2245–2270. URL: https://doi.org/10.1016/j.neucom.2007.07.037, doi:10.1016/j.neucom.2007.07.037.

Suksmono, A.B., Hirose, A., 2002. Adaptive noise reduction of InSAR images based on a complex-valued MRF model and its application t o phase unwrapping problem. IEEE Transactions on Geoscience and Remote Sensing 40, 699–709. URL: https://doi.org/10.1109/tgrs.2002.1000329, doi:10.1109/tgrs.2002.1000329.

Taner, M.T., Koehler, F., Sheriff, R., 1979. Complex seismic trace analysis. Geophysics 44, 1041–1063.

Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J.F., Mehri, S., Rostamzadeh, N., Bengio, Y., Pal, C.J., 2017. Deep complex networks. arXiv preprint arXiv:1705.09792 .

Yilmaz, Ö., 2001. Seismic data analysis. volume 1. Society of exploration geophysicists Tulsa, OK.

Yan *et al.*
Chinese Journal of Mechanical Engineering (2025) 38:5
https://doi.org/10.1186/s10033-024-01173-8

Chinese Journal of Mechanical Engineering

REVIEW

Open Access

Knowledge Driven Machine Learning Towards Interpretable Intelligent Prognostics and Health Management: Review and Case Study

Ruqiang Yan[1*], Zheng Zhou[1], Zuogang Shang[1], Zhiying Wang[1], Chenye Hu[1], Yasong Li[1], Yuangui Yang[1], Xuefeng Chen[1] and Robert X. Gao[2]

Abstract

Despite significant progress in the Prognostics and Health Management (PHM) domain using pattern learning systems from data, machine learning (ML) still faces challenges related to limited generalization and weak interpretability. A promising approach to overcoming these challenges is to embed domain knowledge into the ML pipeline, enhancing the model with additional pattern information. In this paper, we review the latest developments in PHM, encapsulated under the concept of Knowledge Driven Machine Learning (KDML). We propose a hierarchical framework to define KDML in PHM, which includes scientific paradigms, knowledge sources, knowledge representations, and knowledge embedding methods. Using this framework, we examine current research to demonstrate how various forms of knowledge can be integrated into the ML pipeline and provide roadmap to specific usage. Furthermore, we present several case studies that illustrate specific implementations of KDML in the PHM domain, including inductive experience, physical model, and signal processing. We analyze the improvements in generalization capability and interpretability that KDML can achieve. Finally, we discuss the challenges, potential applications, and usage recommendations of KDML in PHM, with a particular focus on the critical need for interpretability to ensure trustworthy deployment of artificial intelligence in PHM.

Keywords PHM, Knowledge driven machine learning, Signal processing, Physics informed, Interpretability

1 Introduction

Prognostics and health management (PHM) is an engineering discipline to extend life cycle of physical systems in service, including anomaly detection to identify binary health state, fault diagnosis to isolate the fault location, fault prognosis to predict remaining useful life, and condition-based maintenance to optimize maintenance schedule. A general workflow is composed of sensor data acquisition, feature extraction, and decision making, where feature extraction is a cornerstone to convert high dimensional sensor reading to low dimensional states and the changes in these states can be detected using pattern recognition approaches. To understand the degradation process of physical systems, there are mainly two distinguishable kinds of approaches to extract feature for PHM, i.e., physics-based and data-driven. For the former, degradation process is represented by concrete physical variables, like crack length [1] or stiffness [2]. The development of physics-based methods has led to various advances in PHM knowledge, like degradation patterns

*Correspondence:
Ruqiang Yan
yanruqiang@xjtu.edu.cn
[1] State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[2] Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH 44106, USA

in life-cycle period [3] or performance model between damage variables with responsible variables [4]. For the latter, measurement data is expected to contain information about the degradation process of the physical systems, like changes in frequency components [5]. Signal processing [5] and machine learning are two mainly approaches to extract such pattern from data. In the last decades, signal processing has developed a variety of transformation methods to analyze data for PHM, like Fourier transformation or wavelet transformation. These transformation methods map data from original representation space to another, making patterns discriminative. Compared with implicit mathematical modeling evolution of system state variables through physics-based approaches or signal processing methods, machine learning (ML) provides a powerful learning system for feature extraction in PHM [6]. This powerful learning system shows strong flexibility and generalization ability in PHM with support of various types of models, like classical multilayer perception (MLP) or currently popular Transformer. These success scenarios are grounded in the different inductive bias of different ML models. For example, (1) MLP is a universal approximation function and can approximate any desired functions or operators in pattern recognition. (2) Convolution neural network has symmetry property of translation, and this inductive bias can improve its generalization ability with respect to period feature, like impulses in vibration signal. (3) Recurrent neural network is relative to wave equation or differential equation and it can capture temporal relation from time series data. (4) Graph neural network is a natural choice to model unstructured relation within multiple sensors or multi-modal data. Essentially, these general inductive biases are merely the intrinsic characters of the models. If there is no further specification, they still require sufficient data samples to train an efficient learning system, while data scarcity and imbalance are persistent problems in PHM. Due to the conservative maintenance strategy, variable conditions, and unknown fault types in PHM domain, sensor data is usually represented in a long tail distribution. Such data with insufficient variation will result in overfitting for ML models. In addition, PHM domains have another requirement for ML models to be interpretable and trustworthy. As PHM domain is risk sensitive, a wrong prediction can lead to heavy loss of life and property. From this perspective, an unexplained model does not allow users to participate in the machine's decision loop, nor does it allow users to make trustworthy decision.

To solve the abovementioned difficulties, current researches attempt to embed specific domain knowledges into ML models to improve generalization ability and interpretability, ranging from expert knowledges, physics knowledges to signal processing knowledges. Expert knowledges, like causal graphs [7] or probability equations [8], have been integrated into ML models to achieve some desired properties such as variable dependence and smoothness in states prediction for PHM. Physics knowledges, like physical equations [9] or simulation [10], also show great success to enhance physical interpretability of neural representation extracting from sensor readings. As long-term development research in PHM, signal processing knowledges provide rich data analysis techniques to specialize the ML pipeline for PHM tasks [5]. Knowledge has broad definition with understanding of empirical facts. In science or engineering, knowledge generally comes from the observation of experimental phenomenon, theory modeling of the first principle, and information system of computational science. As knowledge is expected to have invariant applicability for entire domain, generalization can be seamlessly expected to be improved. More importantly, knowledge integration can constrain the function of ML models into a specific space, and then the prediction logic can conform to the specific knowledge constraint and be more interpretable. Based on the classification of interpretable machine learning approaches (i.e., post-hoc and ad-hoc), most of knowledge-driven methods should fall into the ad-hoc category, as these methods typically involve active intervention in the modeling process, like feature transformation or optimization regularization. This is why more and more attention has been paid in this topic, including physics-informed machine learning [11], theory-guided machine learning [12], or science-guided machine learning [13]. Especially in PHM domain, various knowledge, like inductive experience, physics model, signal processing, have been developed for over decades. Recent research tendency shows the scalability of ML models to be combined with various knowledges in PHM. Through the integration of domain knowledges with data-driven learning systems, such hybrid approaches promote ML pipeline to capture more interpretable patterns from sensor readings and show potential to trustworthy artificial intelligence in engineering applications. Therefore, a systematic review is needed to investigate the common ground and diversity in literatures and then to capture its main research direction.

To summarize existing research in knowledge driven machine learning (KDML) in PHM and identify trends and gaps, we provide a hierarchical structure to organize the advancement of KDML research. Rather than the current literature search methodology to group KDML mainly based on embedding approaches (including three categories: data-centric, model structure-centric, and optimization centric) [14–16], our hierarchical structure refers to a recent survey on informed machine learning

[17] and will provide a roadmap from the knowledge sources, knowledge representations and the knowledge embedding approaches. Especially for PHM task, this roadmap will emphasize the currently mainstream trends of KDML in terms of knowledge source and representation, where signal processing and physical model are the two themes that dominate. Compared to general KDML in other engineering fields [11, 18], KDML in PHM will concern more about the specific challenges in lifecycle management and dynamic operational conditions in predictive maintenance to pursue a reliable and trustworthy solution. In addition, there also exists some reviews of interpretable machine learning for PHM [19, 20], which also paid attention to knowledge-driven machine learning approaches, such as physics-informed neural networks and signal processing-informed neural networks. The difference between our review with existing literatures is that we will establish a hierarchical framework of KDML to provide a practical roadmap, including knowledge sources, knowledge representations, and knowledge embedding approaches.

Our goal is to investigate the main research directions and approaches of knowledge driven machine learning in PHM and then provide basis references for potential users in PHM domain towards interpretable and trustworthy ML applications. We start from the knowledge in PHM by where we can conclude knowledge and how to represent it. The difference between classical ML approaches and knowledge driven machine learning is illustrated to distinguish how to integrate knowledge into ML pipeline. Through hierarchical review in knowledge sources and ML pipeline, we introduce the main research approaches to implement KDML in PHM. Our contributions are listed as follows:

(1) We utilize an inclusive concept to ordinate different approaches like physics-informed or theory-guided to integrate knowledge and data-driven models in PHM domain, named as knowledge driven machine learning. This clarification states the common ground of recent researches which integrate knowledge independent of data acquisition into ML pipeline to improve generalization ability and interpretability.
(2) We provide a hierarchical overview on KDML implementation in PHM by knowledge sources and ML pipeline. This literature analysis methodology emphasizes the classification of different knowledge sources, and then potential users in PHM domain can identify where they can be used.
(3) We present extensive case studies to show the results of KDML in PHM for various knowledges, including signal processing, physics model, and inductive experience. Although the knowledge sources of these case studies are diverse, they all follow a similar hierarchical framework of KDML about knowledge representation and knowledge embedding.

The structure of this paper is as follows: In Section 2, following a brief description of knowledge in PHM domain, the concept of knowledge driven machine learning is introduced. In Section 3, the development of KDML in PHM is summarized and classified depending on different knowledge sources. For each knowledge source, we present the knowledge embedding approaches with respect to ML pipeline. In Section 4, we give several case studies of different knowledge sources to describe generalization and interpretability performance of KDML in PHM. We discuss the future development direction and challenges in Section 5, and conclude in Section 6.

2 Concepts of KDML for PHM

In this section, we will provide the details of hierarchical structure roadmap for KDML in PHM, generally including knowledge sources, knowledge representations and knowledge embedding approaches. In Subsection 2.1, three main knowledge sources in PHM are summarized from advancement of scientific paradigms in PHM, including inductive experience, physics model, and signal processing. In Subsection 2.2, concrete forms of knowledge representations in PHM are described. In Subsection 2.3, we describe the main purpose of KDML in PHM and show the knowledge embedding approaches.

2.1 Knowledge Sources in PHM

As knowledge has broad definition in different fields, we restrict the discussion of knowledge in terms of engineering domain, especially PHM. In general, knowledge is a result from empirical observation or scientific experiment. Therefore, knowledge source usually changes with the evolution of scientific paradigm, ranging from empirical science, theoretical science, computational science, and to current data science. In the early stage of PHM research, a general way is to do statistical analysis to build relationship between external physical phenomenon and inherent degradation factor, where knowledge can be represented in a fault tree or fault causal graph. With deepening the understanding of systems, the first principle theory is used to construct theoretical model, such as lumped parameter models or state space model, where knowledge can be represented in an equation or formula. If further employing advanced computing technique, we can utilize system theory, like finite element method (FEM) or computational fluid dynamic (CFD), to

approximate the complex physical system, where knowledge can be represented in such computational system. If big data of the system is available, advanced data processing techniques can be used to recognize fault pattern from data, like using signal processing or ML, where knowledge implicitly exists in such data driven models.

Here, to make knowledge and data more distinguishable in the scientific paradigm of data science, we exclude the knowledge implicit in ML models, while signal processing will still be considered as it is normally considered to be parallel to ML models in PHM field. In addition, we will group theoretical science and computational science into one category of knowledge source, physics model, as the two paradigms are both originated from the first principle. For empirical science, we specialize it as inductive experience in PHM field and split it to two subset of knowledge source, that is, empirical model with explicit formula in terms of experience modeling (like tool wear model) and without explicit formula in terms of inductive knowledge (like fault tree). Figure 1 illustrates the relationship from scientific paradigms to PHM knowledge sources.

2.2 Knowledge Representations in PHM

After defining the three main knowledge sources in PHM, the next thing is to extract specific knowledge representations from different knowledge sources. This category further subdivides the knowledge sources and provides an interface for the following knowledge embedding. Here, we give a brief overview of these knowledge representations in PHM.

Causal graph: Causal graph represents the relation between different variables, where relation is denoted by edge and variable is denoted by node. A classic example is fault tree, that all the system states of interest are organized in a causal graph and fault reason can be traced in this graph.

Logic rule: Logic rules refer to a symbolic language that demonstrates a phenomenon with a set of Boolean expressions and logical operators (e.g., $\wedge, \vee, \neg, \ldots$). For example, the rule "if condition A and B exists, then event C occurs" can be described as "$A \wedge B \Rightarrow C$". The simple representation is easy for users to understand and apply.

Physics equation: Physics equation is derived from the first principle including the basic governing equations, aiming to describe the working principle of certain physics process or phenomenon. The equations are generally unique for different physics entities. For vibration analysis in mechanical systems, dynamic models are engaged based on the Newton's law.

Simulation: Simulation data refer to the high-fidelity data generated by computer software based on the detailed dynamic models derived from the first principle. The most used simulation method is the finite element method that solves real engineering problems and obtains high-fidelity data as prior knowledge.

Signal equation: Signal equation represents the perspective to analyze the signal, where the perspective can be time domain, frequency domain, and time-frequency

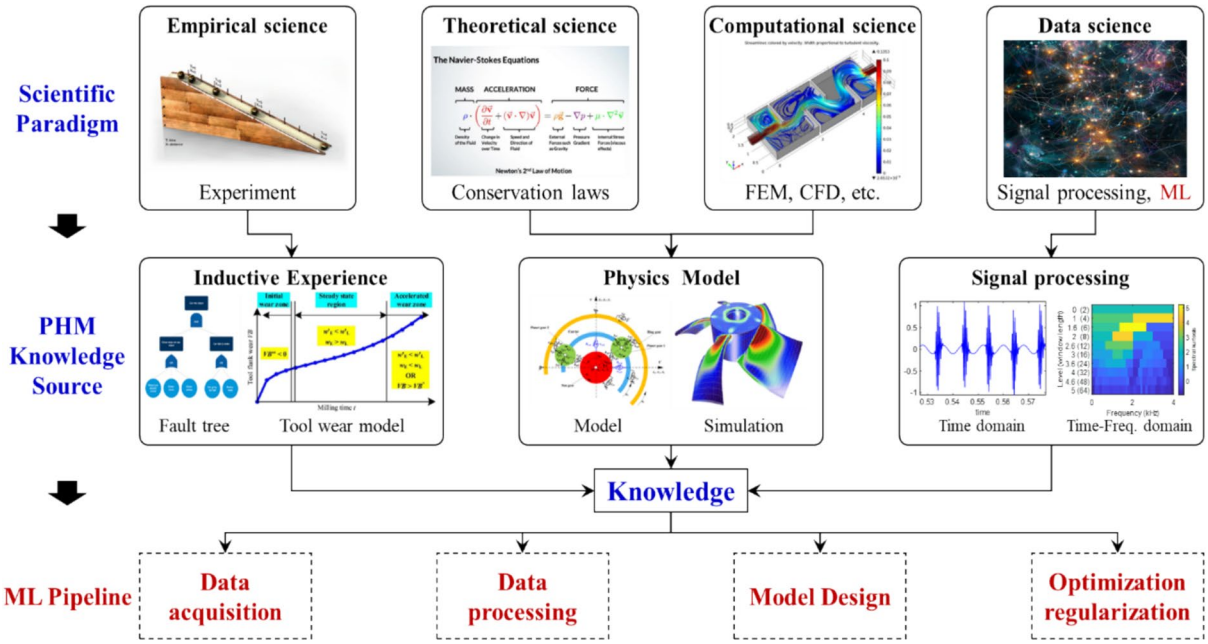


Figure 1 Knowledge sources in PHM domain

domain. For example, with Fourier transform, signal can be view as a linear combination of sinusoidal and cosinusoidal functions (complex exponentials).

Probability equation: Probability equation is capable of describing relationships among random variables. This relationship can be derived from observations and statistical analysis of likelihood. On this basic, probability equations guide observers to make predictions about the likelihood of future events.

Statistics property: Statistical property characterizes the description of behavior from a population, which can usually be achieved through mathematical tools. As a form of empirical knowledge, Statistical property reflects the understanding of the data from observers. This understanding can help observers infer the distribution, trends, and relationships of similar data population to make better predictions and decisions.

Others: Despite the above knowledge representations can cover mainstream knowledge in PHM domain, there still exists some other knowledge for specific application or expert knowledge, like attribute, hybrid model. Therefore, we group these specific methods to an additional category.

2.3 Knowledge Driven Machine Learning Pipeline

The significant difference between traditional machine learning and knowledge driven machine learning is information source for them to train a learning system. For traditional machine learning, data is the starting point of the whole pipeline and then a learning algorithm is designed to approximate the underlying function behind data. In addition, traditional approaches will consider conciseness of algorithm following the Occam's razor principle to reach a simplest hypothesis. The purpose of traditional ML pipeline can be formulated as follows:

$$\text{goal} \propto \text{task performance} + \text{algorithm conciseness}.$$

For knowledge driven machine learning, data and knowledge are two parallel information sources for the learning pipeline. As shown in Figure 1, knowledge can be integrated into each step of pipeline. Even for data acquisition, knowledge can help to optimize sensor placement or acquisition strategy. Therefore, the purpose of knowledge driven machine learning can be formulated as follows:

$$\text{goal} \propto \text{task performance} + \text{algorithm conciseness} + \text{knowledge conformit}.$$

Through integrating knowledge into ML pipeline, generalization ability and interpretability of the learning system are expected to be improved. For generalization, knowledge itself is an invariant representation with respect to distribution shift in data. For interpretability, knowledge integration will actively modify the modeling process of ML pipeline or passively test the prediction of ML models. Such active and passive interpretability is the key to achieve trustworthy AI in PHM.

We utilize a Sankey diagram to visualize the roadmap from scientific paradigms to knowledge driven ML pipeline in Figure 2. The paths illustrate the direction on how to start from a specific knowledge source to embed it into a specific module in ML pipeline. For example, the scientific paradigm of data science can derive the knowledge source in terms of signal processing, and then this kind knowledge source can be described in two kinds of knowledge representations, that is, signal equation and statistic property. Finally, signal equation or statistic property can be integrated in to ML pipeline, including data processing, model design and optimization regularization. Despite each knowledge representation has the potential to adjust each part in ML pipeline, the practical integration approach usually depends on the reality of research, and the literature review shows that model design and optimization regularization are the two main embedding parts in ML pipeline.

3 Description of Knowledge Driven Machine Learning in PHM

In this section, we give a detailed description of KDML approaches in PHM. We organize this description in a hierarchical framework, in which we first group KDML approaches by three knowledge sources, inductive experience, physical model, and signal processing and then for each knowledge source, we group the literatures by their embedding methods in ML pipeline.

3.1 Inductive Experience

Engineers would gain rules and draw conclusions from historical observation or experimentation of mechanical equipment, which is so called "expert knowledge" or "empirical knowledge" in the field of empirical science. In contrast to theoretical science, the justification of empirical science depends on a large number of experiments. Fortunately, empirical knowledges have been obtained from massive experiments with the development of PHM. These knowledges are valuable to guide the learning process and provide interpretability for the trained models. According to whether the empirical knowledge can be represented with algebraic expression, it can be categorized into experience models and inductive knowledge, as shown in Figure 3.

As the approaches to integrating inductive experience with the learning process are mainly concentrated

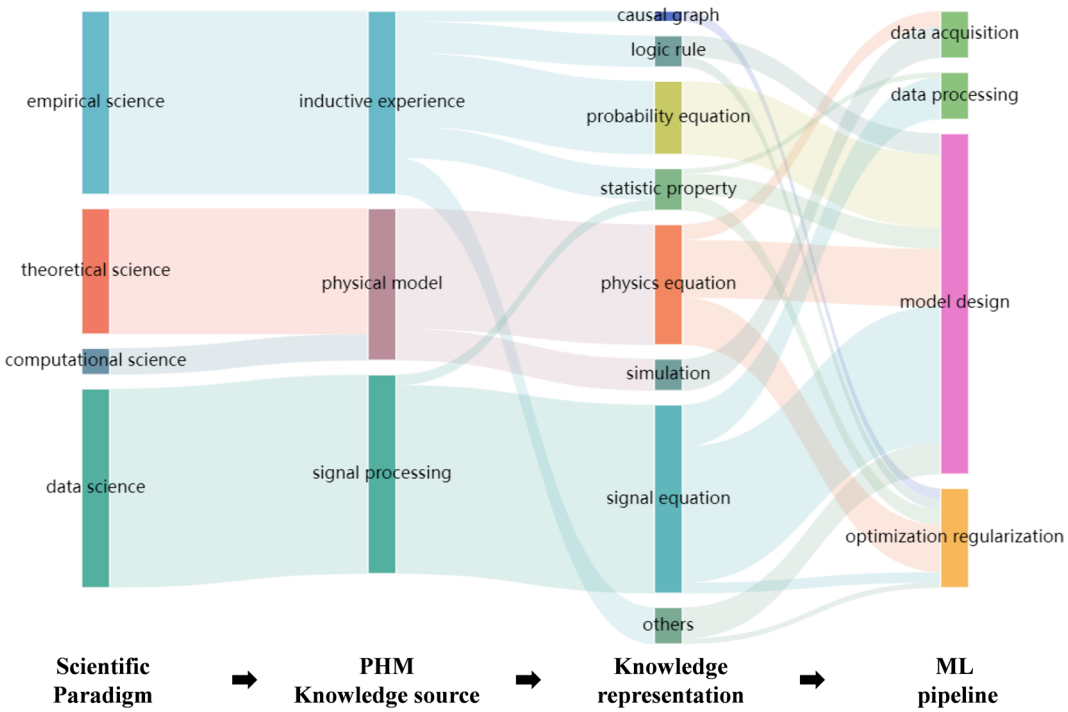


Figure 2 Overview of Knowledge driven machine learning in PHM domain

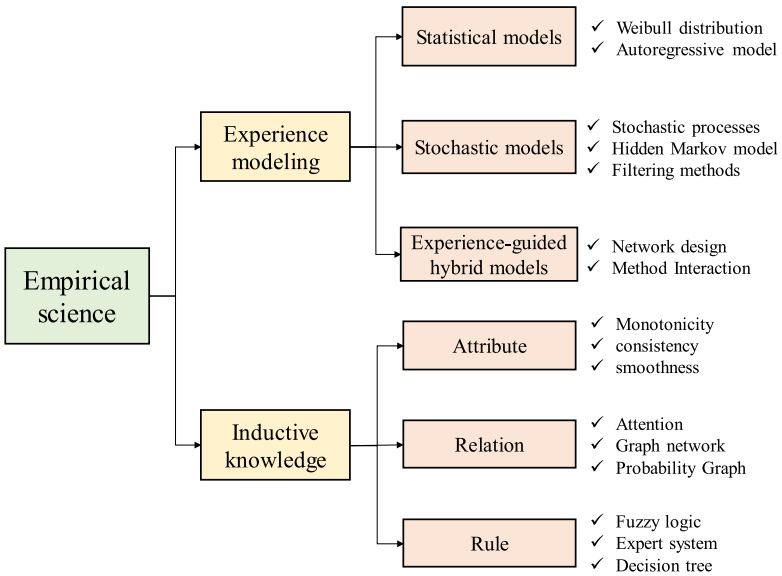


Figure 3 Classification of empirical science

on model design and optimization regularization, we do not classify inductive experience-based methods according to different stages of machine learning pipeline in this section. Instead, we divide them based on different forms of knowledge representation, as described below.

3.1.1 Empirical Model Informed Learning

Empirical modeling strives to establish a correspondence between the model and the modeled object, which represents a process of summarizing from a large amount of experimental data. Empirical models are capable of reflecting the understanding of the system being analyzed from manufacturers, which can usually be proxied in the form of mathematical models. Typically, they do not consider the underlying principles and physical mechanisms of the modeled objects. Therefore, the agent process reflects more on the dependencies of observed behaviors rather than on the inherent mechanisms by which behaviors occur. For instance, the capacity fading of lithium-ion batteries usually follows the form of exponential degradation, which is a principle derived from massive experimental observations [21]. In fact, this failure behavior is physically caused by microscopic changes in the material. This empirical knowledge can be inherited to achieve reasonable modeling of unknown lithium-ion battery. For empirical model informed learning, it integrates inductive empirical knowledge with machine learning methods to achieve good interpretability.

Empirical model informed learning can be categorized into statistical property, probabilistic equations and experience-guided hybrid models. It should be emphasized that the first two are constructed based on the empirical knowledge from the monitoring objects and are inherently interpretable [22]. That is, additional interpretable tools or modules are generally not needed to interpret the model. While for the latter, hybridization with other methods often requires unique interpretable designs. The research in this area has obtained some advancements, which will be summarized and discussed below.

(1) Statistical property

Statistical property is knowledge mined from data and formed into experience, which is expressed by mathematical form. The autoregressive (AR) model is a classic statistical model with good mathematical expression and interpretability. Many researchers have attempted to apply it in degradation modeling. This method extracts association relationships from monitoring data and extends them to the future for predicting degradation trends. For instance, Qian et al. [23] employed multi-dimensional AR model to track the extension of bearing defects, thereby achieving real-time RUL prediction. Ordóñez et al. [24] employed auto-regressive integrated moving average (ARIMA) to predict the next state of the system. Subsequently, the predicted results are fed into the support vector machine (SVM) for RUL prediction of aircraft engines. Weibull distribution is also a classic statistical model, which is usually utilized to describe the RUL probability distribution. Kundu et al. [25] utilized General Log-Linear Weibull distribution to estimate the impact of some external factors (humidity, temperature, pressure, etc.) on degradation progression. On this basis, multiple models under different failure behaviors were constructed to estimate RUL. Kundu et al. [26] further considered the influence of working conditions and degradation degree in Weibull distribution according to the engineering experience, and established a single prediction model for the entire life cycle. Experimental results revealed that modeling these two factors obtained higher accuracy.

(2) Probabilistic equations

Probabilistic equations can attempt to agent the physical evolution process from complex systems through probabilistic modeling. Among them, stochastic processes and dynamic Bayesian networks are the most common models. For example, Zhang et al. [8] utilized the Arrhenius model which is an exponential model to represent the temporal relation of physics degradation process, and induced it as the mean function of Gaussian process (GP) for prognosis of Heating, ventilation, and air-conditioning system. Jin et al. [27] used a gyroscope drift model to replace the global model in GP for prognosis, where the drift model described the fault feature of ball bearings to better capture the trajectory of degradation. Zhang et al. [28] considered uncertainty caused by external factors in degradation modeling based on the Wiener process. To be specific, unobservable factors are modeled by Brownian motion, and measurable covariates are modeled with an Ornstein-Uhlenbeck process, which is thereby linked to degradation rate. Dynamic Bayesian networks model the dependence of variables from adjacent time steps, which are generally utilized to estimate or predict the degradation state of the system. Li et al. [29] developed an improved hidden Markov model (HMM) for tool state prediction, in which changes in cutting conditions are modeled through conditional adaptive state transitions. Lyu et al. [30] believed that the capacity regeneration phenomenon is not conducive to RUL prediction of lithium battery, so variational mode decomposition is utilized to decompose the signal into trend signal and capacity regeneration signal. Then they employ particle filtering to process the trend signal for RUL prediction. Cui et al. [31] proposed a time-varying Kalman filter for bearing RUL prediction, where different degradation stages correspond to different degradation model to achieve accurate RUL prediction. Zheng et al. [32] exploited relevance vector machine to extract to predict the future system state of Lithium-ion battery

and an improved Sage-Husa adaptive Kalman filtering is constructed to enhance the filtering effect.

(3) Experience-guided hybrid models

Experience-guided hybrid models usually utilize empirically extracted knowledge as the logic to interpret the model. As a typical black box model, many scholars try to enhance interpretability by embedding empirical knowledge into the network. Pei et al. [33] combined the superiority of deep learning and stochastic process to develop a hybrid method for bearing prognosis. A deep neural network was used to extract low-dimensional feature from high-dimensional data, and then a diffusion process was used to fit the temporal low-dimensional feature and estimate the uncertainty of degradation process. Inspired by state space models (SSMs), Li et al. [34] proposed a life cycle modeling method for mechanical systems. In the proposed method, emission function and state transition in SSMs are parameterized using DNN. Moreover, the coupling competition degradation prior is embedded into the state transition network as knowledge, and the attention heat map interprets the competition relationship among the three mechanisms. Deng et al. [35] employed the Wiener process to connect the RUL prediction task with DNN and the uncertainty quantification task. This surrogate modeling method approach improves near-failure prediction accuracy in an interactive manner and provides interpretability to operators. Hu et al. [36] designed deep belief networks (DBNs) to extract hidden features and select features with high tendency to be fed into local linear embedding to construct health indicators. Diffusion process was utilized to model degradation evolution according to the health indicators and the uncertainty from prediction results could be quantified. Sun et al. [37] combined Winner process with back propagation neural network to construct a cutting tool condition degradation model, which obtained excellent RUL prediction results. Dai et al. [38] proposed an interpretable wavelet kernel network as RUL prediction model, and the Wiener process was used to evaluate reliability and provide prediction uncertainty. Experiment results conducted on a bearing dataset verified the advantages of the fusion of the two methods. Chen et al. [39] designed a hybrid prognostic approach for bearings including DNN, Winner process and Kalman filtering. More specifically, gated recurrent unit network is utilized to extract degradation representation, Wiener process is employed to adaptively update the degradation state, and Kalman filtering is designed to estimate model parameters and infer real-time RUL distributions.

Empirical model informed learning can well model or describe the behavior of monitored objects through mathematical relationships, which are completely white-box and inherently interpretable. However, empirical knowledge cannot have the high fidelity of physical models, which makes it easy to fail under unknown conditions, causing the established model difficult to generalize. Furthermore, it can be concluded that hybrid methods based on empirical models have more advantages, and this is the future research direction.

3.1.2 Inductive Knowledge Informed Learning

Induction is a reasoning process from individuality to generality. The prerequisite for reasoning and knowledge discovery is the large amount of data collected from historical experiments or daily observations. By analyzing the data and discovering their common attributes, the general rules can be obtained, which we call the "inductive knowledge". It is a similar concept to the empirical model, since the empirical models are sometimes constructed on the observed data. For ease of understanding, we distinguish them by the manifestation of knowledge in this study. That is, empirical models can be described with concrete algebraic expression, while inductive knowledge is a more abstract and broader concept that covers intuitive ideas, perspectives or assumptions concluded from rich engineering practice. It is hard to express them with relational algebraic equations. However, by integrating inductive knowledge into the learning process, the learned models can avoid overfitting in specific scenarios and provide extra interpretability for the calculating mechanisms.

There are massive attempts to embed these knowledges into the intelligent models in existing literatures. The critical difficulty of such type of work is to summarize correct, generalized and optimization-promoting knowledge, thereby improving the interpretability and meanwhile maintaining the performance. According to the different forms of inductive knowledge, inductive knowledge informed learning can be classified as attribute informed learning, relation informed learning, and rule informed learning.

(1) Attribute informed learning

In attribute informed learning, knowledge comes from some key attributes or characters that conform to the observed phenomena, e.g., the monotonicity, consistency, and smoothness of the state indicators during the degradation process of equipment, the separability between normal and abnormal frequency spectrum, and the singularity of faulty signals. These attributes are summarized from data and describe physics rules. Therefore, researchers integrate them with learning process by modifying the model structure or the optimizing objective, which

guides the model to learn some interpretable attributes. Zhou et al. [40] stated the differences between stochastic degradation process and proxy regression labels, and proposed a dynamic governing network to model the degrading trajectory of machines, where discretized ordinary differential equation was parameterized to ensure the monotonicity of the degrading trajectory. Compared with traditional methods, the predicted RUL curve is monotonically decreasing, providing more interpretability for the proposed model. Yan et al. [41] proposed an interpretable weight learning framework with two-stage convex optimization, where the first optimizing objective is to enlarge the separability of health indicators, and the second objective combines the monotonicity and fitness properties of the degrading states to generate piecewise health index. With the property constraints, the model weights reveal informative frequency bands. Motivated by the effectiveness of statistical complexity in quantifying potential dynamic changes, Yan et al. [42] proposed a weight-based sparse degradation model and introduced a set of entropy-based health indicators to quantify the interpretable model weights. Experimental results confirm that the learned weights magnify the weak fault features in online incipient fault detection. Besides, Zhou et al. [43] leveraged the advantages of singular values in revealing weak fault information, and designed graph-modeled singular values that combines graph theory and singular value decomposition (SVD). By constructing graph with singular values, the proposed method realizes a balance between sensitivity to early fault and robustness to noise.

(2) Casual graph informed learning

Although the operating mechanisms of complex systems are hard to obtain, the dependence, causality or logical relationship between variables can sometimes be induced according to observations or physical structures. This form of knowledge can be viewed as an auxiliary constraint to regulate model training, which also endows the model with partial interpretability. The approaches to knowledge integration mainly focus on designs of model structures, such as using attention mechanism, selecting graph neural networks, constructing probabilistic graph model, etc. For example, Ma et al. [7] proposed a graph-based abnormal sensor localization and fault detection method for lithium battery packs, which constructs a graph autoencoder based on the layout of voltage sensors. The reconstruction of both signals and the graph structures are required, which greatly reduces the time delay for fault detection. Li et al. [44] focused on the impulse characteristic of vibration signals and argued that the attention weights to different segments should follow sparse distribution. Sparse constraints are then added in the Transformer encoder. Post-hoc attention visualization reveals that attention weights concentrate on the fault related impulse. Bi and Zhao [45] proposed an orthogonal self-attentive variational autoencoder model, where the casual relationship between process variables and the temporal dependency can be extracted via spatial and temporal self-attention layer. Liao et al. [46] proposed a self-attention assisted physics-informed neural network (PINN) for aeroengine life prediction, where self-attention mechanism is embedded into PINN to learn more accurate physical relationship. Martel et al. [47] mapped the monitored data and time into the latent variable by partial differential equation. The latent features are then used as health approximation to explicitly deduce the temporal behavior of data. When dealing with complex problems like health management in varying working conditions, the causality between health states, working conditions and signals can serve as prior knowledge to assist model training. Li et al. [48] proposed a casual disentanglement network to realize cross-machine knowledge generalization, which combines domain separation loss, classification loss and disentangled loss to capture invariant fault information. Hu et al. [49] utilized information theory to deduce the equivalent optimization objective to extracting condition-independent features, and designed an adversarial training manner by minimizing variational upper bound and self-supervised learning. The learned model can reduce false or missing fault detection and adapt to time-varying scenarios.

(3) Logic rule informed learning

Logic rule informed learning expect to embed rule-based knowledge into model learning. Various expression forms of the extracted rules (e.g., fuzzy logic, decision tree, expert systems) can be inserted as prior knowledge into the deep networks, thereby endowing the network with interpretability. For example, Wong et al. [50] proposed a fuzzy extreme learning machine for industrial fault diagnosis, where fuzzy membership functions, rule-combination matrix are embedded into the extreme learning machine. Without loss of accuracy, the output weights are utilized to form the class and confidence for any rules, providing explicit knowledge in an interpretable manner. Yu and Liu [51] inserted the inductive symbolized rules and confidences into a deep belief network, which enables the model to determine the network adaptively. Wu et al. [52] proposed a cluster-based hidden Markov model to learn the mapping between critical performance index and RUL. Then a semantic rule-based inference module is attached to recognize the root factor for performance degradation. Steenwinckel et al.

[53] proposed a knowledge and data joint driven anomaly detection model to improve the model's representation ability. According to abnormal feedback, knowledge driven methods can generate new rules to realize the adaptive update. Zhou et al. [54] argued that the reasons for decreasing interpretability are parameter over-optimization and the deviations to expert judgements. They evaluated model interpretability by measuring the consistency index of rules, consistency index of rule set, and over-optimization index, achieving both high accuracy and good interpretability. Furthermore, Ming et al. [55] proposed an interpretable diagnosis model integrating both rule base and probability table. Interpretability constraints are added to the adaption evolution strategy of covariance matrix to ensure the model interpretability.

3.1.3 Summary for Inductive Experience in KDML

By contrast to signal processing techniques and physics models that explicitly describes the properties of signal and physical entities, inductive experiences are observed and concluded from historical phenomena. Their interpretability is less explicitly revealed compared to the former two sorts of knowledge. However, there are massive regular patterns implicitly hidden in engineering practice. By integrating them into the optimization process, the model would avoid overfitting and learn the core relationship between input and output. The interpretability and performance of model would be improved simultaneously.

3.2 Physical Model

Physics informed machine learning (PIML) incorporates physical knowledge and data-driven machine learning model. It starts from the observation of working principles for the physical entity, establishing physical models that describe certain physics phenomena or process. Physical models are usually governed by differential equations, and these equations could be derived into different forms for different orientations, i.e., state space models, discrete difference forms, finite element analysis, etc. Physics informed machine learning constructs physical models from the first principle, aiming to figure out the basic governing equations for physical phenomena or process as prior information for machine learning.

Data-driven models establish the relation between inputs and outputs through violent mapping, leading to the lack of interpretability. However, in PIML process, physical models reveal the causality between system excitation and response by governing equations, which assigns physical meaning to machine learning results. Physics knowledge observed from the corresponding physical models is usually unique, and preserves within one or a group of physical entities. For the battery RUL prediction problem, electrochemical models and equivalent-circuit models based on Kirchhoff's law are established to relate the state of health and discharge time. Besides, for most mechanical systems, conservation laws are governing the major parts, including mass, momentum, and energy conservations. As for the degradation process, there are certain phenomenological models for different scenes, such as Paris law for crack propagation. The following works show the discovery mechanism for physics knowledge starting from the first principle. Luo et al. [56] conducted a dynamic model of a mechanical system and assumed that concerned stiffness is coupled parameter of the degradation variable. By using a polynomial function to approximate this relation, this method can reveal the degradation process. Mojallal and Lotfifard [57] utilized Hybrid Bond-Graph theory to construct a multi-physics graphical model of a mechanical system to capture its causal relation, and this method was developed for fault detection and isolation in wind turbines. The mechanism shares the same for different physics entities. Thus, through the engaging methods of physical knowledge and learning process, PIML methods can be divided into physics informed data augmentation, physics informed network architecture design and physics informed loss function construction, as is shown in Figure 4.

3.2.1 Physics Informed Data Augmentation

Physics informed data augmentation aims to incorporate physics knowledge for class imbalance problems in the field of interpretable intelligent diagnostics. High-fidelity physical models are essential to generate time series data, and these models typically composed of the simulation model-based approach which utilizes computers to obtain time series data by solving physical models. To ensure the fidelity, simulation models pay more attention to the details in physical systems, apart from the general conservation equations applied in other subsections. Thus, these models generally require large computational costs and subject to computer science paradigm. The simulation approach focuses more on understanding and interpreting the physical system, and needs to consider the physical meaning and practical application requirements, so as to ensure the interpretability and validity of the model. It is worth noting that while there are also generative models based on GAN and other generative models to generate virtual data, we do not consider this a simulation model when considering that it has no actual physical significance. However, if it is possible to further add physical constraints to the GAN to improve the results, in which case the GAN itself is driven by knowledge, we support such a paradigm.

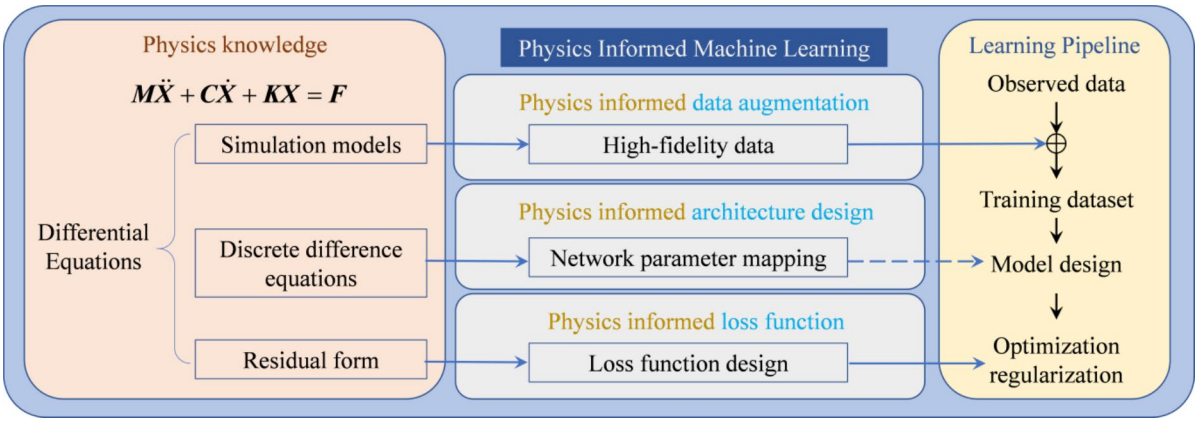


Figure 4 Classification of physics informed machine learning

Researchers have conducted extensive studies on simulation model-based KDMLs. Some scholars use the centralized parameter method to model physical systems and solve simulation models to obtain data to guide the network to learn the knowledge that has physical meaning. Cui et al. [10] developed a comprehensive dynamic model by dividing the degradation process of rolling bearings into four different stages while considering the coupled excitation of time-varying morphology and stiffness. The response of the simulation model was solved to obtain a large amount of performance degradation data for RUL prediction. Yu et al. [58] established a dynamics model of the rotor-bearing system, constructed a source domain dataset, and utilized the diagnostic knowledge obtained from the simulation data to realize domain adaptive transfer learning for mechanical equipment. Ni et al. [59] utilized modal-property-dominant features to learn the underlying physical knowledge embedded in the training and test data, and proposed a physical-informed residual network that considers the physical information in the data.

Since the lumped parameter method simplifies the physical system too much, in order to obtain higher fidelity data, other studies use the finite element method to construct the simulation model and extract a priori information about the physics. Kohtz et al. [60] developed a physically based finite element model for lithium batteries and fused the finite element results with experimental data to construct a multi-fidelity model for battery state of health estimation. He et al. [61] used numerical simulation data of fluid-structure interaction in place of actual monitoring data to realize the RUL prediction of centrifugal pumps based on a ladder network. Agarwal et al. [62] developed a finite element analysis model of the electric motor in an electric vehicle, which introduced faulty defect conditions, obtained surrogate current data from the simulation model, and finally classified the faults using a support vector machine. Zhao et al. [63] developed a finite element model of gears to further obtain a crack propagation model so that the distribution of failure time and RUL can be predicted. Then, the collected condition monitoring data are fused by Bayesian method to estimate the crack length and realize more accurate RUL prediction.

In addition, there are other domain-specific simulation model modeling approaches that also allow access to high-fidelity data. The best known of these is the C-MAPSS simulation model, which provides degradation monitoring data for turbine engines and is widely used for learning and training in the field of remaining life prediction. López de Calle-Etxabe et al. [64] used Matlab Simcape to build a physical model of a linear actuator for generating data in health and failure scenarios, and fused the simulated data with actual measurements to establish a diagnostic model. Djeziri et al. [65] used Bond Graph to physically model the wind turbine to obtain data on normal operation and failures, and used this data to predict the RUL based on geolocation principal.

Physics informed data augmentation approach typically contributes very little to explain the final output of the model or the structure of the network, but they learn by applying the data generated from modeling scientific knowledge in the domain. This introduces prior knowledge of data distribution into the model, corrects the initial iterations of the model, and guides the model to use the priors to discover interpretable knowledge in the actual data. However, the difficulty of this method is that a reliable physical model of the system needs to be established for the simulation data to increase its similarity to the observed data, further ensuring the accuracy of the model iteration direction.

3.2.2 Physics Informed Network Architecture Design

Physics informed network architecture design combines the mathematical calculations of the network model with the solving process of the physical model, assigning specific physical meanings to network parameters, thereby enhancing interpretability. Chen et al. [66] introduced an ordinary differential equation solver based on the residual connection in residual networks or the recursive calculation in recurrent neural networks. This method employs the discrete form or the recursive form of differential equations to neural networks, mapping network parameters to the physical quantities in the equation. Based on this approach, researchers started to study the equations governing health state of certain equipment and incorporated physics equations and neural networks to fault diagnosis and health assessment. Dourado and Viana [1] developed a physics-informed neural network for bias estimation in corrosion-fatigue prognosis. This method utilized the Paris Law to describe the phenomenon of corrosion-fatigue crack propagation process, and used a neural network to model the corrosion part. In this work, the physics knowledge provides the relation of variables in prognosis model while the neural network provides a parameter update method in physics model. Nascimento et al. [67] provided a tutorial on integrating ordinary differential equations into RNN and used two case studies to illustrate it, that is, a fatigue crack growth model and a dynamic two-degree-of-freedom system. The proposed method worked as the classical system parameter identification. Firstly, an evolution model of physical variables is derived based on physics knowledge, such as conservation laws or constitutive relation. Then, a neural network is utilized to fit some variables in the evolution model which are difficult to model or observable. Nascimento and Viana [68] integrated cumulative damage model into RNN for crack length prediction of a fleet of aircraft, and they developed several RNN cell styles to combine physics with data. Yucesan and Viana [69] buried degradation model for bearing fatigue in RNN cells and calibrated the proposed model by visual grease inspection. Besides, Yucesan and Viana [70] improved their bearing fatigue prognosis model by considering the uncertainty caused by grease quality variation. Yucesan and Viana [71] further proposed a hybrid method for wind turbine main bearing fatigue prediction, where a physics model was used to model the L10 fatigue life and a neural network was used to model the grease degradation process. Tipireddy and Tartakovsky [72] presented a physics-informed Gaussian process for monitoring and forecasting of power grid dynamics. The proposed method utilized a random process to model the evolution of unknown physical variables and then compute the covariance matrix from it. Samundsson et al. [73] proposed a variational integrator network to model dynamic system, in which the structure of the neural network matches the discrete-time equation of motion. Chao et al. [74] proposed a hybrid prognosis method by fusing physics features with data features. This fusion first used unscented Kalman filters to estimate the unobservable model parameters, and then merged and fed these parameters with measurements into a deep neural network for prediction. Nascimento et al. [75] developed a hybrid method integrating a reduced-order physics model with RNN for lithium-ion battery modeling and prognosis. This method utilized the Nernst and Butler-Volmer equations to represent the battery discharge process and encoded this reduced-order model into the RNN cell structure.

The design of physics informed network architecture makes the combination of physical knowledge and neural networks more closely. The mapping of physical models and the mathematical models of neural networks assigns the internal parameters of the network with physical meanings and further brings physical interpretability.

3.2.3 Physics Informed Loss Function

Physics informed loss functions are derived from governing equations based on the first principle, constricting the optimization direction of neural networks, and further ensuring the interpretability of learning results. Zhang and Liu [9] established a Parsimony-enhanced sparse Bayesian learning method to discover the governing Partial Differential Equations (PDE) of nonlinear dynamic systems and validated its application in anomaly detection. This method utilized Bayesian inference method to reduce the information loss before the sparse regression procedure. Lutter et al. [76] developed a Deep Lagrangian Network to learn the equations of mechanical motion (i.e., robot tracking control) while ensuring physical plausibility. This method derives the loss function of the neural network from the Lagrangian equation and applies symmetric and positive constraints to the parameter matrix based on the properties of physical quantities. Zhang et al. [77] incorporated the physics knowledge into neural network from the loss function aspect. This method leveraged available yet incomplete physics information, such as governing equation or states relation, to match network parameters and physical ones, and then encoded a physics loss to constrain the solution space of the neural network. Further, Zhang et al. [78] embedded physics knowledge into a CNN for building structure response prediction. This method established a dynamic model to derive a physics loss term, so as to alleviate overfitting problem of CNN. Unfortunately, due to the agnostic of partial variables, this physics loss can only be verified on mathematical simulation but cannot

be used on the real engineering data. Chen et al. [79] proposed a degenerate consistent recursive network to study the physical characteristics of bearing degradation process, and trained the network through loss function constraints to improve the accuracy and interpretability of bearing fault prediction results. Freeman et al. [80] raised a new physically guided rotor blade imbalance fault detection framework, which combines non-invasive fault features obtained from turbine power signals with environmental condition data to customize loss functions for neural networks to enhance fault detection capabilities. Shen et al. [81] established a fault threshold model based on physical knowledge and combined it with a neural network model. By designing a loss function, the influence of physical knowledge is selectively amplified to achieve bearing fault detection. Xu et al. [82] proposed a physically constrained variational neural network for evaluating the wear status of external gear pumps. In this method, spectral methods are involved to establish a pressure pulsation model for gear pumps which is further transformed into a physical loss term, constraining the learning process of neural networks while enhancing physical meanings of learning features. Wang et al. [83] designed physics informed loss function based on the fluid mass conservation law in machine learning process, connecting the learning results to piston wears of axial piston pumps, and realizing the interpretable health assessment.

The loss function determines the direction of network optimization, and the addition of physical loss functions directly constrains the network optimization process with physical knowledge. Designing loss functions based on the different tasks and target systems and balancing the weights of each item in the loss function according to the order of magnitude of the described physical quantities makes the combination of physical knowledge and neural networks clearer and enhances the interpretability of the network.

3.2.4 Summary for Physical Model in KDML

Physics informed data augmentation utilizes the data generated by the simulation model to train the deep network model, allowing the model to obtain a reasonable initial value prior. At the same time, this method relies heavily on the quality of the data from the simulation model, which makes it difficult to obtain high-fidelity simulation data when dealing with extremely complex physical systems. The physics informed loss function, on the other hand, introduces the physics loss as a regularization term to guide the optimization direction of the network. While controlling the direction of the network output to make it more consistent with physical laws, the regularization term also reduces the search space of the network parameters and increases the interpretability of the model's predicted output. However, the physical models they build are generally low-fidelity, and it is extremely difficult to derive a high-fidelity physical model into the formula form of the regularization term. In addition, the internal structure of the network of the above two methods is still a black-box model. In contrast, the physics informed network architecture design happens to enhance the interpretability of some parameters in the network structure. But it usually requires different network structures in different scenarios, and the network structure is less generalizable. The above three methods increase the interpretability of the network in terms of initial value selection, optimization direction and model structure, respectively. Nevertheless, for highly complex physical models, how to further embed the knowledge into the network according to the above three methods needs further research.

3.3 Signal Processing

Signal processing informed neural networks (SPINN) is an interpretable network that combines the prior information of signal processing technology and the data-driven capabilities of deep learning. The prior information comes from the signal analysis and feature extraction methods that have been extensively developed and widely used in the field of health management, such as time domain, frequency domain, and time-frequency domain analysis techniques. Since signal processing prior has been validated in industrial applications, SPINN can often obtain better performance with a reasonable prior. In addition, SPINN can be easily interpreted from the signal processing perspective and thus better understood by users.

From the constituent element of network training, there are three kinds of methods to embed the signal processing prior into the network and construct SPINN, i.e., data, model, and optimization, as is shown in Figure 5.

1) Data: signal processing methods can be easily used for data filtering, data enhancement, and feature extraction. The processed data are then directly used by the network for health management.
2) Model: signal processing methods are used as a component in a network with fixed/learnable parameters or guiding theory to design new network structures.
3) Optimization: signal characteristic priors obtained by signal processing methods are used to generate regularization on the optimization target.

Recent work on SPINN is introduced from the three aspects.

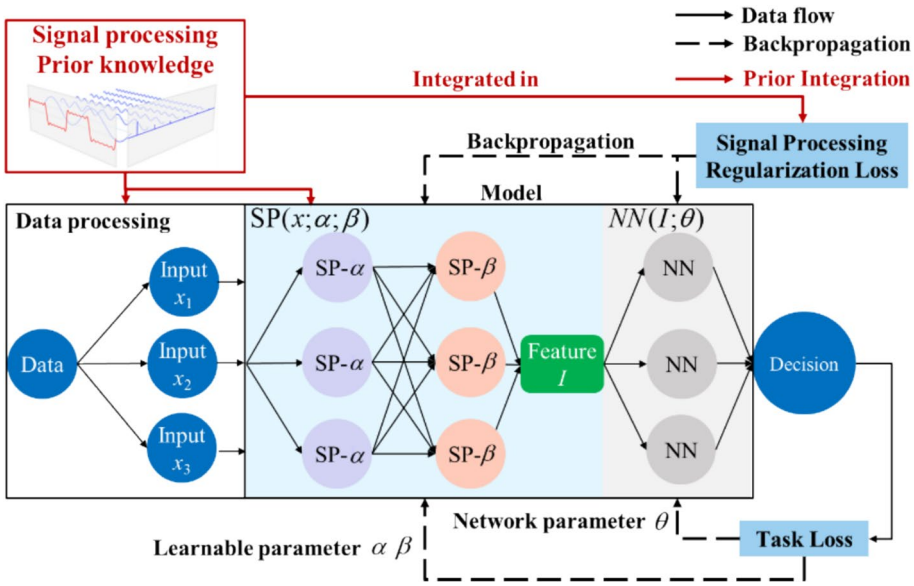


Figure 5 Signal processing informed neural networks

3.3.1 Signal Processing Informed Data Preprocessing

As the simplest way to realize SPINN, data processed by signal processing method can directly introduce prior information without influencing the model and optimization.

1) Data transform or filtering realized by Fourier transform, short-time Fourier transform, wavelet transform, and so on can provide another view of the signal which is better for health management tasks. Lahcen et al. [84] utilized stationary wavelet transform to extract features from raw ECG signals and then input the decomposed wavelet sub-bands into a 1D-CNN for heart disease diagnosis. Wen et al. [85] applied S transform, a time-frequency technique, to transfer the vibration signals to images, and then input into a CNN to classify these images for fault diagnosis. Li et al. [86] utilized short-time Fourier transform to get time-frequency representation of the vibration signal. Then a CNN is used to extract multiscale features and realize RUL prediction.
2) The adjustable parameters of the signal processing method and the procedure of signal analysis and synthesis provide a chance for data augmentation. To solve class-imbalance problem in fault diagnosis, Fu et al. [87] proposed local wavelet similarity fusion to augment high-quality fault data. Wavelet packet transform is used to decompose signal into different frequency bands and the amplitudes of wavelet coefficients in selected bands are distorted by a similarity-based weighting strategy with reference samples. Signals reconstructed from the distorted coefficients serve as the augmented samples. Due to the scarcity of labeled failure data, Kulevome et al. [88] utilized analytic wavelets to obtain augmented scalograms. By successively adjusting the predefined decay parameter of the generalized Morse wavelet, scalograms with different energy concentrations are obtained. Then augmented scalograms and the original scalograms are combined as training data for model optimization.
3) The last way to embed signal processing priors in data is feature extraction. Since a large number of feature extraction and indices design methods based on expert experience have been developed, effective utilization of these methods can embed priors in input and simplify the model's feature extraction module. Meng et al. [89] proposed an ensemble learning method for online joint strength prediction in ultrasonic metal welding. The variability in joint strength is decomposed into a large-scale term characterizing the influence of physics conditions and a small-scale residual term characterized by online sensing data of power, displacement, microphone, and acoustic emission. Discrete wavelet transform is used to extract features from sensing data, which is used by multilayer perceptron (MLP) for tool condition prediction and gradient boosting machines for prediction. Zhao et al. [90] proposed a local feature-based gated recurrent unit network for machine health monitoring. Classical time-frequency methods are used to extract typical features, which are used by a

GRU network and fully connected layers for prediction and diagnosis. Zhu et al. [91] used wavelet transform to get the time-frequency representation of vibration and then input it into a multiscale CNN for bearing remaining useful life (RUL) prediction. Ren et al. [92] utilized several time-frequency techniques to extract features from vibration signals and then input them into a deep neural network for bearing RUL prediction.

3.3.2 Signal Processing Informed Model Design

Currently, the primary approach to implementing SPINN is embedding the signal processing method in the design of models, which deeply integrates the priors of signal processing methods and the data-driven capabilities of deep learning.

1) A simple approach to designing a prior informed model is to directly use signal processing methods as processing or activation layers in the network, with fixed parameters and clear physical meanings. Sadoughi et al. [93] utilized spectral kurtosis and envelope analysis methods to extract sidebands from raw signals and minimize non-transient components. Then a fixed convolution layer designed based on shaft speed and characteristic frequency is utilized to extract fault features from the demodulated signal, which are input into a CNN model for bearing fault diagnosis. Based on the multi-resolution analysis, Jiang et al. [94] proposed a 4D wavelet convolution layer to process infrared thermography, characterizing complex spatiotemporal degradation-related features. Then the features are denoised by a deep image stream denoiser layer and utilized for RUL prediction. Wang et al. [95] embedded discrete wavelet transform (DWT) as a frequency mapping layer in the network. After obtaining wavelet representation, data-driven convolution and frequency attention are utilized for feature extraction, enabling noise-robust fault diagnosis. Liu et al. [96] introduced Harr wavelet transform as a signal preprocessor of the discriminator part in the structure of a generative adversarial network (GAN), enabling higher data generation quality. Ren et al. [97] integrated dual tree complex wavelet into transformer to obtain shift invariance and more discriminative features.
2) Making key parameters of signal processing methods trainable or partially networked is an important approach to implementing end-to-end data-driven SPINN.

Due to the equivalence of convolution and filtering, a series of works have emerged that parameterize the convolutional kernels using signal transformation theory. Ganguly et al. [98] utilized wavelet kernels to initialize the CNN kernels and construct a specific feature learning by neural architecture design, to detect and discriminate signal or multiple partial discharge locations in high voltage power apparatus. Li et al. [99] utilized a continuous wavelet kernel to replace the first convolutional layer of CNN to design meaningful filters in an end-to-end manner and then realized machine fault diagnosis. Chen et al. [100] embedded different time-frequency transforms into convolutional layers as different trainable kernels, realizing a first-layer physically interpretable model for fault diagnosis.

Signal denoising often requires parameters that are adapted to the signals. Data-driven approaches are well-suited to address this issue effectively. Shang et al. [5] proposed the concept of SPINN and designed a wavelet denoising network for fault diagnosis. Zhao et al. [101] analyzed the regularization constraints of denoising problems in the reproducing kernel Hilbert space (RKHS). The parameters controlling the bandwidth and signal smoothness are selected as trainable parameters and designed as an interpretable denoising layer. Zhao et al. [102] implemented denoising based on the soft thresholding method, where the threshold is adaptively determined by a network branch. Shao et al. [103] improved the autoencoder by replacing the activation function with Morlet wavelet function, enabling better feature extraction for fault diagnosis. The wavelet parameters controlling frequency bandwidth and central frequency are optimized by the fruit fly optimization algorithm.

3) Signal processing algorithms and the feature extraction procedure can be reformulated as network structures. Based on the similarity of the convolutional network and wavelet transform, scattering transform is proposed by Mallat [104, 105]. Furthermore, Liu et al. [106] developed a normalized wavelet scattering convolutional network for fault diagnosis. Informed by signal time-scale representation theory, Kim et al. [107] designed a health-adaptive time-scale representation layer and embedded it into CNN for gearbox fault diagnosis. Based on the lifting wavelet transform theory, Pan et al. [108] designed a LiftingNet for adaptative feature learning from noisy data, which is used for bearing fault diagnosis. The predictor and updater are realized by the convolutional layer. Yuan et al. [109] designed smart lifting wavelet kernels with strict theoretical constraints from signal processing theory, enabling impact fault feature extrac-

tion for machine fault diagnosis. To track mechanical degradation, Jiang et al. [110] proposed learnable lifting scheme with spatiotemporal dynamic convolution layer. The infrared thermography is proposed by the lifting scheme and the fused subbands energy features are used for degradation prediction. Informed by DWT, Fink et al. [111] developed a learnable deep discrete wavelet transform for high-frequency time series analysis in machine monitoring. This method integrated the fast discrete wavelet transform into an unsupervised autoencoder framework, which makes both the wavelet bases and denoising thresholds fully learnable. Inspired by the similarity between DWT and autoencoder, Shang et al. [112, 113] proposed learnable M-band wavelet network to obtain discriminative reconstruction error between normal and abnormal signal. Zhao et al. [114] developed a model-driven deep unrolling method for fault diagnosis, which unrolled a corresponding optimization algorithm into a neural network with the characteristics of interpretability and noise robustness. An et al. [115] unrolled a sparse coding model and developed an adversarial algorithm unrolling network for anomaly detection. The encoder and decoder in the generator are designed based on the sparse coding algorithm. Based on morphological analysis, Ye et al. [116] developed a deep morphological convolutional network with learnable structure elements for gearbox fault diagnosis.

Feature extraction priors can also be realized in a network structure. Wang et al. [117] incorporated wavelet transform, square envelope, and Fourier transform into the input layer of extreme learning machine (ELM). Sparsity measures are induced into the hidden nodes of ELM to establish an interpretable neural network for machine condition monitoring. Borghesani et al. [118] embedded the signal processing method of gear diagnostics into the network and achieved adaptive spectrum editing. A fault index related to the average log-ratio is obtained through the network. Based on signal processing priors on the bearing, Lu et al. [119] developed a weighting layer to assign higher weights for features located closer to the bearing fault characteristic frequency. Xie et al. [120] designed frequency learning branches and corresponding loss functions for different fault signals. This allows the network to adaptively extract frequency priors and combine them with deep features for fault diagnosis.

To discover mappings between two infinite-dimensional function spaces, Rani et al. [121] proposed wavelet neural operator, a data-driven framework consisting of uplifting transformation, wavelet integral block, and downlifting transformation. Then generative adversarial wavelet neural operator is constructed to obtain the distribution of multivariate time series data and the reconstruction error is used as a fault indicator.

3.3.3 Signal Processing Informed Optimization Constraint

By leveraging signal processing methods, signals can be described from different perspectives, allowing for additional constraints to be imposed on network training.

With extra frequency information reconstruction loss, Russell et al. [122] enhanced the data compression performance of deep autoencoder for industrial condition monitoring. Yao et al. [123] utilized the statistical property of signal in the wavelet domain to weight different scale coefficients, applying a regularization constraint on the temporal reconstructor. This regularization enables the model to capture both the temporal and frequency patterns of the signal. Dai et al. [124] proposed an acceleration-guided acoustic signal denoising framework based on a learnable wavelet transform for slab track condition monitoring. The acceleration-guided wavelet feature alignment constraint introduces clean signal information and improves the robustness of condition monitoring.

3.3.4 Summary for Signal Processing in KDML

In conclusion, although embedding priors into data can be easily realized, the selection of priors relies on expert experience and cannot be modified in a data-driven manner. Once the selection of the prior is incorrect, it can easily impact the performance of the model. When embedding knowledge into the model design, the most popular way is to parameterize signal processing methods. As for optimization loss design, priors generally come from the distribution characteristics of the signal in a particular representation domain. Moreover, this often overlaps with the previous two methods because introducing priors from a signal processing perspective often requires the object being optimized to have been processed by signal processing methods.

4 Case Study

In this section, we provide four case studies to show the result KDML can bring in PHM, including inductive experience, physical model, and signal processing. In each case study, we describe its knowledge source and representation, knowledge embedding approach, and experimental result.

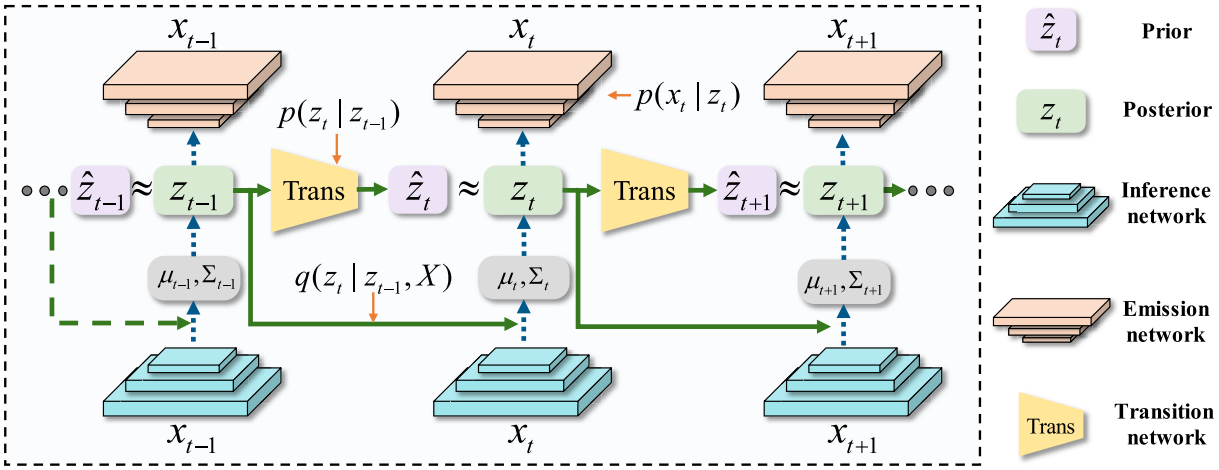


Figure 6 Framework of $C^2D^2M^2$ [34]

4.1 Life-Cycle Modeling Driven by Coupling Competition Degradation

4.1.1 Knowledge Source and Representation

In this subsection, an empirical model informed learning method constructed by Li et al. [34] is described in detail, whose name is Coupling Competition Degradation based Deep Markov Model ($C^2D^2M^2$). The overall network framework is inspired by the probabilistic graph of SSMs. As illustrated in Figure 6, the representation ability of $C^2D^2M^2$ can be improved by parameterizing the emission function and transition function in SSM. Variational inference is employed to estimate true degradation states, which is realized through the inference network in Figure 6. The specific objective loss function can be found in Ref. [34]. The source of knowledge is the prior assumption of SSM on the degradation process, which is the two independent assumptions of HMM, the homogeneous Markov assumption and the observation independence assumption. To be specific, the observed variable at a certain moment depends only on the latent state variable at that moment, while the latent state variable at a certain moment depends only on the latent state variable from the previous moment. This dependence is represented by designing a unique neural network, which can infer the hidden degradation state from noisy observation signals. Moreover, the knowledge of coupling competition degradation mechanism (CCDM) induced from experimental phenomena is embedded into the transition network as prior. The empirical knowledge of CCDM is that the degradation of entities generally does not obey a single mechanism, but a coupling of multiple mechanisms, among which there is a competitive relationship.

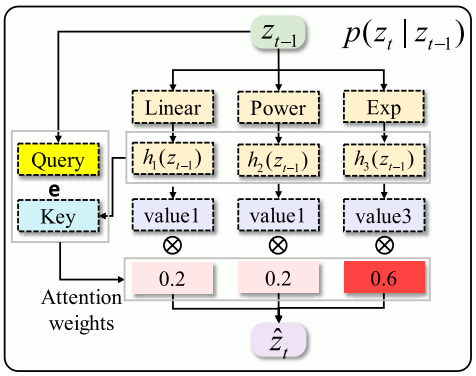


Figure 7 Transition network structure [34]

4.1.2 Knowledge Embedding Approach

The difference equations from three degradation mechanisms are encoded into the transition network, including linear degradation, exponential degradation and power rate degradation. Figure 7 illustrates the specific embedding approach. The differential equations of the latent degradation states at adjacent moments are derived based on these three degradation modes. In the transition network, the latent state z_{t-1} at the previous moment is fed into the differential equation to estimate the latent state at the next moment. The query matrices are obtained by linearly projecting z_{t-1}, and the outputs of the three degradation modes are linearly projected to obtain the key matrices. The dot product operation of query and key matrices is utilized to calculate the similarity and obtain the attention weights. The attention weights are then employed to perform a weighted summation of the outputs from three degradation modes to obtain prior estimation $\hat{z}_t$ at the next moment. The competition relationships can be reflected in the optimization of attention weights.

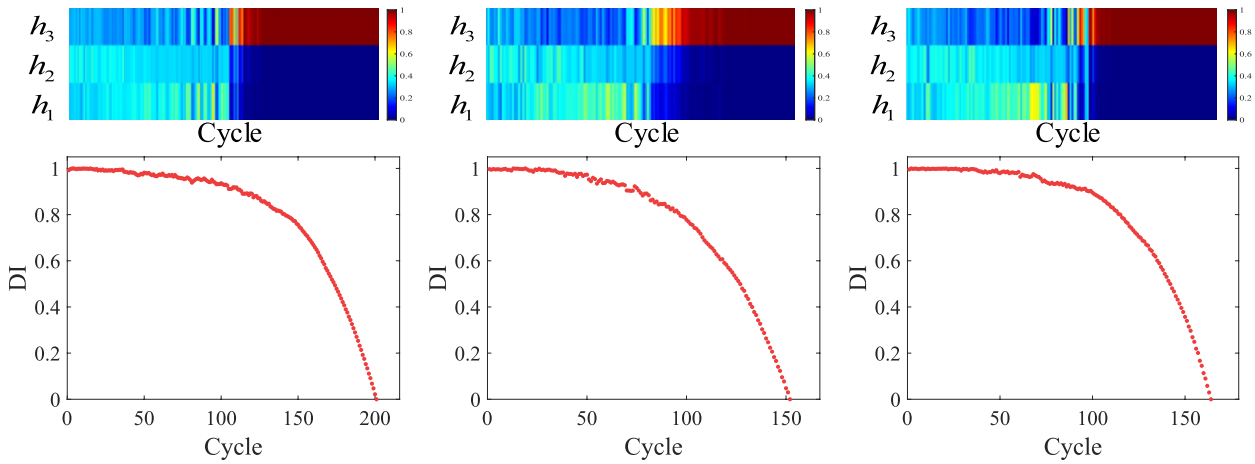


Figure 8 DI curves from three instances and the corresponding attention weights [34]

Table 1 RMSE comparison with existing methods

Algorithms	FD001	FD002	FD003	FD004
AE	14.40	23.16	17.42	24.05
VAE	14.70	23.25	18.12	24.17
DMM	14.00	22.26	16.22	23.16
BiGRU-AS [126]	13.68	20.81	15.53	27.31
SUR-TSMAE [127]	14.46	21.10	17.16	22.61
BiLSTM-ED [128]	14.74	22.07	17.48	23.49
$C^2D^2M^2$	**12.32**	**20.81**	**15.32**	**22.43**

4.1.3 Experimental Results

In the experiment, Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset is employed for validation and the detailed dataset information can be found in Ref. [125]. Figure 8 shows one-dimensional degradation curves generated from several engines, which is inferred from latent features. The attention weights learned by the network are visualized accordingly, where linear, power rate and exponential degradation are denoted as $h1$, $h2$ and $h3$, respectively. For the early stage, $h1$ and $h2$ obtains higher weights, indicating a tendency towards linear and power rate degradation. As for the later stage, the attention weights from exponential degradation are the highest, which reveals that it dominates the competitive process. The network architecture itself is developed based on SSM and is therefore interpretable. Post-hoc attention heat map analysis indicates the evolution of engine operating conditions during the degradation process, which can also provide interpretability for monitoring personnel. To illustrate the superiority of embedded knowledge, the experiment also designed a RUL prediction framework based on similarity matching. Comparison approaches include autoencoders (AE), variational autoencoders (VAE), deep Markov models (DMM), and three existing methods. Table 1 shows the comparison results of the root mean square error (RMSE) from seven methods. It can be observed that the RUL prediction results of $C^2D^2M^2$ on the four sub-datasets are the best (bold values). This shows that embedding the degradation pattern prior can encourage the model to learn degradation indicators that reflect the true latent degradation state.

4.2 Physics Informed Neural Networks for Fault Severity Identification

Physics knowledge is derived from the first principle of target systems. To introduce the detailed incorporating process of physics knowledge and machine learning, a PINN model for axial piston pump fault severity identification by Wang et al. [83] is chosen as an example. The overall framework of fault severity identification is shown in Figure 9.

4.2.1 Physics Knowledge Derivation

This work concentrates on the fluid mass conservation law within a control volume as the first principle, aiming to analyze instantaneous pressure change and figure out the basic governing pressure build-up equation as prior physics knowledge. The objective control volume is chosen as a section of fluid pipeline at pump outlet.

Pressure analysis is based on the Euler method and lamped parameter approach, which declare the assumption of uniformly pressure distribution within the control volume. The physics knowledge in this case is derived as a differential equation of pressure p, time t and the physical parameters to be determined.

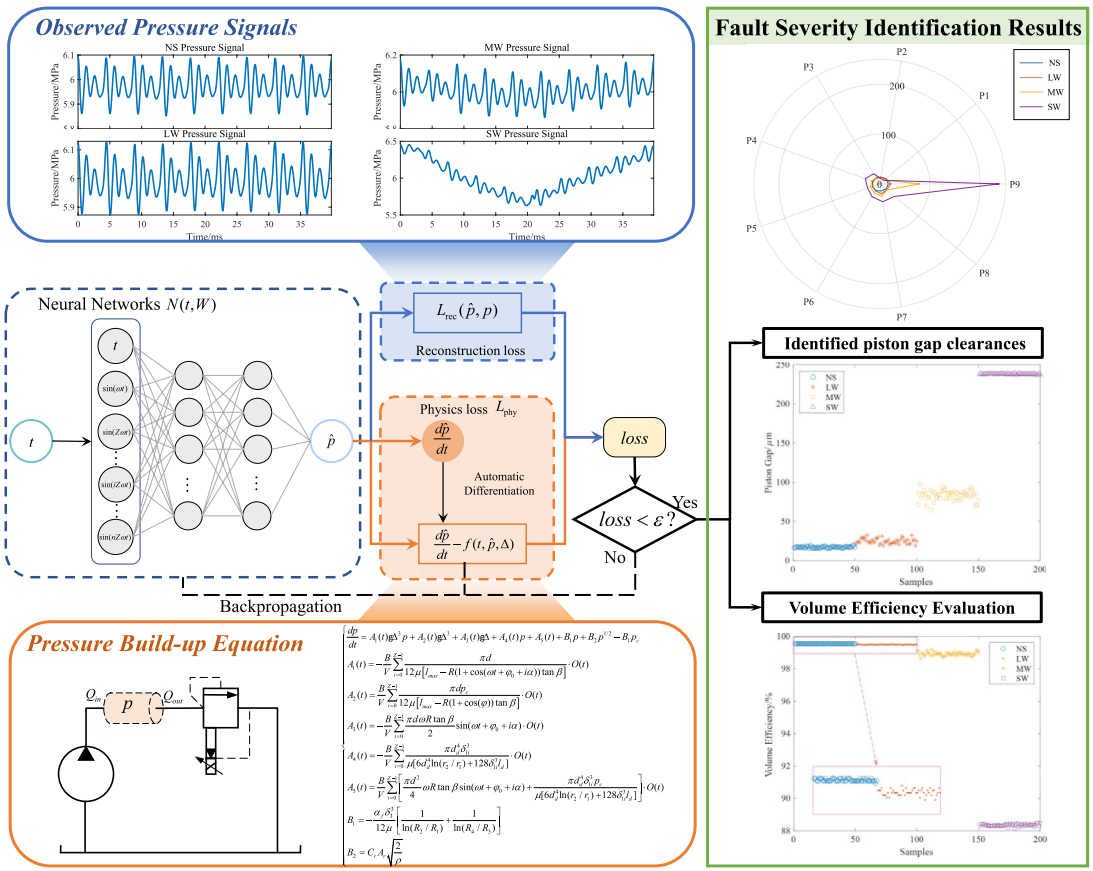


Figure 9 The overall framework of fault severity identification based on PINN [83]

4.2.2 Physics Knowledge Embedding Approach

In this case, physics knowledge is derived to certain loss functions and refines the learning process of neural networks, which is classified into the physics informed loss function case as reviewed previously.

The detailed approach is based on a PINN model which is composed of neural network model for data science and physics equations for theoretical science. The neural network model is involved to estimate the natural functional mapping from time coordinates to pressure amplitude. As for the physics knowledge, it is embedded in the optimization process of neural networks by design corresponding loss functions. In order to obtain the loss function, automatic differentiation method is firstly applied to the neural works, evaluating the differentiations of estimated pressure. Then, the loss function can be parameterized by the network outputs, which bridges the physics equations and neural networks. This function is formulated as the residual of the former mentioned physics equation, which contains fault severity information, describing the gap clearance of piston/cylinder block interface.

4.2.3 Experimental Results

To analysis the results of physics knowledge embedded learning approach, a fault simulation experiment is performed on an axial piston pump test bench. The severity of piston wear is classified into four degrees, including normal state, slight wear, medium wear, and severe wear. In this experiment, the pressure signal is collected at pump outlet by a high frequency pressure sensor with sampling frequency of 10240 Hz.

After optimizing the proposed model, the undetermined physics parameters inside physics informed loss function is identified. There are 50 signal samples fed into PINN model under each fault mode. The fault severity identification results are depicted in the results part of Figure 9, and the identified gap clearances are illustrated by a polar coordinate system, with the angular domain P1–P9 representing the 9 pistons inside the axial piston pump, and the radial axis representing the magnitude of gap clearance. Besides, the maximum identified clearances are scattered in a new figure. The proposed method is capability of distinguish the position of wearing pistons and the fault severities. Moreover, the identified results are

substituted to the physics equations and further used to calculate the physical performance degradation indicator, i.e., volume efficiency. The proposed model reveals the relation between the degradation stage of pump performance and piston wears, offering an interpretable approach for the fault severity identification of axial piston pumps.

4.3 Adversarial Algorithm Unrolling Network

4.3.1 Knowledge Source and Representation

Data collected in industrial scenarios often cover massive noise. Intelligent anomaly detection methods can obtain good performance in accuracy, but fail to ensure the credibility of the detection results. Sparse coding is a representation learning algorithm with explicit probabilistic inference formula. The form of its loss function is relevant to the noise distribution, which makes the denoising features well-explainable. Therefore, An et al. [115] construct such a signal processing informed learning framework, where knowledge source is the explainable sparse coding theory, and the knowledge can be represented as the solving algorithm of the multilayer sparse coding model.

4.3.2 Knowledge Embedding Approach

The informed learning framework is built upon adversarial autoencoder (AAE) and algorithm unrolling techniques. It consists of four steps. First, a sparse coding model is built based on data priors for encoding and decoding process. Second, the iterative solving algorithm is derived to determine the basic encoding/decoding units. After that, unrolling technique is applied to basic encoding/decoding units. The encoder, decoder and discriminator of AAE are modified to form the adversarial algorithm unrolling network. In such way, the knowledge contained in interpretable sparse coding approach is embedded into the reconstruction network. Finally, post-hoc interpretability analysis can be realized by visualizing the reconstructed and component features.

Table 2 Detection indicators on the SQI dataset

Algorithms	TPR	FPR	AUC	F1 score	ACC
VAE	71.77	**0.03**	99.71	83.52	92.26
AAE	96.73	1.86	98.75	95.91	97.74
GANomaly	31.62	0.05	97.56	48.00	81.29
AAU-Net	**96.82**	0.25	**99.81**	**98.04**	**98.94**

4.3.3 Experimental Results

To evaluate the detection performance and interpretability of the unrolling network, we carry out a fault experiment on the SQI dynamics simulator [115]. Four health states, i.e., normal, root crack, wear and missing teeth, are preset on the parallel shaft gearbox. The rotating frequency of input shaft is 30 Hz. The sampling frequency is 20480 Hz. Samples are truncated with a window length of 1024 and an overlap ratio of 0.8. Finally, 5780 normal samples are selected for training, and 5780 normal samples and 2173 abnormal samples are saved for testing. The proposed method is compared with other anomaly detection algorithms including VAE [129], AAE [130], and GANomaly [131], and their detection performances are listed in Table 2. The bold values in Table 2 represent the best result in the column.

As can be observed, AAU-Net obtained almost the best performance on five detection indicators, indicating that embedding correct knowledge into learning process can boost the model performance. With a decision threshold of 0.5, the rolling network achieves the accuracy of 98.94 and the F1 score of 98.04, greatly widening the gap between other intelligent detection

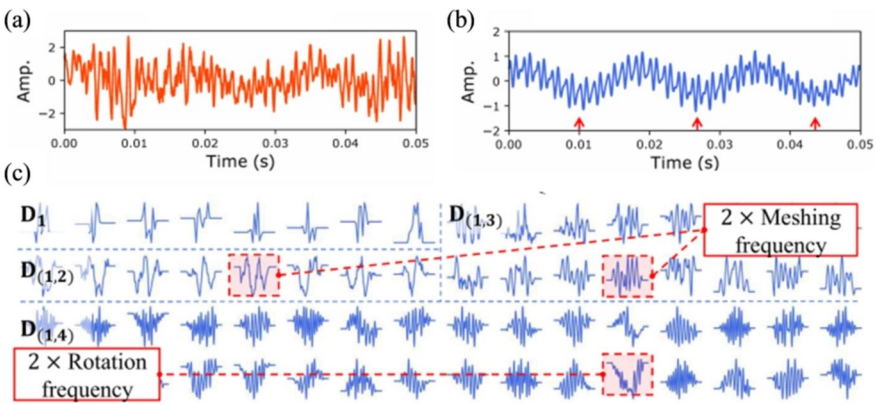


Figure 10 Visualization on the raw signal, reconstructed signals and component features [115]: (**a**) Raw signal, (**b**) The decoded features by adversarial algorithm rolling network, (**c**) Component features of the learned dictionary

methods. More importantly, the visualization on the reconstructed features and component features can indicate related frequency information of the system, as shown in Figure 10. According to the rotating frequency and the structural parameters of the gearbox, the meshing frequency between the planetary gear and the ring gear is computed to be 500 Hz. In Figure 10(a), system-related features are submerged by noises and hard to be detected, while in Figure 10(b), the decoded signal shows clear 2× rotating frequency and 2× meshing frequency, where the former is pointed with red arrows. Besides, the atoms ($\mathbf{D_1}$ to $\mathbf{D_{(1,4)}}$) of the learned dictionaries are listed in Figure 10(c), showing the component features at different scales. Among them, the components of rotating frequency and meshing frequency can be found. It indicates that the proposed model can extract system dynamics-related features from normal signals, helping users to better understand the model and improving the credibility of the results.

4.4 SPINN: Denoising Fault-Aware Wavelet Network

Here, a SPINN model informed by wavelet denoising and kurtosis-based feature selection is provided [5].

4.4.1 Signal Processing Priors

The signal processing priors are derived from a universal fault diagnosis procedure of signal processing: filter-feature-decision. For the specifical representation, firstly, wavelet transform is used to provide nonstationary signal representation from the time-frequency domain. Then, in the feature extraction stage, the feature should be noise-robust and fault-related. Thus, wavelet hard threshold denoising is used, which can effectively remove noise from the signal. Finally, to obtain discriminative features for fault diagnosis, index-based feature selection is important. For wavelet coefficients, the energy of the coefficients can be used as the index. In addition, spectral kurtosis (SK) is also used to search the optimal band-pass filter for fault information extraction.

4.4.2 Signal Processing Informed Model Design

In this section, signal processing priors mentioned above are used as the design principles of the network module. The network design also follows the flow of filter-feature-decision.

In the stage of filter, since wavelet transform is similar to the definition of convolutional layers, embedding wavelet transform priors can be achieved by replacing the convolutional kernels with wavelet basis functions. The scale parameter is set to learnable, and the translation parameter is replaced by the stride parameter. Additionally, multiple wavelet bases are used in the fused wavelet convolutional layer to obtain different fault features. In the feature extraction stage, since the coefficients of the noise are zeroed in the hard threshold denoising function, data-driven denoising is realized by constructing a noise classification network. The coefficients classified as noise are zeroed. Then the energy of wavelet coefficients is used for generating a data-driven new index for feature selection. Furthermore, an SK-based optimization loss is used for better wavelet kernel optimization. Finally, in the decision stage, a decision layer composed of a general convolutional network is incorporated. The

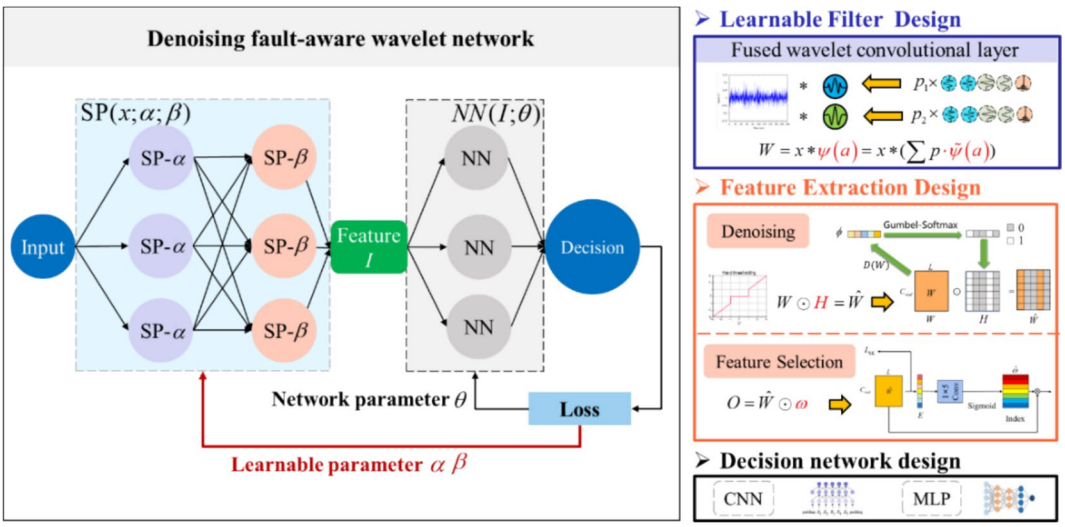


Figure 11 SPINN: denoising fault-aware wavelet network [5]

Table 3 Diagnosis performance of different models

Model	Max-acc (%)	Min-acc (%)	Avg-acc (%)
CNN	76.42	74.67	75.55 ± 0.55
WCNN	89.52	88.77	88.47 ± 0.62
MKCNN	87.77	86.03	86.72 ± 0.49
DSN	83.84	80.79	82.27 ± 0.79
SincNet	90.39	89.52	89.74 ± 0.29
DFAWNet	92.36	93.32	92.76 ± 0.49

overall design of signal processing informed denoising fault-aware wavelet network (DFAWNet) is shown in Figure 11.

4.4.3 Experimental Results

Firstly, different methods are validated on the Machinery Failure Prevention Technology dataset [132]. The results shown in Table 3 indicate that DFAWNet performs best among shallow convolution neural network (CNN) [133], an anti-noise model with a wide kernel size of 64 in the first layer (WCNN) [134], a multiscale kernel model with multi-resolution property (MKCNN) [135], a model with residual shrinkage module (RSNet) [102], an explainable model with a learnable sinc function as the filter (SincNet) [136].

For interpretability, this network is analyzed from a frequency perspective on the Machinery Failure Prevention Technology dataset. As shown in Figure 12, for the normal and fault signals, the frequencies of optimal filters are between 10 kHz and 14 kHz. With data-driven training, the frequency of wavelet kernels shifts from 6.10 kHz to 12.21 kHz, positioning it in the middle of 10 kHz and 14 kHz. This experiment verified the adaptive fault feature extraction ability of DFAWNet.

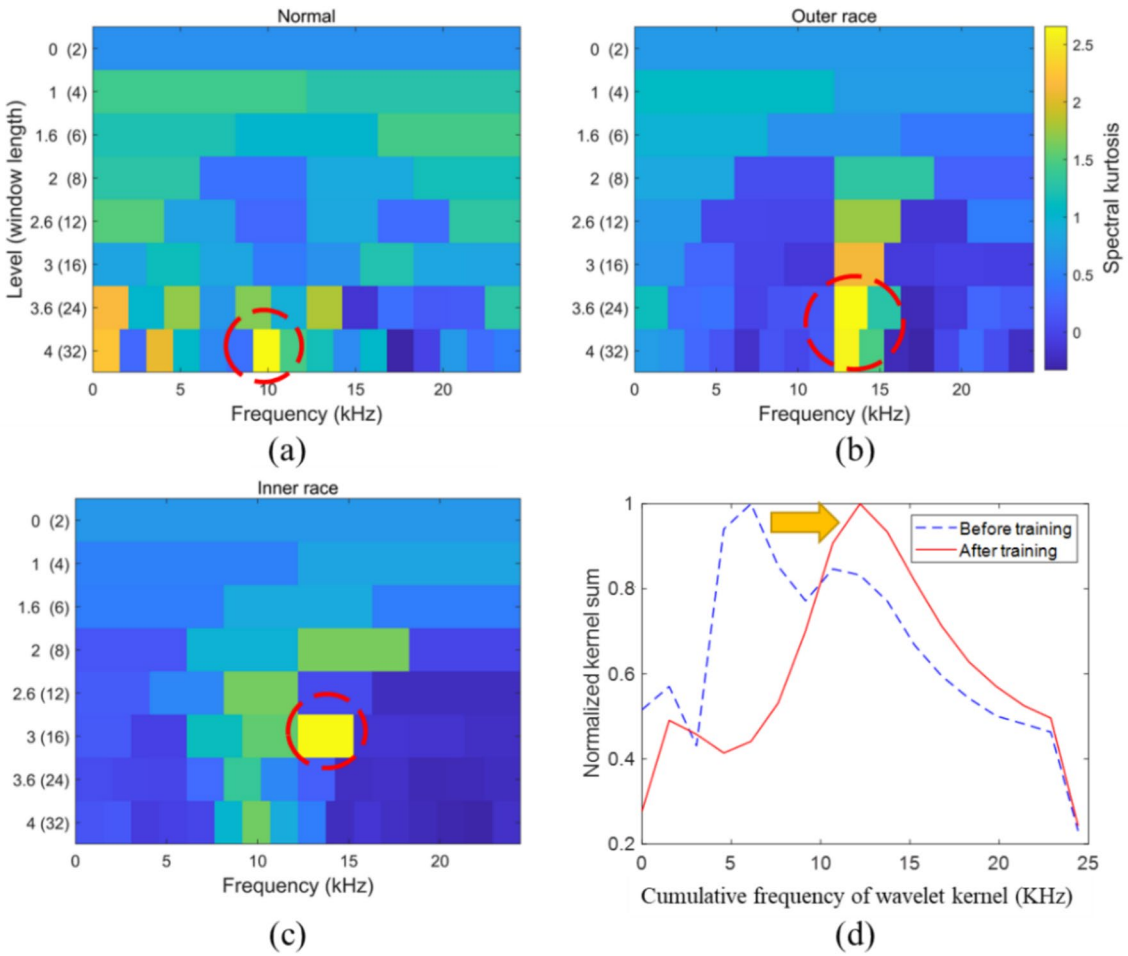


Figure 12 Fault characteristic frequency bands and frequency change of wavelet kernels [5]: (**a**) Normal, (**b**) Inner race fault, (**c**) Outer race fault, (**d**) Frequency change of wavelet kernels after training

4.5 Physics-Informed Machine Learning for Pressure Sensor Placement Optimization

In this subsection, a physics-informed machine learning approach for surrogate modeling of wind pressure and optimization of pressure sensor placement is illustrated [137], which shows potential in structural health monitoring for civil structures.

4.5.1 Knowledge Source and Representation

The knowledge prior of this work is derived from conservation laws and represented by partial differential equations. To optimize the pressure sensor placement in a civil structure subjected to different wind conditions, this work developed a surrogate model to predict the full-field pressure from scattered sensor measurements. To construct the surrogate model, physical knowledge informed by a turbulence model is introduced. Such knowledge prior makes the establishment of surrogate model without the requirement of a large amount of measurement data owing to the highly condensed knowledge embedded in the physical principles.

4.5.2 Knowledge Embedding Approach

To construct the surrogate model to predict the full-field wind pressure, a turbulence model represented by partial differential equations is used to design a loss function to regular the optimization process of neural network. Therefore, this knowledge driven approach can be classified into the physics informed optimization regularization.

The objective of this approach is two-fold, as shown in Figure 13(a). The first is to use a physics-informed neural network to construct the surrogate model for wind pressure prediction. In this step, the physical equations will be utilized to design a loss function to train neural network, so this surrogate model can recover the full-field pressure profile from scattered measurement data with the help of highly condensed physical knowledge. The second is to use the above physics-informed surrogate model as a fast evaluation predictor to learn the best placement for a given number of pressure sensors. In the second step, the problem is stated as an optimization problem to achieve the best predictive accuracy over a wide range of wind conditions.

4.5.3 Experimental Results

To evaluate performance of the proposed method, a finite element method (FEM) is used to generate synthetic data of a classical flat roof. As the optimization problem of sensor placement has to decide both the number and location of sensors, this work first investigates the effort of the number of sensors for one specific wind condition and then optimize their locations for various conditions. The results can be found in Figure 13(b, c). It can be observed that the proposed method can predict a more accurate full-field pressure and get a moderate number of sensors.

4.6 Physics-Informed Machine Learning for Data Augmentation in Battery Prognostics

In this subsection, a physics-informed machine learning model for battery state of health prognostics [138] is introduced. In this case, a physical mode is used to

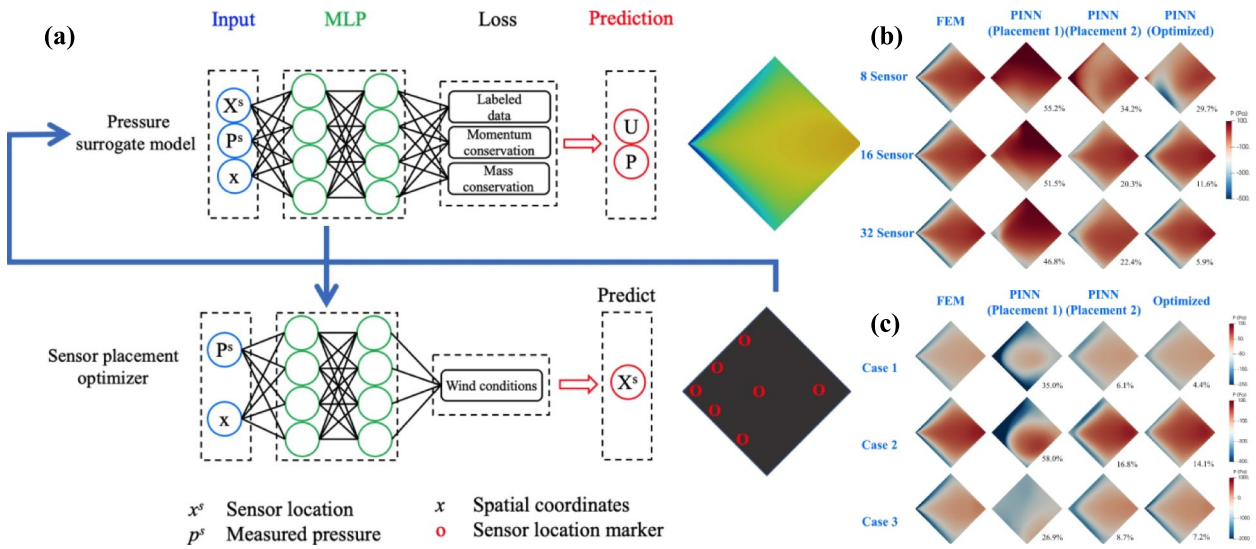


Figure 13 PINN for optimization of pressure sensor placement [137]: (**a**) Overview of PINN model, (**b**) Optimization results of different number of sensors, (**c**) Optimization results under different wind conditions

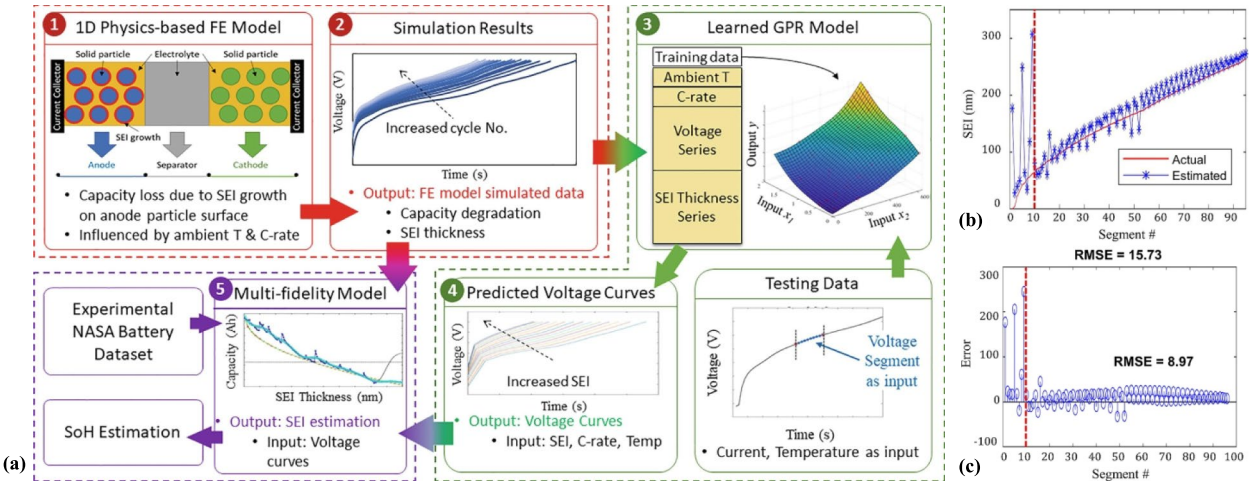


Figure 14 PIML for battery prognostics [138]: (**a**) Framework of PIML method for battery prognostics, (**b**) The estimated result of the health state of battery, (**c**) The estimated error with respect to segments

generate synthetic data and then experimental data will be fused to develop a multi-fidelity framework for battery prognostics.

4.6.1 Knowledge Source and Representation

Knowledge prior of this work depends on the physical model of the influences of dominating aging mode. More specifically, this knowledge is represented by a physics-based finite element model to describe the relation between solid electrolyte interface growth with anode particle surfaces and capacity loss of battery.

4.6.2 Knowledge Embedding Approach

To establish a multi-fidelity framework, the physics-based finite element model is used to generate simulated data. Then the simulated data is combined with experimental data to train a gaussian process regression model. In this case, physical knowledge is embedded into the data acquisition process in ML pipeline. The entire framework is shown in Figure 14(a).

4.6.3 Experimental Results

Numerical results are presented to evaluate the performance of the proposed method, as shown in Figure 14(b, c). It can be observed that the proposed model can perform well at later cycles of battery health, while remains a high error in the early cycles. Such physics-informed data augmentation method allows to train a predictive model with few-shot realistic samples.

5 Challenges, Potential Applications and Recommendations

The insight of knowledge driven machine learning is not a recent objective in PHM. A historical effort is to employ various L-norm regularization methods to constrain model to have some specific properties, like robustness to measurement noise. An additional and parallel way is hybrid methods in PHM, such as integrating physics model with data-driven model or directly combining multiple data-driven models. These approaches have gained a lot of attention to improve prediction accuracy and developed for a long time. On the other hand, with the quick development of Deep Learning since 2012, many researches shifted to novel neural network structures or learning algorithms. Less attention was focused on knowledge driven, especially for physics model. The recent PINN work in 2017 [139] revived awareness of additional benefits that such knowledge driven paradigm can bring, such as generalization ability and interpretability, and thus exploded hot spots of this topic in PHM. Moreover, the desire for trustworthy AI in PHM domain also requires such evolution. However, there still exists challenges in knowledge driven machine learning.

5.1 Challenges

5.1.1 The Tradeoff between Fidelity and Complexity in Physical Models

In modeling the same physical system, there exist multiple methods that can be used to construct models with different fidelities. As the fidelity of the model increases, the accuracy of the obtained solutions to the physical model increases accordingly, thus enabling better embedding of knowledge into the network. However,

high fidelity models are often accompanied by increased solution complexity, which is not advantageous for network design and training for learning. On the other hand, physical systems in the real world are becoming more and more complex, causing difficulty to build high-fidelity physical models, and oversimplification of the physical systems can lead to model distortion. Therefore, how to balance the fidelity and complexity of the physical model is very important for the design of physical informed neural networks.

5.1.2 Quantification of Interpretability

Deep learning models have extremely complex structures and parameters. In order to explain the internal mechanisms of these models, metrics need to be developed to assess the amount and quality of interpretable information contained in the network structure and output. This further brings the issue of quantifying interpretability. The quantitative data provided by such metrics can also further help us to understand and optimize the models, to achieve trade-offs between accuracy and transparency, and to provide reliable explanations of the predicted results. It is through quantifying interpretability that we can strike a balance between conflicting metrics and further advance the application of deep learning in the field of intelligent diagnosis to achieve more reliable and understandable diagnostic results.

5.1.3 Iterative "Human in the Loop"

Interpretable neural network can be regarded as an offline human-in-the-loop strategy by embedding the existing expert experience and knowledge into the network, and the output of the network is fed back to the expert for final interpretation and decision making. However, how to design iterative "human-in-the-loop" intelligent systems so that they can most effectively collaborate with human experts and ensure that decisions and feedback from both sides can be accurately aligned is a major challenge. This further requires the development of intuitive and easy-to-use user interfaces so that human experts can easily interact with intelligent networks. In addition, consideration needs to be given in protecting sensitive data generated during human-computer interactions to prevent data leakage or misuse.

5.1.4 Modularization and Efficient Collaboration of Physical Knowledge

In the field of interpretable intelligent diagnosis, the applied expert experience and knowledge is complex and diverse. Effectively integrating this information into a learning system is a challenging task. One possible idea is to modularize the physical knowledge separately. The modularization approach allows different domains and levels of physics knowledge to be decomposed into smaller modules, each focusing on a specific task. This provides flexibility and scalability in embedding physical knowledge and enables knowledge sharing and reuse. Suitable knowledge modules can be appropriately selected for different tasks and problems. The reorganization and synergy of module knowledge increases accuracy while further enhancing the interpretability of the learning system and understanding the decision-making process of the system as each module is inherently understandable.

5.2 Potential Applications

5.2.1 Interpretable AI towards Trustworthy Decision

The ultimate goal of PHM is to ensure human security and reduce maintenance costs, so this domain is risk sensitive to wrong decision. This is why advanced data-driven models have been slow to spread in this conservative domain. Despite pure data-driven models can provide state-of-the-art performance, the loss caused by making wrong prediction even once is unacceptable in PHM domain. Therefore, how to enhance transparency of data-driven model is a capital step to gain the trust of decision makers. Knowledge driven machine learning provides an active strategy to intervene ML pipeline, and thus convert its prediction logic to an interpretable level that humans can understand.

5.2.2 Active Optimization for Machine Learning Pipeline

Through embedding knowledge into ML pipeline, KDML is expected to learn intrinsic pattern from data which can be generalized to different data domains. For example, the construction of physical models is based on the first principle, like energy conservation law for thermodynamics systems or momentum conservation law for mechanical systems. In this case, physical models can describe the system dynamics under different working conditions or initial conditions. Therefore, knowledge is expected to be broadly adaptable to these varying conditions. Moreover, this active embedding is a potential way to troubleshoot invalidation in ML pipeline and then to optimize it. With respect to the three main parts in ML pipeline, data, model, and optimization, knowledge can guide their construction before forming ML pipeline. For example, an important but seldom noticed issue is data acquisition. Structure knowledge of the physics system can help to optimize sensor layout to monitor principal components. Knowledge of specific fault types can help to optimize parameter setting of sensors, like suitable sampling frequency to capture fault features under limited data storage. Therefore, knowledge still has additional benefits to optimize ML pipeline, except for generalization ability improvement.

5.2.3 Knowledge Discovery to Feedback Smart Manufacturing

Machine learning methods, especially neural networks, are transforming from simple data processing tools to complete knowledge discovery frameworks. Outcome of the ML pipeline is an operational model, which can learn pattern and relation from finite data samples. Based on the learned pattern or relation, further investigation can be performed to find causal mechanism between physics variable, then to construct interpretable theory or hypothesis. In PHM domain, knowledge can be discovered from operation and maintenance data to feedback product design or manufacturing. For example, ML methods can build a causal graph from monitoring data to represent relation within system variables, and then vulnerable spot of a complex system can be located through inference of causal graph. This potential application also points out the importance of interpretability, that the learned pattern or relation should be totally comprehensible in theory.

5.3 Usage Recommendations for Different PHM Problems

In this subsection, we will discuss how to select the appropriate knowledge driven methods for specific industrial scenarios for PHM domain, including anomaly detection, fault diagnosis, and fault prognosis.

5.3.1 Anomaly Detection

Anomaly detection is understood to identify abnormal data whose dynamic behavior deviates significantly from healthy states. Through the literature analysis of current research as shown in Figure 15, it can be observed that there are mainly two roadmaps to implement KDML for anomaly detection, i.e., integrating statistic property for optimization regularization and integrating signal equation for model design, which can be a recommendation for potential usage.

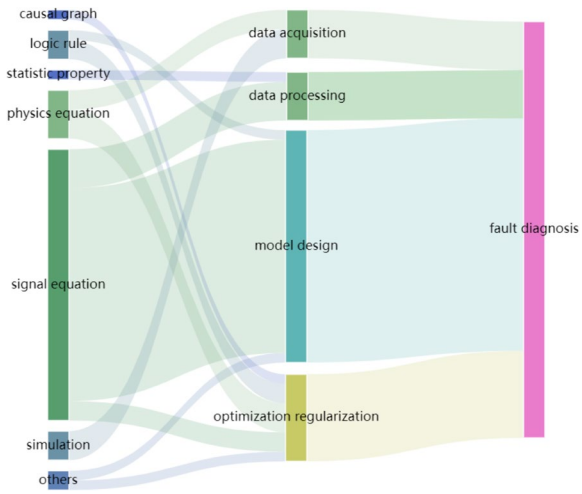


Figure 16 Sankey diagram for KDML in fault diagnosis

5.3.2 Fault Diagnosis

Fault diagnosis is used to identify fault types or fault locations with diverse dynamics behaviors. As shown in Figure 16, we can find that signal equation is the majority among knowledge representations and has been embedded in several parts of ML pipeline, including data processing, model design, and optimization regularization. As signal processing has been a well-researched technique for fault diagnosis over the past decades and has different representations for different physical systems,

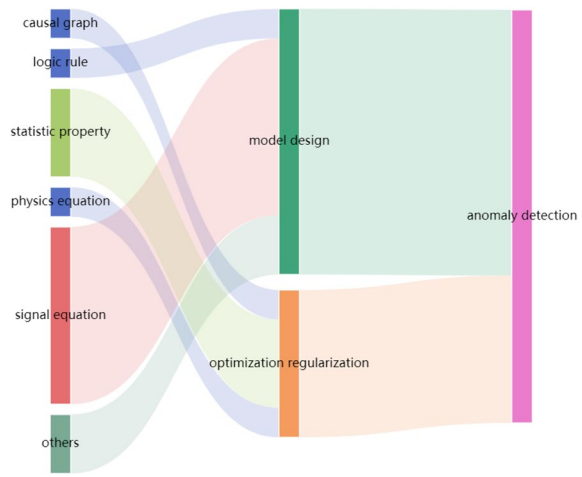


Figure 15 Sankey diagram for KDML in anomaly detection

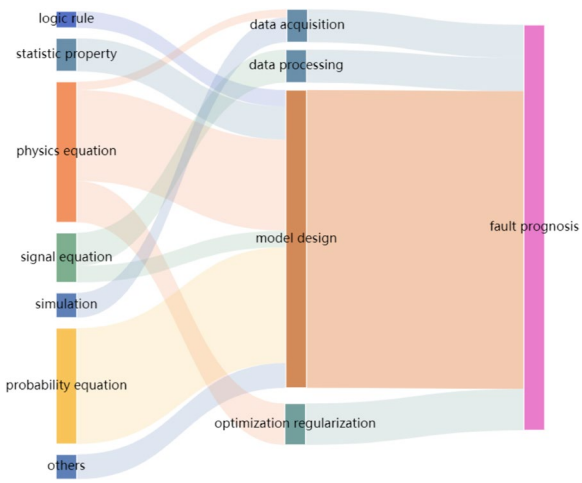


Figure 17 Sankey diagram for KDML in fault prognosis

a recommended roadmap is to utilize signal equation to design a specific ML model for fault diagnosis.

5.3.3 Fault Prognosis

Fault prognosis focuses on predicting the time at which a physical system will not perform its function. As shown in Figure 17, we can observe that physics equation and probability equation are the two main kinds of knowledge representation methods, such as cumulative fatigue models or stochastic degradation processes. For the knowledge embedding approaches, model design is the main part to be modified for KDML. It is recommended to use physical knowledge for KDML to provide interpretable solutions for fault prognosis.

6 Conclusions

In this paper, we proposed a universal concept, knowledge driven machine learning, for integrating diverse knowledge into machine learning pipeline in PHM domain. Our main contribution is to classify KDML in a hierarchical framework, ranging from knowledge sources, knowledge representation, to knowledge embedding in ML pipeline. In addition, we provided several case studies to illustrate usage of different KDML methods in PHM domain. The proposed hierarchical framework of KDML and extensive case studies can help PHM users to find suitable way for their applications.

In addition, popularity of knowledge driven machine learning indicates a new research paradigm in science and engineering. In the era of big data, development of data science is unstoppable. Some even refer to the rise of data science as "the end of theory" [140], as data driven model can easily generate an operable model from big data. But knowledge driven machine learning provides a new perspective to evaluate the effect brought by data science, that knowledge, even theory, can be extracted from data. It means KDML starts a new direction to find theory rather than its end.

Acknowledgements
Not applicable.

Authors' Contributions
RY: Review-editing & supervision; ZZ: Writing-review & editing; ZS: Writing-review & editing; ZW: Writing-review & editing; CH: Writing-review & editing; YL: Writing-review & editing; YY: Writing-review & editing; XC: Review-editing & supervision; RG: Review-editing & supervision. All authors read and approved the final manuscript.

Funding
Supported in part by Science Center for Gas Turbine Project (Project No. P2022-DC-I-003-001) and National Natural Science Foundation of China (Grant No. 52275130).

Availability of Data and Materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing Interests
The authors declare no competing financial interests.

Received: 7 June 2024 Revised: 10 October 2024 Accepted: 23 December 2024
Published online: 17 January 2025

References

[1] A Dourado, F A C Viana. Physics-informed neural networks for missing physics estimation in cumulative damage models: A case study in corrosion fatigue. *J Comput Inf Sci Eng*, 2020, 20:061007.
[2] J Luo, M Namburu, K Pattipati, et al. Model-based prognostic techniques. *Proceedings AUTOTESTCON 2003, IEEE Systems Readiness Technology Conference*, IEEE, 2003:330-340.
[3] B Li. A review of tool wear estimation using theoretical analysis and numerical simulation technologies. *International Journal of Refractory Metals and Hard Materials*, 2012, 35:143-151.
[4] M A Chao. *Combining deep learning and physics-based performance models for diagnostics and prognostics*. ETH Zurich, 2021.
[5] Z Shang, Z Zhao, R Yan. Denoising fault-aware wavelet network: A signal processing informed neural network for fault diagnosis. *Chin J Mech Eng*, 2023, 36:9.
[6] Z Wei, Z Zhao, Z Zhou, et al. Collaborative-sequential optimization for aero-engine maintenance based on multi-agent reinforcement learning. *Expert Syst Appl*, 2024, 247:123358.
[7] G Ma, S Xu, C Cheng. Fault detection of lithium-ion battery packs with a graph-based method. *J Energy Storage*, 2021, 43:103209.
[8] J Zhang, C Liu, R X Gao. Physics-guided Gaussian process for HVAC system performance prognosis. *Mech Syst Signal Process*, 2022, 179:109336.
[9] Z Zhang, Y Liu. Parsimony-enhanced sparse Bayesian learning for robust discovery of partial differential equations. *Mech Syst Signal Process*, 2022, 171:108833.
[10] L Cui, X Wang, H Wang, et al. Remaining useful life prediction of rolling element bearings based on simulated performance degradation dictionary. *Mech Mach Theory*, 2020, 153:103967.
[11] G E Karniadakis, I G Kevrekidis, L Lu, et al. Physics-informed machine learning. *Nat Rev Phys*, 2021, 3: 422-440.
[12] A Karpatne, G Atluri, J H Faghmous, et al. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Trans Knowl Data Eng*, 2017, 29: 2318-2331.
[13] N Sharma, Y A Liu. A hybrid science-guided machine learning approach for modeling chemical processes: A review. *AIChE J*, 2022, 68:e17609.
[14] W Deng, K T P Nguyen, K Medjaher, et al. Physics-informed machine learning in prognostics and health management: State of the art and challenges. *Appl Math Model*, 2023, 124: 325-352.
[15] Y Xu, S Kohtz, J Boakye J, et al. Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges. *Reliab Eng Syst Saf*, 2023, 230:108900.
[16] S P Zhu, L Wang, C Luo, et al. Physics-informed machine learning and its structural integrity applications: state of the art. *Philos Trans R Soc Math Phys Eng Sci*, 2023, 381:20220406.
[17] L Rueden, S Mayer, K Beckh, et al. Informed machine learning – A taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Trans Knowl Data Eng*, 2023, 35: 614-633.
[18] D Li, K Deng, M Zhao, et al. Knowledge-driven machine learning: Concept, model and case study on channel estimation. arXiv preprint arXiv:2012.11178, 2020.
[19] S Vollert, M Atzmueller, A Theissler. Interpretable machine learning: A brief survey from the predictive maintenance perspective. *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation* (ETFA), IEEE, 2021: 01-08.
[20] G Chen, J Yuan, Y Zhang, et al. Enhancing reliability through interpretability: A comprehensive survey of interpretable intelligent fault diagnosis in rotating machinery. *IEEE Access*, 2024.

[21] B Saha, K Goebel. Modeling li-ion battery capacity depletion in a particle filtering framework. *Proceedings of the Annual Conference of the PHM Society*, 2009.

[22] S Vollert, M Atzmueller, A Theissler. Interpretable machine learning: A brief survey from the predictive maintenance perspective. *2021 26th IEEE Int. Conf. Emerg. Technol. Fact. Autom*. ETFA, Vasteras, Sweden: IEEE, 2021: 1-8.

[23] Y Qian, R Yan, R X Gao. A multi-time scale approach to remaining useful life prediction in rolling bearing. *Mech Syst Signal Process*, 2017, 83: 549-567.

[24] C Ordóñez, F S Lasheras, J Roca-Pardiñas, et al. A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines. *J Comput Appl Math*, 2019, 346.

[25] P Kundu, S Chopra, B K Lad. Multiple failure behaviors identification and remaining useful life prediction of ball bearings. *J Intell Manuf*, 2019, 30: 1795-1807.

[26] P Kundu, A K Darpe, M S Kulkarni. Weibull accelerated failure time regression model for remaining useful life prediction of bearing working under multiple operating conditions. *Mech Syst Signal Process*, 2019, 134:106302.

[27] S M M Hassani N, J Jin, J Ni. Physics-based Gaussian process for the health monitoring for a rolling bearing. *Acta Astronaut*, 2019, 154: 133-139.

[28] S Zhang, Q Zhai, Y Li. Degradation modeling and RUL prediction with Wiener process considering measurable and unobservable external impacts. *Reliab Eng Syst Saf*, 2023, 231:109021.

[29] W Li, T Liu. Time varying and condition adaptive hidden Markov model for tool wear state estimation and remaining useful life prediction in micro-milling. *Mech Syst Signal Process*, 2019, 131: 689-702.

[30] G Lyu, H Zhang, Y Zhang, et al. An interpretable remaining useful life prediction scheme of lithium-ion battery considering capacity regeneration. *Microelectron Reliab*, 2022, 138:114625.

[31] L Cui, X Wang, H Wang, et al. Research on remaining useful life prediction of rolling element bearings based on time-varying Kalman filter. *IEEE Trans Instrum Meas*, 2020, 69: 2858-2867.

[32] X Zheng, H Wu, Y Chen. Remaining useful life prediction of lithium-ion battery using a hybrid model-based filtering and data-driven approach. *2017 11th Asian Control Conf.* ASCC, Gold Coast, QLD: IEEE, 2017: 2698-2703.

[33] H Pei, X S Si, C H Hu, et al. An adaptive prognostics method for fusing CDBN and diffusion process: Application to bearing data. *Neurocomputing*, 2021, 421: 303-315.

[34] Y Li, Z Zhou, C Sun, et al. Life-cycle modeling driven by coupling competition degradation for remaining useful life prediction. *Reliab Eng Syst Saf*, 2023, 238:109480.

[35] Y Deng, A D Bucchianico, M Pechenizkiy. Controlling the accuracy and uncertainty trade-off in RUL prediction with a surrogate Wiener propagation model. *Reliab Eng Syst Saf*, 2020, 196:106727.

[36] C H Hu, H Pei, X S Si, et al. A prognostic model based on DBN and diffusion process for degrading bearing. *IEEE Trans Ind Electron*, 2020, 67: 8767-8777.

[37] H Sun, D Cao, Z Zhao, et al. A hybrid approach to cutting tool remaining useful life prediction based on the Wiener process. *IEEE Trans Reliab*, 2018, 67: 1294-1303.

[38] L Dai, J Guo, J L Wan, et al. A reliability evaluation model of rolling bearings based on WKN-BiGRU and Wiener process. *Reliab Eng Syst Saf*, 2022, 225:108646.

[39] Z Chen, T Xia, Y Li, et al. A hybrid prognostic method based on gated recurrent unit network and an adaptive Wiener process model considering measurement errors. *Mech Syst Signal Process*, 2021, 158:107785.

[40] Z Zhou, T Li, Z Zhao, et al. Time-varying trajectory modeling via dynamic governing network for remaining useful life prediction. *Mech Syst Signal Process*, 2023, 182:109610.

[41] T Yan, D Wang, T Xia, et al. Online piecewise convex-optimization interpretable weight learning for machine life cycle performance assessment. *IEEE Trans Neural Netw Learn Syst*, 2024: 1-13.

[42] T Yan, D Wang, M Zheng, et al. Interpretable sparse learned weights and their entropy based quantification for online machine health monitoring. *Mech Syst Signal Process*, 2023, 199:110493.

[43] T Zhou, E L Droguett, A Mosleh. Physics-informed deep learning: A promising technique for system reliability assessment. *Appl Soft Comput*, 2022, 126:109217.

[44] Y Li, Z Zhou, C Sun, et al. Variational attention-based interpretable transformer network for rotary machine fault diagnosis. *IEEE Trans Neural Netw Learn Syst*, 2022: 1-14.

[45] X Bi, J Zhao. A novel orthogonal self-attentive variational autoencoder method for interpretable chemical process fault detection and identification. *Process Saf Environ Prot*, 2021, 156: 581-597.

[46] X Liao, S Chen, P Wen, et al. Remaining useful life with self-attention assisted physics-informed neural network. *Adv Eng Inform*, 2023, 58:102195.

[47] S Cofre-Martel, E Lopez Droguett, M Modarres. Remaining useful life estimation through deep learning partial differential equation models: A framework for degradation dynamics interpretation using latent variables. *Shock Vib*, 2021, 2021: 1-15.

[48] J Li, Y Wang, Y Zi, et al. Causal disentanglement: A generalized bearing fault diagnostic framework in continuous degradation mode. *IEEE Trans Neural Netw Learn Syst*, 2023, 34: 6250-6262.

[49] C Hu, J Wu, C Sun, et al. Mutual information-based feature disentangled network for anomaly detection under variable working conditions. *Mech Syst Signal Process*, 2023, 204:110804.

[50] S Y Wong, K S Yap, H J Yap, et al. On equivalence of FIS and ELM for interpretable rule-based knowledge representation. *IEEE Trans Neural Netw Learn Syst*, 2015, 26: 1417-1430.

[51] J Yu, G Liu. Knowledge extraction and insertion to deep belief network for gearbox fault diagnosis. *Knowl-Based Syst*, 2020, 197:105883.

[52] Z Wu, H Luo, Y Yang, et al. K-PdM: KPI-oriented machinery deterioration estimation framework for predictive maintenance using cluster-based hidden Markov model. *IEEE Access*, 2018, 6: 41676-87.

[53] B Steenwinckel, D D Paepe, S Vanden Hautte, et al. FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning. *Future Gener Comput Syst*, 2021, 116: 30-48.

[54] Z Zhou, Z Ming, J Wang, et al. A novel belief rule-based fault diagnosis method with interpretability. *Comput Model Eng Sci*, 2023, 136: 1165-1185.

[55] Z Ming, Z Zhou, Y Cao, et al. A new interpretable fault diagnosis method based on belief rule base and probability table. *Chin J Aeronaut*, 2023, 36: 184-201.

[56] J Luo, M Namburu, K Pattipati, et al. Model-based prognostic techniques [maintenance applications]. *Proc. AUTOTESTCON 2003 IEEE Syst. Readiness Technol. Conf.*, 2003: 330-340.

[57] A Mojallal, S Lotfifard. Multi-physics graphical model-based fault detection and isolation in wind turbines. *IEEE Trans Smart Grid*, 2018, 9: 5599-5612.

[58] K Yu, Q Fu, H Ma, et al. Simulation data driven weakly supervised adversarial domain adaptation approach for intelligent cross-machine fault diagnosis. *Struct Health Monit*, 2021, 20: 2182-2198.

[59] Q Ni, J C Ji, B Halkon, et al. Physics-informed residual network (PIResNet) for rolling element bearing fault diagnostics. *Mech Syst Signal Process*, 2023, 200:110544.

[60] S Kohtz, Y Xu, Z Zheng, et al. Physics-informed machine learning model for battery state of health prognostics using partial charging segments. *Mech Syst Signal Process*, 2022, 172:109002.

[61] R He, Y Dai, J Lu, et al. Developing ladder network for intelligent evaluation system: Case of remaining useful life prediction for centrifugal pumps. *Reliab Eng Syst Saf*, 2018, 180: 385-393.

[62] D Agarwal, M Kumari, B Srinivasan, et al. Fault diagnosis and degradation analysis of PMDC motors using FEA based models. *2020 IEEE Int. Conf. Power Electron. Smart Grid Renew. Energy PESGRE2020*, Cochin, India: IEEE, 2020: 1-6.

[63] F Zhao, Z Tian, Y Zeng. Uncertainty quantification in gear remaining useful life prediction through an integrated prognostics method. *IEEE Trans Reliab*, 2013, 62: 146-159.

[64] K López De Calle-Etxabe, C Ruiz-Cárcel, S Starr, et al. Hybrid modelling for linear actuator diagnosis in absence of faulty data records. *Comput Ind*, 2020, 123:103339.

[65] M A Djeziri, S Benmoussa, R Sanchez. Hybrid method for remaining useful life prediction in wind turbine systems. *Renew Energy*, 2018, 116: 173-187.

[66] R T Q Chen, Y Rubanova, J Bettencourt, et al. Neural ordinary differential equations. *Adv. Neural Inf. Process. Syst.*, 2018, 31.

[67] R G Nascimento, K Fricke, F A C Viana. A tutorial on solving ordinary differential equations using Python and hybrid physics-informed neural network. *Eng Appl Artif Intell*, 2020, 96:103996.

[68] R G Nascimento, F A C Viana. Cumulative damage modeling with recurrent neural networks. *AIAA J*, 2020, 58: 5459-5471.

[69] Y A Yucesan, F A C Viana. Hybrid physics-informed neural networks for main bearing fatigue prognosis with visual grease inspection. *Comput Ind*, 2021, 125:103386.

[70] Y A Yucesan, F A C Viana. A hybrid physics-informed neural network for main bearing fatigue prognosis under grease quality variation. *Mech Syst Signal Process*, 2022, 171:108875.

[71] Y A Yucesan, F A C Viana. A physics-informed neural network for wind turbine main bearing fatigue. *Int J Progn Health Manag*, 2023, 11.

[72] R Tipireddy, A Tartakovsky. Physics-informed machine learning method for forecasting and uncertainty quantification of partially observed and unobserved states in power grids. arXiv preprint arXiv:1806.10990, 2018.

[73] S Saemundsson, A Terenin, K Hofmann, et al. Variational integrator networks for physically structured embeddings. *Proc. Twenty Third Int. Conf. Artif. Intell. Stat.*, PMLR, 2020: 3078-3087.

[74] M A Chao, C Kulkarni, K Goebel, et al. Fusing physics-based and deep learning models for prognostics. *Reliab Eng Syst Saf*, 2022, 217:107961.

[75] R G Nascimento, M Corbetta, C S Kulkarni, et al. Hybrid physics-informed neural networks for lithium-ion battery modeling and prognosis. *J Power Sources*, 2021, 513:230526.

[76] M Lutter, C Ritter, J Peters. Deep Lagrangian networks: Using physics as model prior for deep learning. arXiv preprint arXiv:1907.04490, 2019.

[77] R Zhang, Y Liu, H Sun. Physics-informed multi-LSTM networks for metamodeling of nonlinear structures. *Comput Methods Appl Mech Eng*, 2020, 369:113226.

[78] R Zhang, Y Liu, H Sun. Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. *Eng Struct*, 2020, 215:110704.

[79] X Chen, M Ma, Z Zhao, et al. Physics-informed deep neural network for bearing prognosis with multisensory signals. *J Dyn Monit Diagn*, 2022: 200-207.

[80] B Freeman, Y Tang, Y Huang, et al. Physics-informed turbulence intensity infusion: A new hybrid approach for marine current turbine rotor blade fault detection. *Ocean Eng*, 2022, 254:111299.

[81] S Shen, H Lu, M Sadoughi, et al. A physics-informed deep learning approach for bearing fault detection. *Eng Appl Artif Intell*, 2021, 103:104295.

[82] W Xu, Z Zhou, T Li, et al. Physics-constraint variational neural network for wear state assessment of external gear pump. *IEEE Trans Neural Netw Learn Syst*, 2022: 1–11.

[83] Z Wang, Z Zhou, W Xu, et al. Physics informed neural networks for fault severity identification of axial piston pumps. *J Manuf Syst*, 2023, 71: 421-437.

[84] L E Bouny, M Khalil, A Adib. An end-to-end multi-level wavelet convolutional neural networks for heart diseases diagnosis. *Neurocomputing*, 2020, 417: 187-201.

[85] L Wen, L Gao, X Li, et al. A jointed signal analysis and convolutional neural network method for fault diagnosis. *Procedia CIRP*, 2018, 72: 1084-1087.

[86] X Li, W Zhang, Q Ding. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab Eng Syst Saf*, 2019, 182: 208-218.

[87] S Fu, L Lin, Y Wang, et al. High imbalance fault diagnosis of aviation hydraulic pump based on data augmentation via local wavelet similarity fusion. *Mech Syst Signal Process*, 2024, 209:111115.

[88] D K B Kulevome, H Wang, B M Cobbinah, et al. Effective time-series data augmentation with analytic wavelets for bearing fault diagnosis. *Expert Syst Appl*, 2024, 249:123536.

[89] Y Meng, C Shao. Physics-informed ensemble learning for online joint strength prediction in ultrasonic metal welding. *Mech Syst Signal Process*, 2022, 181:109473.

[90] R Zhao, D Wang, R Yan, et al. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Trans Ind Electron*, 2018, 65: 1539-1548.

[91] J Zhu, N Chen, W Peng. Estimation of bearing remaining useful life based on multiscale convolutional neural network. *IEEE Trans Ind Electron*, 2019, 66: 3208-3216.

[92] L Ren, J Cui, Y Sun, et al. Multi-bearing remaining useful life collaborative prediction: A deep learning approach. *J Manuf Syst*, 2017, 43: 248-256.

[93] M Sadoughi, C Hu. Physics-based convolutional neural network for fault diagnosis of rolling element bearings. *IEEE Sens J*, 2019, 19: 4181-4192.

[94] Y Jiang, T Xia, D Wang, et al. Spatiotemporal denoising wavelet network for infrared thermography-based machine prognostics integrating ensemble uncertainty. *Mech Syst Signal Process*, 2022, 173:109014.

[95] H Wang, Z Liu, D Peng, et al. Interpretable convolutional neural network with multilayer wavelet for noise-robust machinery fault diagnosis. *Mech Syst Signal Process*, 2023, 195:110314.

[96] Y Liu, H Jiang, C Liu, et al. Data-augmented wavelet capsule generative adversarial network for rolling bearing fault diagnosis. *Knowl-Based Syst*, 2022, 252:109439.

[97] J Ren, C Hu, Z Shang, et al. WavFormer: An interpretable wavelet-constrained transformer for industrial acoustics diagnosis. *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, IEEE, 2024: 1-6.

[98] B Ganguly, S Chaudhury, S Biswas, et al. Wavelet Kernel based convolutional neural network for localization of partial discharge sources within a power apparatus. *IEEE Trans Ind Inform*, 2020: 1–1.

[99] T Li, Z Zhao, C Sun, et al. WaveletKernelNet: An interpretable deep neural network for industrial intelligent diagnosis. *IEEE Trans Syst Man Cybern Syst*, 2021, 52: 2302-2312.

[100] Q Chen, X Dong, G Tu, et al. TFN: An interpretable neural network with time-frequency transform embedded for intelligent fault diagnosis. *Mech Syst Signal Process*, 2024, 207:110952.

[101] B Zhao, C Cheng, G Tu, et al. An interpretable denoising layer for neural networks based on reproducing Kernel Hilbert space and its application in machine fault diagnosis. *Chin J Mech Eng*, 2021, 34:44.

[102] M Zhao, S Zhong, X Fu, et al. Deep residual shrinkage networks for fault diagnosis. *IEEE Trans Ind Inform*, 2020, 16: 4681-4690.

[103] H Shao, M Xia, J Wan, et al. Modified stacked autoencoder using adaptive Morlet wavelet for intelligent fault diagnosis of rotating machinery. *IEEE-ASME Trans Mechatron*, 2022, 27: 24-33.

[104] J Bruna, S Mallat. Invariant scattering convolution networks. *IEEE Trans Pattern Anal Mach Intell*, 2013, 35: 1872-1886.

[105] J Andén, S Mallat. Deep scattering spectrum. *IEEE Trans Signal Process*, 2014, 62: 4114-4128.

[106] C Liu, X Ma, T Han, et al. NTScatNet: An interpretable convolutional neural network for domain generalization diagnosis across different transmission paths. *Measurement*, 2022:112041.

[107] Y Kim, K Na, B D Youn. A health-adaptive time-scale representation (HTSR) embedded convolutional neural network for gearbox fault diagnostics. *Mech Syst Signal Process*, 2022, 167:108575.

[108] J Pan, Y Zi, J Chen, et al. LiftingNet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification. *IEEE Trans Ind Electron*, 2018, 65: 4973-4982.

[109] J Yuan, S Cao, G Ren, et al. LW-Net: an interpretable network with smart lifting wavelet kernel for mechanical feature extraction and fault diagnosis. *Neural Comput Appl*, 2022.

[110] Y Jiang, T Xia, D Wang, et al. A spatiotemporal dynamic wavelet network for infrared thermography-based machine prognostic. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023, 54: 1658-1665.

[111] G Michau, G Frusque, O Fink. Fully learnable deep wavelet transform for unsupervised monitoring of high-frequency time series. *Proc Natl Acad Sci*, 2022, 119:e2106598119.

[112] Z Shang, Z Zhao, R Yan, et al. M-band wavelet network for machine anomaly detection from a frequency perspective. *Mechanical Systems and Signal Processing*, 2024, 216: 111489.

[113] Z Shang, Z Zhao, S Wang, et al. Anomaly detection from a frequency perspective: M-band wavelet packet anomaly detection network. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024: 5550-5554.

[114] Z Zhao, T Li, B An, et al. Model-driven deep unrolling: Towards interpretable deep learning against noise attacks for intelligent fault diagnosis. *ISA Trans*, 2022, 129: 644-662.

[115] B An, S Wang, F Qin, et al. Adversarial algorithm unrolling network for interpretable mechanical anomaly detection. *IEEE Trans Neural Netw Learn Syst*, 2023: 1-14.
[116] Z Ye, J Yu. Deep morphological convolutional network for feature learning of vibration signals and its applications to gearbox fault diagnosis. *Mech Syst Signal Process*, 2021, 161:107984.
[117] D Wang, Y Chen, C Shen, et al. Fully interpretable neural network for locating resonance frequency bands for machine condition monitoring. *Mech Syst Signal Process*, 2022, 168:108673.
[118] P Borghesani, N Herwig, W Wang, et al. Embedding signal processing knowledge in neural networks-an application to gear diagnostics. *AIAC 2023 20th Aust. Int. Aerosp. Congr. 20th Aust. Int. Aerosp. Congr., Engineers Australia Melbourne*, 2023: 669-677.
[119] H Lu, V P Nemani, V Barzegar, et al. A physics-informed feature weighting method for bearing fault diagnostics. *Mech Syst Signal Process*, 2023, 191:110171.
[120] T Xie, Q Xu, C Jiang, et al. The fault frequency priors fusion deep learning framework with application to fault diagnosis of offshore wind turbines. *Renew Energy*, 2023, 202:143-153.
[121] J Rani, T Tripura, H Kodamana, et al. Generative adversarial wavelet neural operator: Application to fault detection and isolation of multivariate time series data. arXiv preprint arXiv:2401.04004, 2024.
[122] M Russell, P Wang. Physics-informed deep learning for signal compression and reconstruction of big data in industrial condition monitoring. *Mech Syst Signal Process*, 2022, 168:108709.
[123] Y Yao, J Ma, Y Ye. Regularizing autoencoders with wavelet transform for sequence anomaly detection. *Pattern Recognit*, 2023, 134:109084.
[124] B Dai, G Frusque, Q Li, et al. Acceleration-guided acoustic signal denoising framework based on learnable wavelet transform applied to slab track condition monitoring. *IEEE Sens J*, 2022: 1–1.
[125] A Saxena, K Goebel, D Simon, et al. Damage propagation modeling for aircraft engine run-to-failure simulation. *2008 Int. Conf. Progn. Health Manag.*, 2008: 1-9.
[126] Y Duan, H Li, M He, et al. A BiGRU autoencoder remaining useful life prediction scheme with attention mechanism and skip connection. *IEEE Sensors Journal*, 2021, 21(9): 10905-14.
[127] S Fu, S Zhong, L Lin, et al. A novel time-series memory auto-encoder with sequentially updated reconstructions for remaining useful life prediction. *IEEE Trans. Neural Netw. Learn. Syst.*, 2022, 8.
[128] W Yu, I Y Kim, C Mechefske. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mech. Syst. Signal Proc.*, 2019, 129: 764-780.
[129] L Li, J Yan, H Wang, et al. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *IEEE Trans. Neural Netw. Learn. Syst.*, 2021, 32(3): 1177-1191.
[130] D Li, Q Tao, J Liu, et al. Center-aware adversarial autoencoder for anomaly detection. *IEEE Trans. Neural Netw. Learn. Syst.*, 2022, 33(6): 2480-2493.
[131] S Akcay, A Atapour-Abarghouei, T P Breckon. GANomaly: Semi-supervised anomaly detection via adversarial training. *Proc. Asian Conf. Comput. Vis. Cham*, Switzerland: Springer, 2018: 622-637.
[132] E Bechhoefer. Condition based maintenance fault database for testing diagnostics and prognostic algorithms. *MFPT Data*, 2013.
[133] Y LeCun, L Bottou, Y Bengio, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, 86(11): 2278-2324.
[134] W Zhang, G Peng, C Li, et al. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, 2017, 17(2): 425.
[135] R Liu, F Wang, B Yang, et al. Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions. *IEEE Transactions on Industrial Informatics*, 2019, 16(6): 3797-3806.
[136] M Ravanelli, Y Bengio. Speaker recognition from raw waveform with sincnet. *2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018: 1021-1028.
[137] Q Zhu, Z Zhao, J Yan. Physics-informed machine learning for surrogate modeling of wind pressure and optimization of pressure sensor placement. *Computational Mechanics*, 2023, 71(3): 481-491.
[138] S Kohtz, Y Xu, Z Zheng, et al. Physics-informed machine learning model for battery state of health prognostics using partial charging segments. *Mechanical Systems and Signal Processing*, 2022, 172: 109002.
[139] M Raissi, P Perdikaris, G E Karniadakis. Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561, 2017.
[140] C Anderson. The End of Theory: The Data Deluge Makes the Scientific Method Obsolete. *The Wired*, 2008-06-23.

Ruqiang Yan received the Ph.D. degree in mechanical engineering from the *University of Massachusetts at Amherst, MA, USA*, in 2007. He is currently a Professor of mechanical engineering with *Xi'an Jiaotong University, China*. His research interests include data analytics, AI, and energy-efficient sensing for health diagnosis of large-scale, complex, dynamical systems. Dr. Yan is a Fellow of *ASME* (2019) and *IEEE* (2022). He is also the Editor-in-Chief of the *IEEE Transactions on Instrumentation and Measurement*, an Associate Editor of the *IEEE Sensors Journal*, and Associated Editor-in-Chief of *Chinese Journal of Mechanical Engineering* and Editorial Board Member of *Journal of University of Science and Technology of China*.

Zheng Zhou received the B.S. degree in mechanical engineering from *Xi'an Jiaotong University*, China, in 2019. He is currently working toward his Ph.D degree in mechanical engineering at *School of Mechanical Engineering, Xi'an Jiaotong University*. From 2023, he is jointly working in *Intelligent Maintenance and Operations Systems lab* at EPFL. His current research is focused on physics-informed machine learning, automatic machine learning, and fault prognosis.

Zuogang Shang received the B.S. degree in mechanical engineering from *Xi'an Jiaotong University*, China, in 2020. He is currently working toward his Ph.D degree in mechanical engineering at *School of Mechanical Engineering, Xi'an Jiaotong University*. His current research is focused on explainable deep learning, mechanical fault diagnosis, and anomaly detection.

Zhiying Wang received the B.S. degree in mechanical engineering from *Dalian University of Technology,* China, in 2020. He is currently working toward his Ph.D degree in mechanical engineering at *School of Mechanical Engineering, Xi'an Jiaotong University*, China. His current research interests include physics-informed machine learning, dynamic modeling of hydraulic pimps, and digital twins.

Chenye Hu received the B.S. degree in mechanical engineering from *Xi'an Jiaotong University*, China, in 2020. He is currently working toward his Ph.D degree in mechanical engineering at *School of Mechanical Engineering, Xi'an Jiaotong University*. His current research interests include deep learning and mechanical fault diagnosis.

Yasong Li received the B.S. degree in mechanical engineering from *Jiangsu University*, China, in 2020. He is currently working toward his Ph.D degree in mechanical engineering at *School of Mechanical Engineering, Xi'an Jiaotong University*, China. His current research interests include interpretable deep learning, attention mechanism, remaining useful life prediction.

Yuangui Yang received the B.S. degree in mechanical engineering from *Xi'an Jiaotong University*, China, in 2020. He is currently working toward his Ph.D degree in mechanical engineering at *School of Mechanical Engineering, Xi'an Jiaotong University*. His current research interests include explainable deep learning, graph neural network, and gearbox fault diagnosis.

Xuefeng Chen received the Ph.D. degree in mechanical engineering from *Xi'an Jiaotong University,* China, in 2004. He is currently a Full Professor with *School of Mechanical Engineering, Xi'an Jiaotong*

University. He has authored more than 100 SCI publications in areas of composite structure, aeroengine, wind power equipment, and so on. His research interests include intelligent maintenance, advanced sensing technology, and deep learning. Dr. Chen was the recipient of the National Excellent Doctoral Thesis Award in 2007, the First Technological Invention Award of the Ministry of Education in 2008, the Second National Technological Invention Award in 2009, the First Provincial Teaching Achievement Award in 2013, and the First Technological Invention Award of the Ministry of Education in 2015. He was awarded the Science and Technology Award for Chinese Youth in 2013. He is also the Executive Director of the Fault Diagnosis Branch in Chinese Mechanical Engineering Society.

Robert X. Gao received the Ph.D. degree in mechanical engineering from the *Technical University of Berlin,* Berlin, Germany, in 1991. He is currently the Cady Staley Professor of engineering and the Chair of the *Department of Mechanical and Aerospace Engineering, Case Western Reserve University,* Cleveland, OH, USA. He holds 13 patents, authored or coauthored three books and more than 400 technical papers, including more than 200 journal papers. His research interests include physics-based sensing, multiresolution data analysis, stochastic modeling, and machine learning for improving the observability of cyber physical systems, toward improved process and product quality control. Dr. Gao is a Senior Editor for *IEEE/ASME Transactions on Mechatronics*. He was the Lead Guest Editor of the Special Issue on Data Science-Enhanced Manufacturing of the *ASME Journal of Manufacturing Science and Engineering*, and was an Associate Editor for several journals of the IEEE ASME and IFAC. He is a Fellow of ASME, SME, and CIRP, and a Distinguished Fellow of IIAV, the International Institute of Acoustics and Vibration. He was the recipient of several professional awards, including the ASME Milton C. Shaw Manufacturing Research Medal, IEEE Best Application in Instrumentation and Measurement Award, SME Eli Whitney Productivity Award, ASME Blackall Machine Tool and Gage Award, IEEE Instrumentation and Measurement Society Technical Award, NSF Early CAREER Award, and several best paper awards. He was a Distinguished Lecturer of the IEEE Instrumentation and Measurement Society and IEEE Electron Devices Society.

frontiers | Frontiers in Artificial Intelligence

TYPE Review
PUBLISHED 13 October 2023
DOI 10.3389/frai.2023.974295

OPEN ACCESS

EDITED BY
Zahra Ahmadi,
L3S Research Center, Germany

REVIEWED BY
Luigi Celona,
University of Milano-Bicocca, Italy
Maryam Amir Haeri,
University of Twente, Netherlands

*CORRESPONDENCE
Feng Xu
✉ fengxu@fudan.edu.cn

RECEIVED 21 June 2022
ACCEPTED 25 September 2023
PUBLISHED 13 October 2023

CITATION
Liu Z and Xu F (2023) Interpretable neural networks: principles and applications. *Front. Artif. Intell.* 6:974295. doi: 10.3389/frai.2023.974295

Interpretable neural networks: principles and applications

Zhuoyang Liu[1,2] and Feng Xu[1]*

[1]Key Lab of Information Science of Electromagnetic Waves, Fudan University, Shanghai, China, [2]Faculty of Math and Computer Science, Weizmann Institute of Science, Rehovot, Israel

In recent years, with the rapid development of deep learning technology, great progress has been made in computer vision, image recognition, pattern recognition, and speech signal processing. However, due to the black-box nature of deep neural networks (DNNs), one cannot explain the parameters in the deep network and why it can perfectly perform the assigned tasks. The interpretability of neural networks has now become a research hotspot in the field of deep learning. It covers a wide range of topics in speech and text signal processing, image processing, differential equation solving, and other fields. There are subtle differences in the definition of interpretability in different fields. This paper divides interpretable neural network (INN) methods into the following two directions: model decomposition neural networks, and semantic INNs. The former mainly constructs an INN by converting the analytical model of a conventional method into different layers of neural networks and combining the interpretability of the conventional model-based method with the powerful learning capability of the neural network. This type of INNs is further classified into different subtypes depending on which type of models they are derived from, i.e., mathematical models, physical models, and other models. The second type is the interpretable network with visual semantic information for user understanding. Its basic idea is to use the visualization of the whole or partial network structure to assign semantic information to the network structure, which further includes convolutional layer output visualization, decision tree extraction, semantic graph, etc. This type of method mainly uses human visual logic to explain the structure of a black-box neural network. So it is a post-network-design method that tries to assign interpretability to a black-box network structure afterward, as opposed to the pre-network-design method of model-based INNs, which designs interpretable network structure beforehand. This paper reviews recent progress in these areas as well as various application scenarios of INNs and discusses existing problems and future development directions.

KEYWORDS

model decomposition, semantic graph, interpretable neural networks, electromagnetic neural network, interpretability

1. Introduction

Human natural intelligence arises from the evolutionary innate brain after empirical learning. Human intelligence has invented computing technology, and now people hope to use it to implement artificial intelligence (AI). With massive big data and high-performance computing, the emergence of deep learning, that is, DNNs have led to the explosive development of AI. However, DNNs are still essentially a function-fitting technique. They are black-box methods lacking interpretability and have weak generalization ability when the network doesn't have enough high-quality training data. The ability to logically reason is one of the basic characteristics of human intelligence. Getting inspiration from the

process of human logical reasoning to realize explainable AI is one of the directions of next-generation AI.

Human intelligence's logical reasoning can be classified as either deductive or inductive reasoning (Goswami, 2011). Deductive reasoning starts with a clear premise, which often is a well-known fact or truth. It can be used to construct a theoretical model through principles, so it has a rigorous expression (Clark, 1969; Johnson-Laird, 1999). Inductive reasoning is similar to data analysis, fitting, and clustering in that it draws on prior experience to predict current or future events (Sternberg and Gardner, 1983; Heit, 2000). As can be seen, existing deep learning (DL) approaches are analogous to inductive reasoning, that is, inducing principles from massive datasets. However, human inductive reasoning is interpretable since the process of human induction follows a well-defined semantic framework. Specifically, humans use eyes to sense the world and then use inductive reasoning to infer the type of a new thing and obtain semantic information from prior knowledge. Hence, by learning from the nature of human inductive reasoning, existing DL methods can accomplish semantic INN. On the contrary, traditional non-machine learning methods are similar to deductive reasoning, which refers to the process of developing theoretical models based on domain knowledge by specialists. Then the appropriate algorithms are developed to solve these problems by utilizing the theoretical models. Thus, interpretability can obviously be achieved by drawing inspiration from theoretical model decomposition.

As shown in Figure 1, this paper reviews and analyzes the existing INN research according to these two ideas. Model decomposition alternative INN learns from traditional theoretical models, that is, the combination of DL models and theoretical models to realize INN with the domain knowledge embedded in the network designing. On the contrary, semantic INN is closer to human semantic interpretation, and it is the combination of DL with the process of semantic inductive reasoning, which adds clear semantic information to neural networks (NNs) afterward. These two approaches can help mitigate issues of current data-driven approaches such as weak generalizability, inexplicability, and low fidelity. The principles of defining and implementing INN are mainly discussed in this paper. This section discusses the origin of INN along with the development and limitations of DL, and finally gives its definition and development as well as the practical applications.

1.1. Demands and challenges of INN

The rapid development of DNN has benefited from big data, improved algorithms, and high-efficiency computing. However, the current DL methods are completely data-driven, which means that very large-scale annotated data are required for training to get ideal results (McCulloch and Pitts, 1943; Deng et al., 2009; Dahl et al., 2011; He et al., 2015; Krizhevsky et al., 2017; Zhou et al., 2017; Montavon et al., 2018). As a black-box approach, it has serious drawbacks in terms of robustness and interpretability. In many AI-powered application scenarios such as autonomous driving, target recognition, etc., interpretability and robustness are crucial aspects of AI technology. Gregor and LeCun (2010) were the first people to put forward the theory of interpretability of neural networks. They adopted the method of combining sparse coding with traditional NNs so that DL inherited the model-based method's interpretability and the learning-based method's efficiency. According to the current research, we summarize the existing approaches of constructing an INN as two groups, which are the model decomposition alternative INN and the semantic INN, based on the way to perform inference, as shown in Figure 2.

Conceptually, the common thread of delivering applications based on the model decomposition alternative INN is related to human deductive reasoning. While implementing the model decomposition alternative INN, the investigator decomposes the conventional algorithm based on the mathematical-physical model into several calculation steps, which can be transformed into the computation process of a NN. Similarly, the common thread of explaining applications based on the semantic INN is relevant to human inductive reasoning. The implementation of the semantic INN is to construct the explanation graph with the assistance of engineers, which in turn helps them determine whether the network is working correctly. Following the above INN construction principles, Sections 1.2, 1.3 describe the application of two types of INN in detail.

1.2. Applications of model decomposition alternative INN

The model decomposition alternative INN is based on the facts of the real world. It decomposes a complex mathematical model or physical model into several modules that are easier to handle. Then, according to the prior knowledge, it transforms the computational process of these modules into NNs' hyper-parameters or hidden layers so that the NNs are interpretable (Zhang et al., 2018; Shlezinger et al., 2020). This kind of interpretable method is equivalent to unfolding the "black box" of the original NNs and using some artificial and controllable parameters and structures to replace the weights without mathematical and physical meaning in DNN. In order to extract these artificial and controllable parameters and structures, the problem must have a theoretical model. Applications of INNs based on mathematical models, physical models, and some other models are given in Figure 2.

For example, the mathematical modeling problem solved by convex optimization or non-convex optimization algorithms can be used to guide the designing of the objective function. This method can be used to solve common partial differential equation (PDE) (Rudy et al., 2017; Zhang et al., 2019b; Rackauckas et al., 2020) or image deblurring, super-resolution, and other tasks (Daubechies et al., 2004; Wang et al., 2015; Li et al., 2020).

The role of the physical model in model decomposition alternative INN is different from that of the mathematical model. The computing process and parameters that have physical meanings of standard algorithms solving physical models are replaced by hidden layers and weights in NNs. In the field of electromagnetic physics, Fan et al. combine the finite difference time domain (FDTD) method to construct an recurrent neural network (RNN) to model the propagation of the wave equation and estimate the medium parameters (Hughes et al., 2019).

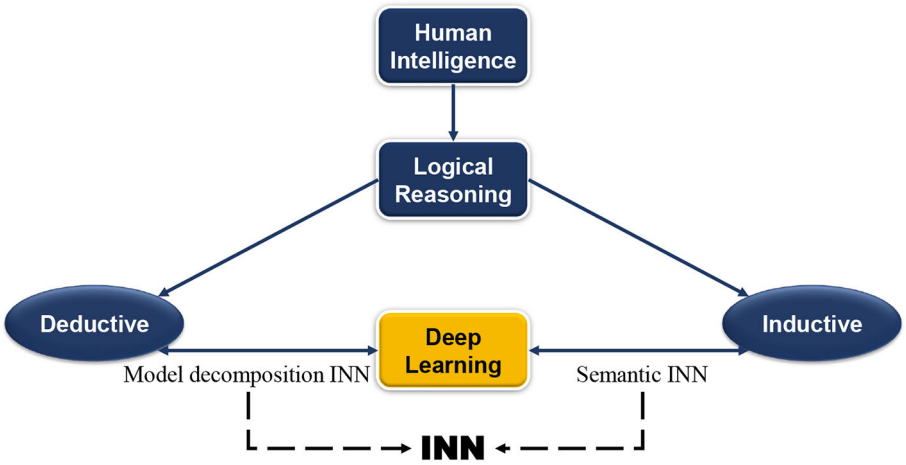


FIGURE 1
Human intelligence and artificial intelligence.

Guo et al. (2021) use the method of moment (MOM) to construct an INN, which computes the forward scattering field in a two-dimensional plane and predicts the inverse scattering parameters (Li et al., 2018; Wei and Chen, 2019; Xu et al., 2020). In the field of fluid mechanics and dynamics, Fang et al. (2021) use data-driven acceleration to deduce the speed and position of turbulent motion and used a physics-informed neural network (PINN) to discover the parameters of the higher-order nonlinear Schrodinger equation (NLSE). There are also many pieces of research using PINN to solve dynamic equations with a special INN method (Brunton et al., 2016; Sirignano and Spiliopoulos, 2018; Kochkov et al., 2021).

Other models in Figure 2 include non-mathematical physical models, such as problems in the field of biochemistry. By dealing with fluorescent images, Belthangady and Royer (2019) and Li et al. (2021) summarized applications of using DL to achieve microwave fluorescence image reconstruction. In ultrasound imaging, the intensity of clutter signals is usually relatively large and the distribution range is relatively wide, which seriously affects the accuracy of ultrasound imaging. The INN combined with principal component analysis (PCA) is proposed to achieve main beam extraction and clutter removal in Chien and Lee (2017); Lohit et al. (2019); Solomon et al. (2019). The performance of model decomposition INN is closely related to the physical limitations of the specific theoretical models.

1.3. Applications of semantic INN

Another direction is semantic INNs, and their interpretability mainly comes from the perspective of the human brain to realize the interpretable meaning of DNNs (Fan et al., 2021). Obviously, this is closely related to semantics, which means features or attributes described by people in language, and it reflects the process of people's understanding of the real world. Furthermore, it includes three aspects, i.e. visualization of convolution neural network (CNN), decision tree regularization, and semantic knowledge graph.

Visualization CNN is an interpretable method for a trained model. The basic idea is to display the output of each feature map of the network in the form of a weight heat map to show what role each layer played in accomplishing a given task (Wang et al., 2018a; Zhang and Zhu, 2018). In other words, it can activate different regions in the layers of NNs to distinguish the meaningful parts of the input (Guidotti et al., 2018). Its interpretability is reflected in the visualization, and the NN model itself is still a "black box."

The methods based on decision trees are proposed to help achieve the interpretability of NNs (Frosst and Hinton, 2017; Wu et al., 2018). The decision tree is a directed graph composed of parent nodes and child nodes. Its parent nodes and child nodes have semantic information, and the directed connections of decision trees make the path between the parent node and each child node also meaningful. Combining the decision trees which are the prior knowledge with layers of NNs can enhance the interpretability and robustness of DNNs. According to the region where the regularization acts, the interpretability method of decision tree extraction is divided into three types, which are global, local, and regional regularization decision trees, respectively (Lapuschkin et al., 2019; Wu et al., 2020).

The third approach is an interpretable graph neural network (GNN) that combines semantic graphs and DNNs, and its main idea is to utilize the semantic information contained in graphs to enhance the interpretability of DNNs. Zhang et al. (2017) use the AND-OR structure to realize target recognition (Si and Zhu, 2013; Akula et al., 2019), and the knowledge graph (KG) was mapped to the convolutional layers and the pooling layers. George et al. (2017) add side connections to form a recursive cortical network (RCN) which realized the verification code images denoising. The recently emerging zero-shot learning utilizes a mixture of KG, GNNs, and DNNs, which are combined with KG to realize the function of NNs inference learning, and multi-sample detection or recognition (Lampert et al., 2009; Palatucci et al., 2009; Kipf and Welling, 2016; Wang et al., 2018b; Chen et al., 2019; Lu et al., 2019; Yue et al.,

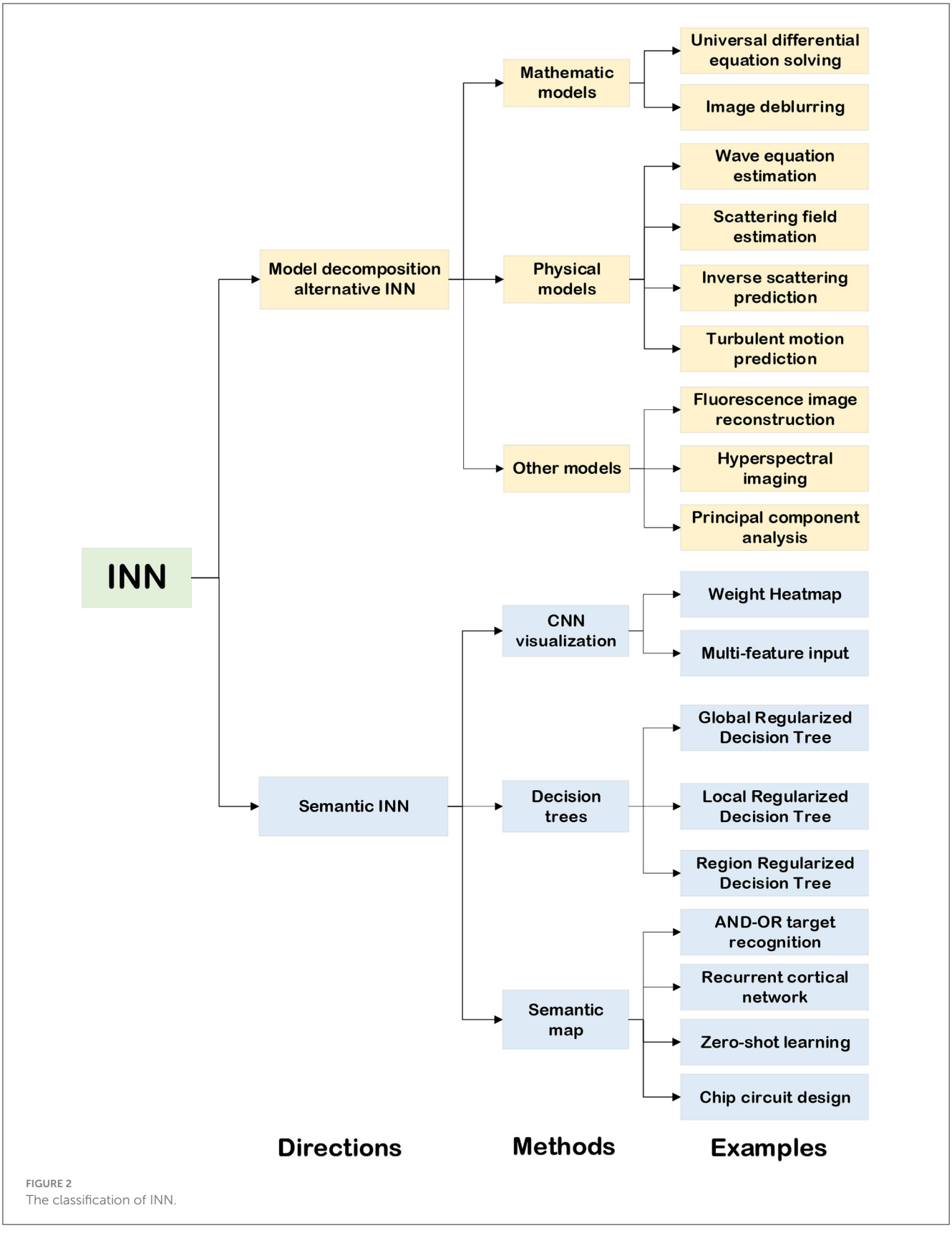


FIGURE 2
The classification of INN.

2021). In the field of integrated circuit (IC) design, Mirhoseini et al. regard the circuit diagram as a GNN and used the semantic feature extraction to complete efficient and accurate IC design (Mirhoseini et al., 2021). Chen et al. (2021) use a GNN expansion to crop an overlapped graph, extract the main parts of the graph, and realize graph denoising.

1.4. Comparison of model decomposition alternative INN and semantic INN

The research on interpretability is currently in the development stage. A large body of literature describes the implementation of explaining NNs and the construction of INNs. The above sections introduce two types of INN techniques and list the applications of INNs in signal processing, image classification, solving differential equations, etc. To better comprehend the basic principles of INNs, we emphasize that model decomposition alternative INN provides the mapping between the mathematical-physical model and NN's parameters or structures, while the semantic INN extracts the explanation graphs from NN by engineers using standard coding methods. The former converts mathematical-physical models that humans can understand into operators that computers can recognize, while the latter transforms the output of computers into semantics that humans understand.

Specifically, for the presented application of the model decomposition INN method, such as solving differential equations and image restoration, embedding a mathematical-physical model into the NN enhances the robustness of network training and convergence performance. However, those applications mentioned here have no semantics, so they are not reasonable for verifying network results with semantic INN analysis. Similarly, considering image classification tasks using semantic INN, extracting explanation graphs from pre-trained NNs assists engineers in evaluating the training state of NNs and improving their reliability. Still, the image classification task is hard to describe as a mathematical-physical model, thus it is not realistic to modify the network structure by embedding the traditional model. In other words, those two types of INNs are suitable for different tasks, and we need to choose the corresponding INN method according to the requirements.

To sum up, INNs are widely used in various fields. People pay great attention to the principles of high efficiency of NNs, and INN can give a reasonable explanation that ensures the reliability and security of network outputs. This paper focuses on the definition of INN and how to use INNs. In the following sections, the model decomposition alternative INNs and semantic INNs are introduced in detail. Finally, we present the application of INNs to solve practical electromagnetic problems and conclude with a summary.

2. Model decomposition alternative INN

In this section, we will analyze the way to use mathematical, physical, and other models of a given task to achieve model decomposition alternative INNs from the perspective of different models. Figure 3, presents the alternative approaches of implementing model decomposition alternative INNs and the interpretable regions of the NNs.

2.1. Mathematical model-decomposition INN

First of all, the mathematical model has a very broad concept. Almost all problems can be represented by a mathematical model. Without losing the generality, the mathematical model in this article can be expressed by the function $f(x,\theta)$, where x represents the input variables, θ represents the parameters of the mapping relationship of the function $f(x,\theta)$. Subsequently, the process of network training is the process of recovering the parameters of the mapping, holds that

$$\hat{y} = f(x_1, x_2, ..., x_i, ..., x_n; \theta_1, \theta_2, ..., \theta_j, ..., \theta_m), \tag{1}$$

where the $\hat{y}$ denotes the estimated output of the mathematical model. The specific expression of the function $f(x,\theta)$ is not our major concern here. It is decomposed as the optimized objective of the INN for training. The next step is to define the optimized objective or loss function according to the mapping function f. Generally, the loss function is denoted by $L(\hat{y},y)$, as shown in Equation (2).

$$L(\hat{y},y) = Distance(\hat{y}-y), \tag{2}$$

where y represents the true value of the solution, and $L(\hat{y},y)$ refers to the "distance" between the true value and the model output. In classification fields, "distance" can be expressed in terms of probability, that is, they choose the cross-entropy loss as the loss function. In regression tasks, "distance" is usually expressed in terms of norms, and l_1-norm and l_2-norm are both common choices. In image processing, "distance" reflects the reconstruction performance between the real image and the processed image, and the structural similarity index method (SSIM) is usually used as the evaluation standard for images. Accordingly, it's essential to choose the most suitable loss function when dealing with different types of problems.

The last step of the INN based on the mathematical model is to decompose the optimized objective, and the alternating direction method of multiplier (ADMM) (Boyd et al., 2011), half-quadratic splitting (HQS) (Wang et al., 2008), and conjugate gradient (CG) (Liu and Storey, 1991; Hager and Zhang, 2006) are widely used in convex optimization problems. In addition, the Markov chain Monte Carlo (MCMC) method (Geyer, 1992; Pereyra et al., 2020) combined with Bayesian estimation is applied to solve non-convex optimization problems. This subsection starts with the regression problem of solving PDE and the image processing problem of image deblurring. It then expands the basic principle of INN based on a mathematical model and gives its general pipeline in Figure 4.

2.1.1. Universal partial differential equations

The mesh-based techniques are widely used in solving differential equations. The basic idea of it is to mesh the differential

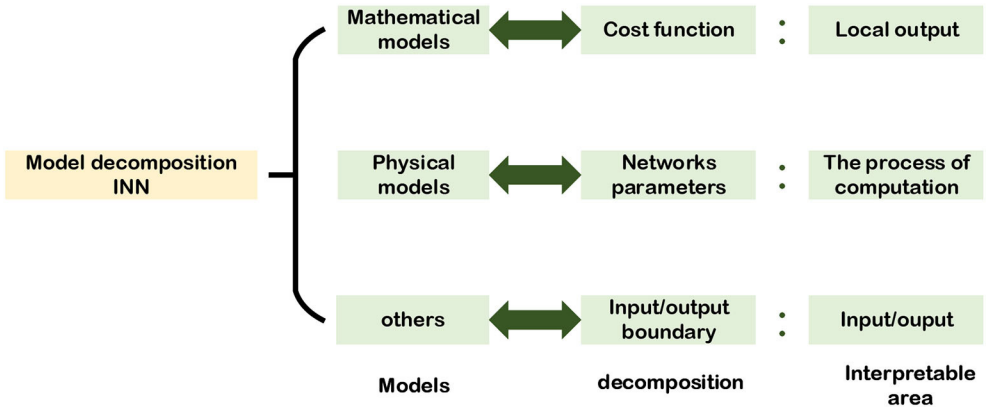


FIGURE 3
Model decomposition alternative INN.

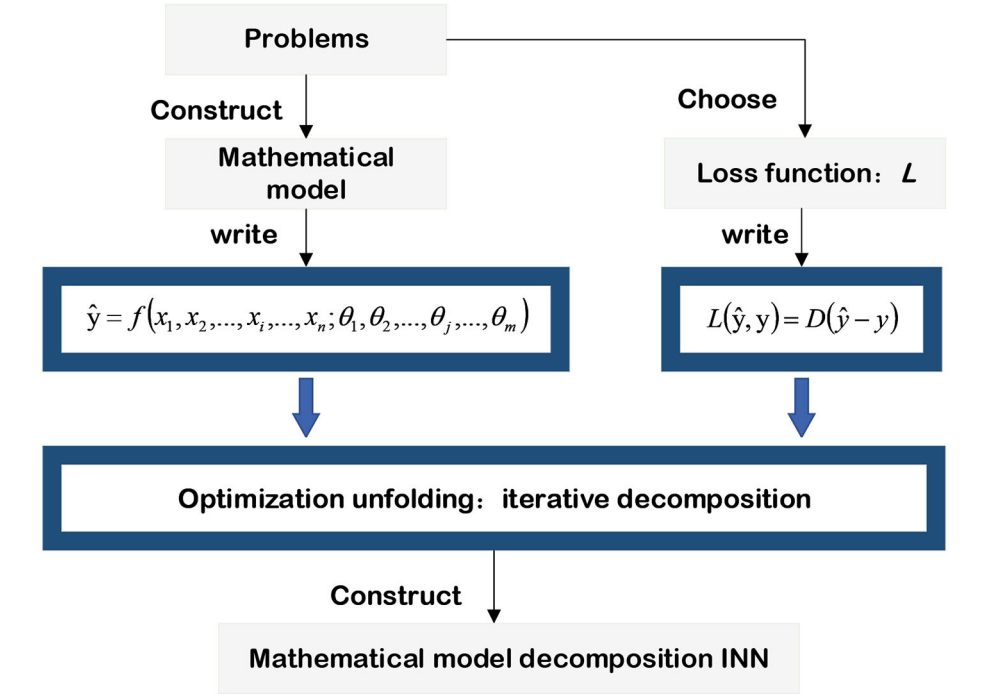


FIGURE 4
Mathematical model-decomposition INN.

equations into grids according to the small amount Δt of each step, and then the relationship $f(t_k; t_{k+1})$ between the previous moment and the next moment is written with the unit small amount Δt. Finally, by using iterative processes, the relationship between the start time and the end time will be found. However, due to the exponential growth in the number of mesh points with the number of dimensions, it is impossible to solve high-dimensional PDEs with mesh-based techniques. In contrast, the data-driven approach of machine learning (ML) can be more flexible and allows one to drop the simplifying assumptions that are needed to derive theoretical models from the data. Therefore, many scholars consider using ML to solve differential equations, and Rudy et al. (2017) found the terms of the controlling PDE that most properly described the data from a wide library of probable candidate functions using data-driven sparse regression techniques. The basic idea is to use the value of $f(x, y)$ at a space-time sampling grid to infer the PDE which is satisfied by the system. First, they assume that the PDE can be represented by a series of functions:

$$f_t = N(f, f_x, f_{xx}, ..., x, \mu), \tag{3}$$

where the subscript represents the differential of the function f in time or space, $N(\cdot)$ is the uncertain parts in PDEs, and μ represents other parameters that may be related to configuration.

Rudy et al. (2017) replace the combination of multiple functions with F, and replace other influencing parameters with P. Then, the PDE of this system can be written as:

$$F_t = \Theta(F, P)\xi. \tag{4}$$

The dictionary Θ contains all possible entries in the PDE for a given system. ξ is a sparse vector, and each non-zero item of it indicates that there is a corresponding entry of the dictionary Θ in the controlled PDE of the system. Each entry of F is a specific candidate term for a certain point in space at a certain moment, and each entry of P represents the influenced input of the system, which is also assigned to each point and moment. Using sparse regression to find the controlled PDE of a given system without searching for all possible components can effectively reduce the calculation complexity. However, there are still problems with a large number of matrix calculations and the lack of scientific principles in data-driven models. In Rackauckas et al. (2020), Rackauckas proposed an ML method that combines domain scientific knowledge and called this combined model the universal differential equation (UDE). The scientific knowledge is incorporated into the NNs while achieving two goals: reducing the size of the NN structure and speeding up the solution of differential equations.

Let's consider a quadratic ordinary differential equation (ODE) problem. Assuming that there is a natural ecosystem that consists of prey and predators. The variation of prey $\dot{x}$ is related to its birth rate and capture probability, while the increase of predators $\dot{y}$ is also related to its birth rate and capture influence. In particular, the capture effect of the prey and predators is mutual, and the variance of both prey and predators can be written as an ODE, namely

$$\dot{x} = ax - bxy, \tag{5}$$

$$\dot{y} = cy - dxy, \tag{6}$$

where a and c are the birth rates of the prey x and predators y, and the b and d are the mutual influence rate of the prey and predator, respectively. In this case, the mutual influence rates of both targets need to be estimated. To address this problem in a standard method, it is necessary to mesh y and x into data points and then extract the correlation coefficients b and d by data fitting. Finally, according to the initial ODE, the value of data points at the next moment can be deduced as follows:

$$\frac{x_k - x_{k-1}}{t} = ax_{k-1} - bx_{k-1}y_{k-1}, \tag{7}$$

$$\frac{y_k - y_{k-1}}{t} = cy_{k-1} - dx_k y_{k-1}, \tag{8}$$

$$x_k = (at + 1)x_{k-1} - bty_{k-1}x_{k-1}, \tag{9}$$

$$y_k = (ct + 1)y_{k-1} - dtx_k y_{k-1}. \tag{10}$$

The traditional method for deriving the solutions of ODEs is only suitable for the case of low order and low dimension. In the mathematical model-decomposition INN, DL approaches are used to learn unknown interactions between x and y, which means that the second parts in Equations (5, 6) correspond to NNs.

As shown in Figure 5, combined with the iterative processes for solving ODEs, a universal ordinary differential equation (UODE)-based symbolic regression is constructed, and scientific knowledge and prior conditions are integrated into the process of discovering and solving ODEs, which reduces the number of trials and errors of the network (Li et al., 2020). And due to the prior conditions of a realistic system, it is no longer essential to construct a dictionary matrix containing each derivative term of the independent variable when applying a UODE-based symbolic regression to discover and solve ODEs, and only the finite term polynomial coefficients need to be estimated. Therefore, using the mathematical model-decomposition INN can greatly reduce its computational complexity.

2.1.2. Image deblurring

In the area of image processing, traditional model-based algorithms include image erosion and expansion, edge gradient extraction, Fourier transform, wavelet transform, and matched filtering (Ramella and Sanniti di Baja, 2007; Danielyan et al., 2011; Burger et al., 2012). Due to the domain transformation and matrix inversion procedures in traditional image processing, the edge shadow and ringing effect may happen in the image deblurring. Then, the DL approach has become popular over time, and many researchers employ ResNet for target recognition, UNet for image segmentation, VGGNet for target detection, and generative adversarial network (GAN) for image deblurring (Nah et al., 2017; Kupyn et al., 2018; Tao et al., 2018). However, there are still difficulties for DNNs in realizing image processing with a small sample size, which motivates the development of INNs. These three types of methods and their advantages and disadvantages are compared in Table 1. This subsection mainly introduces how to build an INN for single-image deblurring.

According to the flowchart given in Figure 4, the first step in realizing a mathematical model-decomposition INN for image processing is to establish a general mapping function. Combined with the properties of the image degradation model, it is assumed that the mapping function can be written as

$$y = Wx, \tag{11}$$

where W represents the blurred kernel, x is the original image, and y is the blurred image, which is also the input of the INN. It's defined as an inverse problem, and the final output is the recovered x. Then, by adopting the optimization algorithm of the iterative shrinkage and thresholding algorithm (ISTA), the objective function can be expressed as:

$$x = \underset{x}{\operatorname{argmin}} \|y - Wx\|_2^2 + \lambda \|x\|_1, \tag{12}$$

where λ is the regularization parameter, which is used to ensure the sparsity of the results. According to the ISTA, the optimization problem of Equation (12) can be transformed into iterative solutions, where each iteration computes one step of x. Then x can

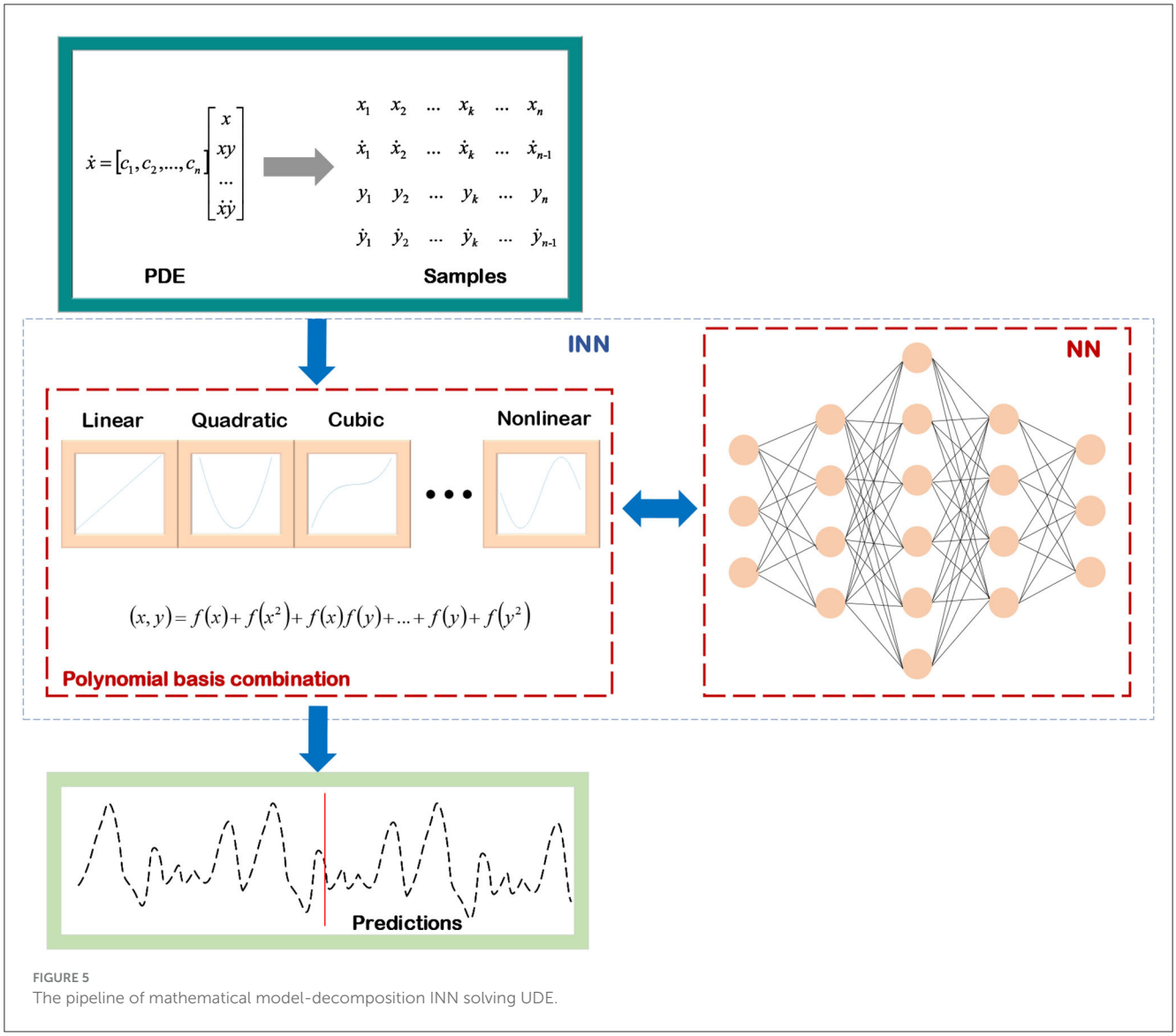


FIGURE 5
The pipeline of mathematical model-decomposition INN solving UDE.

TABLE 1 Companion of various image deblurring methods.

Methods	Advantages	Disadvantages
Matched filters	It can be used to process pictures of unknown blurred kernels with interpretability.	Matched filters require a transform domain, potentially causing ringing and loss of resolution.
Purely DL	It's efficient in real-time with low computational complexity.	DL cannot handle images of types of images that are unknown in the training set.
INN	It can reconstruct pictures of unknown blurred kernels with high efficiency, interpretability, and low computational complexity.	Designing and training INN are complicated techniques.

be estimated as Beck and Teboulle (2009)

$$x_{k+1} = S_\lambda \left(x_k - 2t_k W^T \left(W x_k - y \right) \right), \tag{13}$$

where S_λ is a shrinkage operator that updates x by performing a soft threshold operation on the outputs. The update formula is as follows

$$S_\lambda = \text{sign}(x) \cdot \max\{|x| - \lambda, 0\}. \tag{14}$$

In order to better represent the updated x in each iteration, Beck and Teboulle (2009) separate the input variable y and output variable x of each iteration in Equation (13), and the suitable step size of gradient descent is replaced by $\mu_k = 1/2t_k$. Then, it can be recast as

$$x_{k+1} = S_\lambda \left\{ \left(1 - \frac{1}{\mu_k} W^T W \right) x_k + \frac{1}{\mu_k} W^T y \right\}. \tag{15}$$

Based on the Equation (15), (Li et al., 2020) proposed to unfold the iterative processes into the learnable module. They transfer the parameters in the iteration to the network at the same time and use the method of minimizing the training loss function to continuously estimate the NN's parameters to achieve adjusting

the network structure. At this point, the mathematical model-decomposition INNs require fewer iterations than the model-based method, the parameters calculated in the NN have mathematical meanings, and the training of INNs no longer relies on large-scale datasets. Zhang et al. (2020) proposed an INN based on the image degradation model to achieve single image super-resolution (SISR) (Daubechies et al., 2004), the specific process is shown in Figure 6. First, the image degradation as shown in Equation (16) is constructed.

$$y = k * x + n, \tag{16}$$

where k is the kernel, x is the sharp image, y is the blurred image, and n is the additive white Gaussian noise (AWGN). In order to achieve the SISR, the energy function in the form of Equation (12) is constructed. The goal of deblurring is to minimize the energy function, and its expression is as follows:

$$E(x) = \frac{1}{2\sigma^2}\|y - (k * x)\|^2 + \lambda\Phi(x), \tag{17}$$

where λ is used to control the weight of the prior term $\Phi(x)$ on the deblurring process, and σ represents the noise coefficient. They use HQS to separate the prior term and the data term of Equation (17), and apply the ISTA method aforementioned to complete SISR. The decoupled data term and prior term are expressed as Monga et al. (2021)

$$z_k = \arg\min_z \|y - (k * z)\|^2 + \mu\sigma^2 \left\|z - x_{k-1}\right\|^2, \tag{18}$$

$$x_k = \arg\min_x \frac{\mu}{2}\|z_k - x\|^2 + \lambda\Phi(x), \tag{19}$$

where μ is the step size.

The pseudo-inverse algorithm is used to directly calculate the value of z in the data part. Nevertheless, the pseudo-inverse bias matrix is a large filter kernel, which can be solved by cascading into multiple smaller kernels. For the SISR problems, the prior term is considered an image denoising process of z, which can be replaced by the form of a ResNet. Following the method given in Daubechies et al. (2004), this subsection implements the deblurring on the DIV2K dataset (Timofte et al., 2018), and Figure 7 shows the deblurring results of the INN when the images are degraded with different blurred kernels.

2.2. Physical model-decomposition INN

Physical model-decomposition INNs are primarily concerned with issues in physical electromagnetism and dynamics. Actually, the term "model" here not only refers to physical models described by a formula but also includes constraints and principles in physics. Compared with the mathematical model-decomposition INNs, the basic idea of this approach is to convert the domain knowledge contained in the physical model into NN parameters and replace the calculation processes in the physical model with the layers of the NNs, as shown in Figure 8. Starting with an electromagnetic model, the tasks of wave dynamics prediction and forward and inverse scattering predictions are described in detail and the turbulent motion prediction is introduced briefly.

2.2.1. Wave equation prediction

The electromagnetic wave radiates outward with a specific pattern in free space, and its fluctuation mode is determined by the exciting source and medium characteristics. The propagating direction and fluctuation state at each point in the wave propagation are related to the previous moment, implying that wave propagation is the same as the continuous time series. This subsection introduces a continuous physical model for wave propagation in free space. Assuming that the exciting source $f(r, t)$ emits spherical waves, and Equation (20) shows the time domain wave-based dynamics of the scalar electric field u in free space.

$$\frac{\partial^2 u}{\partial t^2} - c^2\nabla^2 u = f(r, t), \tag{20}$$

where c and t represent the speed of light and time steps, respectively. Compared with solving PDEs with mathematical model-decomposition INN, the formula (20) is discretized by finite difference to obtain the form of the wave equation related to the temporal step Δt, as formulated:

$$\frac{u_{t+1} - 2u_t + u_{t-1}}{\Delta t^2} - c^2\nabla^2 u_t = f_t(r, t), \tag{21}$$

where the subscript t is the value of a scale electric field at the given time. Fan et al. (2021) built a mapping between the physical parameters in the discrete wave equation and the neurons in the RNN (Hughes et al., 2019), as shown in Figure 9.

In an iterative process, the state vector h_t is defined as the combination of scalar field's values within a temporal step Δt, which is a column vector connected to sampling time and can be expressed as Equation (22). Then, they substitute it into Equation (21) to obtain the state vector of the scalar field, as shown:

$$h_t = \begin{bmatrix} u_{t+1} \\ u_t \end{bmatrix}, \tag{22}$$

$$h_t = \begin{bmatrix} 2 + \Delta t^2 c^2\nabla^2 & -1 \\ 1 & 0 \end{bmatrix} h_{t-1} + \Delta t^2 \begin{bmatrix} f_t(r, t) \\ 0 \end{bmatrix}. \tag{23}$$

Considering the wave equation prediction, the process of calculating the state vector of the scalar electric field is converted into a layer of RNN. Then, the hierarchical model of the RNN can be written as

$$h_t = \sigma^{(h)}\left(W^{(h)} \cdot h_{t-1} + W^{(x)} \cdot x_t\right), \tag{24}$$

$$y_t = \sigma^{(y)}\left(W^{(y)} \cdot h_t\right), \tag{25}$$

where $W^{(h)}$, $W^{(x)}$, $W^{(y)}$ are the trainable parameters in the RNN, $\sigma^{(h)}$, $\sigma^{(y)}$ are nonlinear activation functions. Combining Equations (23, 24), the mapping between the physical parameters of the discrete wave equation and the weight parameters of the RNN state equation can be constructed. They are shown as:

$$W^{(h)} = \begin{bmatrix} 2 + \Delta t^2 c^2\nabla^2 & -1 \\ 1 & 0 \end{bmatrix}, \tag{26}$$

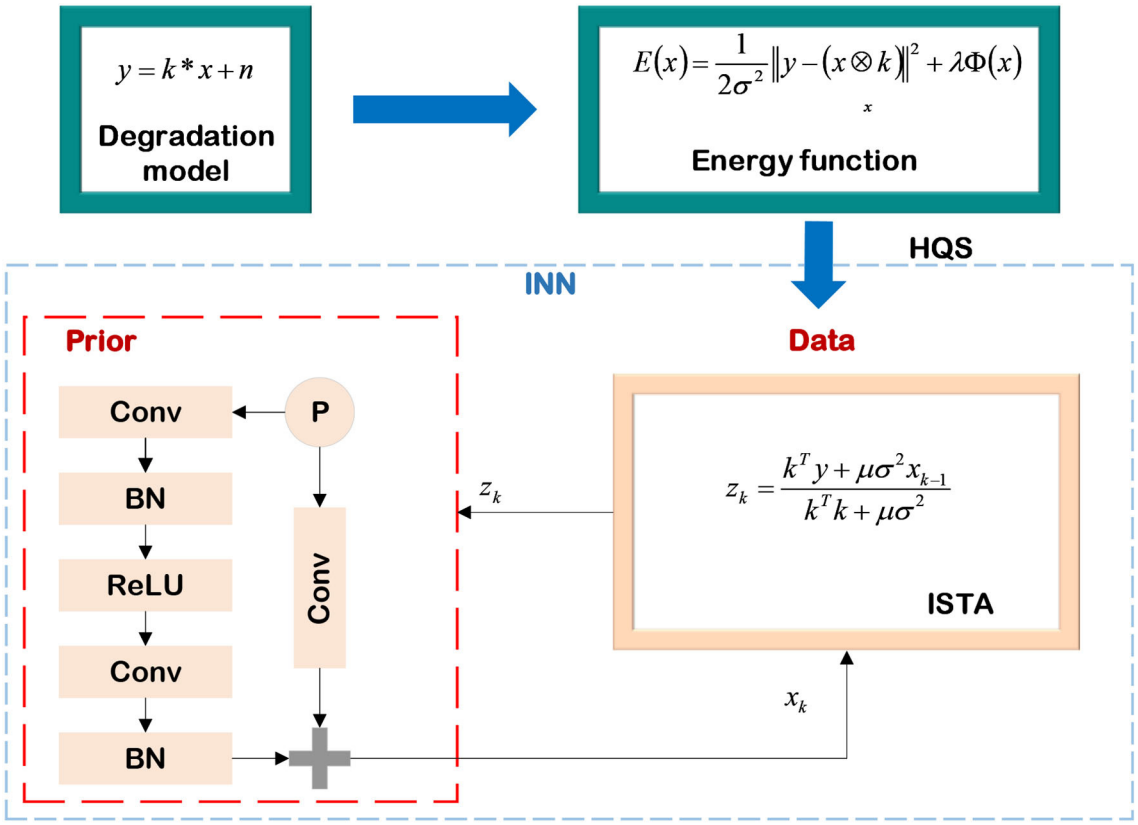


FIGURE 6
Single image super-resolution based on INN.

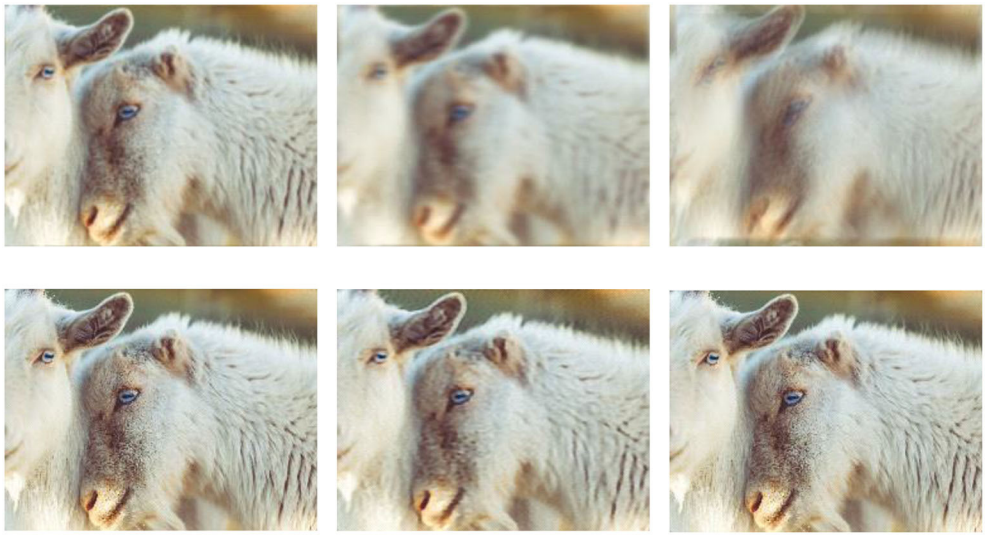


FIGURE 7
Examples of SISR with different blurred kernels. The first row includes three low-resolution images, and the images in the second row correspond to the super-resolution results of the low-resolution images in the first row. The images from left to right are distinguished by different blurred kernels, which are homogeneous Gaussian kernels, anisotropic Gaussian kernels, and motion-blurred kernels, respectively.

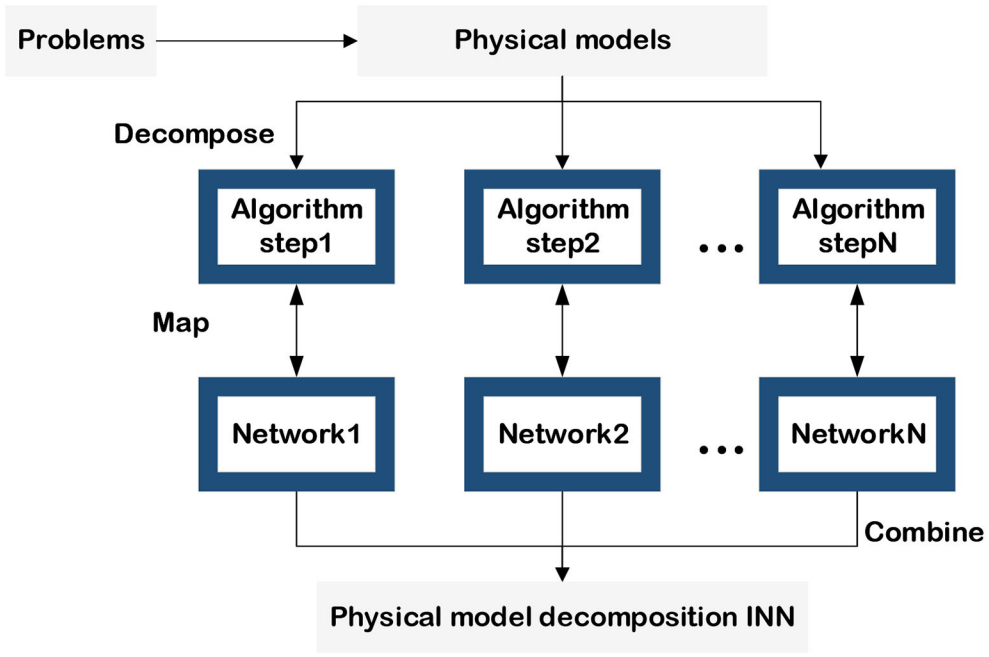


FIGURE 8
Physical model-decomposition INN.

$$W^{(x)} = \Delta t^2. \tag{27}$$

The input parameters x_t of the wave-based RNN are determined by the exciting source $f_t(r, t)$ in free space, and the trainable weights of the RNN are related to the speed and scope of wave propagation. This means that the structure of the RNN is reasonably mapped to a physical model, and the parameters of the NNs have clear physical meanings. The wave-based RNN combined with FDTD to achieve wave equation prediction has certain interpretability. Furthermore, assuming that there is a medium in space, it's achievable to obtain the mode of wave propagation in the medium with the backward of the state matrix and estimate the dielectric constant of the dielectric layer and some other dielectric parameters from the correspondence between the trainable weight matrix $W(h)$ and the propagation velocity.

2.2.2. Electromagnetic scattered field estimation

The numerical calculation method for the electromagnetic scattering problem can be adaptive to estimate the scattering field of various shapes of the medium, but it is faced with obstacles of high computation complexity in complicated models. Using the DL method to accelerate the numerical calculation of electromagnetic scattering problems is a new direction in this field, which combines the parallel computing ability and high efficiency of NNs with the generalization ability and stability of numerical calculation methods. The concept of this type of INN accelerating numerical calculation can help us address more electromagnetic scattering issues in the future.

This subsection mainly discusses how to use physical model decomposition INNs to solve the forward scattering problem of dielectric layers as shown in Figure 10. Starting with converting the continuous forward scattering equation into a discrete scattered field model and then combining it with the conventional electromagnetic calculation algorithm to solve the discrete scattered field problem, it is ultimately replaced with a NN.

Considering a lossy dielectric scatterer in two-dimensional free space. The position of this scatterer is denoted as $r = (r_x, r_y)$. It is assumed that the electromagnetic wave emitted by the transmitting antenna is a transverse electromagnetic wave in the z direction. Then the permeability of the scatterer remains consistent with free space, and the complex permittivity varies with distance and frequency, which can be expressed as

$$\varepsilon(r) = \varepsilon_0 \varepsilon_r(r) - \frac{j\sigma(r)}{\omega}, \tag{28}$$

where ε_0 is the permittivity in free space, ε_r is the relative permittivity, σ is the conductivity, and ω is the angular frequency of the incident electromagnetic wave. For any direction incident field $E^{inc}(r)$, the computation principle of scattering in the far field is consistent with the electric field integral equation (EFIE). Therefore, the total electric field $E^{tot}(r)$ can be measured with the incident electric field plus the scattered electric field obtained by the secondary radiation on the surface of the scatterer, as shown in (29):

$$E^{tot}(r) = E^{inc}(r) + k_b^2 \int_D G\left(r - r'\right) \chi\left(r'\right) E^{tot}\left(r'\right) dr', \tag{29}$$

where k_b represents the wave number. In the two-dimensional case, Green's function of free space in cylindrical coordinates is denoted by $G(r - r')$. The contrast of permittivity is $\chi(r)$ and $E^{sca}(r^R)$ is the scattered field at the distance r^R. The scattered electric field can be

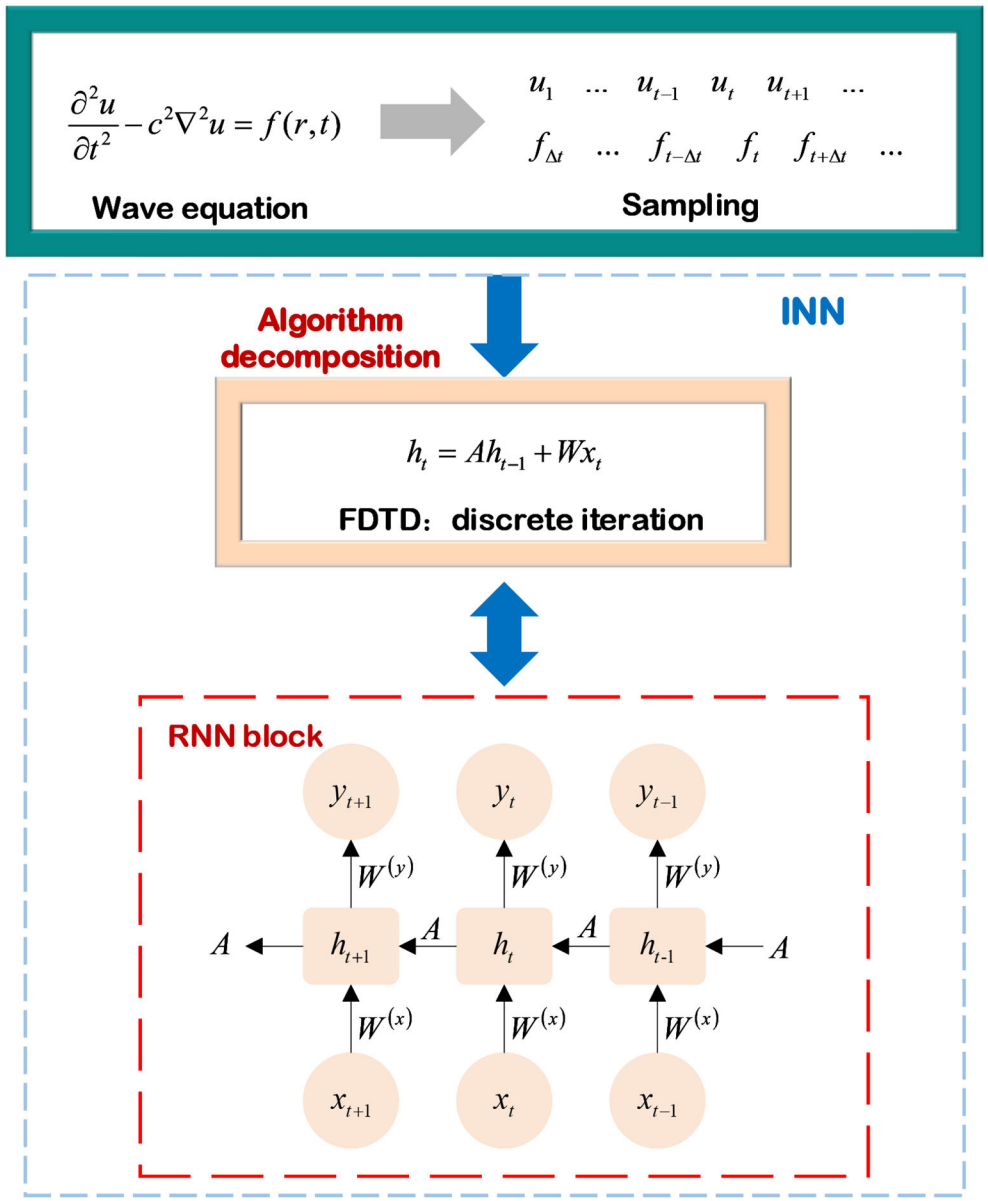


FIGURE 9
Wave equation prediction based on INN.

regarded as the secondary radiation of the induced current J, which can be written as

$$J(r) = \chi(r)E^{tot}(r). \tag{30}$$

In this subsection, the impulse function is used as the basis function, and the discretized matrix equation is constructed by the point test function. The scattered region D is divided into M sub-regions, and in the m-th subregion, its EFIE can be written as

$$E_m^{tot} + \frac{j}{4}k_b^2 \sum_{s=1}^{M} \chi E_s^{t,ot} \int_{D_S} H_0^{(2)}\left(k_b \left|r_m - r_s'\right|\right) dr_s' = E_m^{inc}. \tag{31}$$

Then the matrix equation for all regions of interest can be formulated as

$$(I + Z\chi)E^{tot} = E^{inc}. \tag{32}$$

The conjugate CG, which is a hybrid of the steepest descent algorithm and the Newton iterative approach, is used to solve the EFIE problem. Moreover, it is also one of the most efficient algorithms for addressing nonlinear optimization problems and solving sparse systems of linear equations. The CG method was first proposed by Hestenes and Stiefel (1952), and its essential point is that in the computation process, each search direction is conjugated to each other, and these search directions are calculated by the negative gradient and the search direction in the previous step. Wei and Chen (2019) replaced the process of updating the gradient

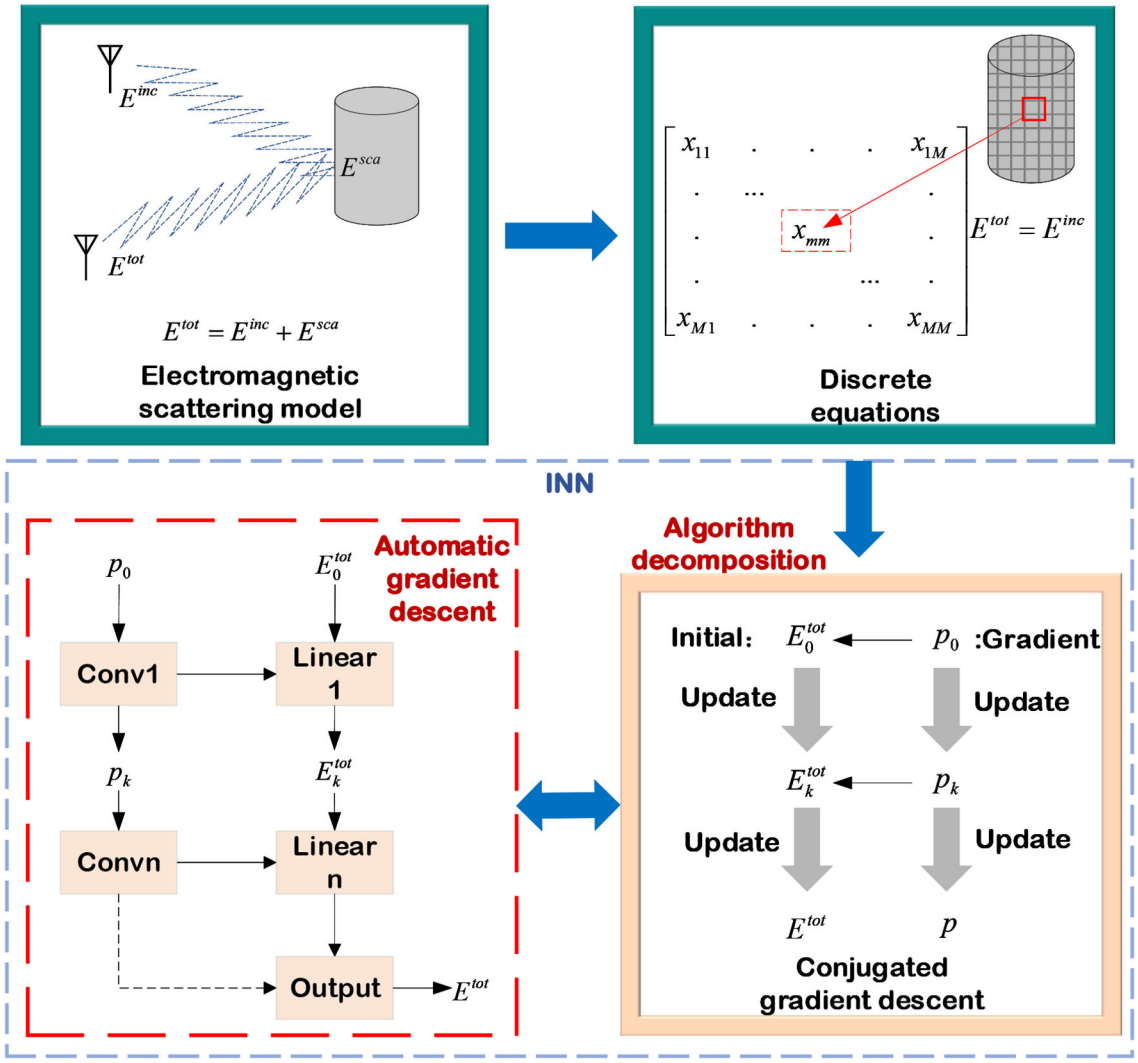


FIGURE 10
INN accelerating electromagnetic calculation.

direction and the total electric field E^{tot} with two cascaded NNs, respectively. In the first replacement, NN is used to predict the gradient direction of the next stage, and the step size and weight automatically assigned by the network are used for updating. The input of the network includes the residuals of the previous two moments, denoted by pr_k and r_{k-1}, and the gradient direction of the previous moment, denoted by p_k. Then, the process of updating the gradient direction can be consequently expressed as

$$p_{k+1} = f\left(p_k, r_k, r_{k-1}, \theta_k^p\right), \tag{33}$$

where θ_k^p is the weight in NNs. Similarly, the process of computing the total electric field based on the CG algorithm is replaced with several cascaded NNs, while the step size and weights are automatically updated by NNs' back-propagation. The process of finally computing the total electric field E^{tot} is shown as:

$$E_{k+1}^{tot} = E_k^{tot} + f_p\left(p_k, r_k, r_{k-1}, \theta_k^p\right), \tag{34}$$

where $f_p(\cdot)$ is the optimized NN of the total electric field. The NN replaces the standard gradient descent approach in the forward scattered field computation. In conventional electromagnetic calculation methods, each iterative update requires calculations of the gradient direction and selections of step size. Only by selecting the appropriate step size, the forward scattered field can be estimated quickly and accurately. In comparison to the pure data-driven technique, the whole network structure of theINN incorporates some of the theoretical information. As a result, it does not need a large amount of data to predict the mapping function, allowing the capacity to minimize data dependency.

2.2.3. Turbulent motion prediction

Predicting the direction and speed of turbulence has crucial uses. The motion of plasma turbulence, for example, can interfere with satellite operations and space communications in interplanetary space. The movement of atmospheric turbulence in the atmosphere influences the trajectories of tornadoes, tsunamis,

and cold waves. Therefore, being able to accurately predict the trajectory of turbulence is one of the most essential research in the field of fluid dynamics. According to the principle of Figure 3, turbulence motion prediction can be regarded as a task of solving PDEs. In the process of constructing PDEs, some physical constraints are drawn into account to control the multiple-order terms contained in the differential equations. Hence it is possible to solve this prediction by employing an INN that is the same as PDEs. This kind of physical model-decomposition INN for turbulence prediction is finally divided into the following four steps (Kochkov et al., 2021):

1. Constructing differential equations combined with physical constraints.

   According to dynamic principles, the velocity and trajectory of turbulent flow are only related to a few influencing factors. As a result, the general model of turbulence motion is sparse in space and can be composed of finite non-zero terms.
2. Time series discrete sampling

   The constructed turbulence model is discretely sampled according to Δt of each step, and the discrete model solved by the iterative approach can be split into smaller components.
3. Discrete turbulence motion model combined with NNs.

   The PDEs are decomposed into finite terms containing certain parameters and unknown equation terms that are related to known terms.
4. Training and testing the INN.

The decomposition of known and unknown terms from a turbulence model is a critical step in building an INN. It not only helps to reduce unknown parameters in NNs but also confines the convergence of the loss function. Furthermore, the model decomposition affects the ultimate accuracy and the difficulty of network training. The known terms in the turbulence prediction INN guarantee that the final prediction results are comparable in overall trend to the precise solution. Simultaneously, the unknown items related to the known items adjust the model at certain tiny values, allowing the final outcome to satisfy our expectations.

2.3. Other model-decomposition INN

There are many challenges in the biochemistry area right now that cannot be properly represented by a mathematical model, yet they nonetheless include a wealth of domain knowledge in processing. In this section, other models are used to generalize such issues, and all the processes of using domain knowledge to modify the input and output of DL are collectively referred to as other model-decomposition INNs. Therefore, these INNs focus on improving the front-end input data or correcting the terminal output results. Its interpretability is mainly realized in data processing and hyper-parameter configuration rather than in NN's layer design. Following the successful training of a "black box" NN, the analysis of interpreting the network structure, results, datasets, and so on is called "*post-hoc* interpretability," which means that *post-hoc* interpretability does not affect the NN before training. For example, in fluorescence image reconstruction, the probability and shape of the target appearing in a specific area are determined by a theoretical model, and these theoretical models constrain the final output through template matching. In ultrasound imaging, noise is mixed with the input signal. Using PCA to process the ultrasound will greatly improve imaging performance. For a complex value classification network, dealing with the real and imaginary parts of a complex separately cannot reflect the backward of complex values. Redefining the forward calculation and backward propagation of the complex value network makes the NN training more realistic and increases the input information.

3. Semantic INN

Semantic INN is the interpretable analysis method designed for the engineer. Its core concept is to explain the NNs from the standpoint that the engineer enables to see visually, analyze logically, and understand attributes. To this end, the first for the engineer who works on the semantic INN is to evaluate results visually and then analyze reasons logically, which are used to explain the reason in an accessible way. In this section, semantic INN designed for engineers is divided into three aspects which are vision, logic, and attributes, respectively. As shown in Figure 11, it illustrates the semantic INN structure and its interpretable regions based on the three aspects.

Semantic INN starts with the visualization of convolutional layers by plotting the heat map of each layer to reflect changes during network training. Then, combining decision trees and DL methods, logic calculations are drawn into the NNs so that there is certain logic information in the network layers, and explainable trees are extracted from NNs to explain the network structures. At the same time, there are also many studies directly starting from the attribute semantics of the target to build INNs.

3.1. Visualization of CNN

From a visual point of view, we hope to see the relationship between each output result of the network during the training process and the input data, especially in the classic image recognition classification problem, and analyze how the network recognizes the target from the input image. This method started with AlexNet visualizing the convolution kernel of the first layer, and then Zeiler and Fergus (2014) proposed a more explicit visualization method to comprehend the visualization results of convolutional layers, which was the pioneering work of visualization research. Moreover, numerous scholars who analyze and understand visualization results in image classification and recognition (Yosinski et al., 2015). The core concept of CNN visualization is to draw all the feature maps of each hidden layer in the CNN and examine the activation values of feature maps in the CNN. Finally, the visualization results are realized by extracting the convolution kernels from the pre-trained network, which is a process of deconvolution.

This section mainly discusses how to inversely map the feature map to the original pixel image, and comprehend the function between the feature map of each layer in CNN and the pixel image. Firstly, the process of convolution calculation in a pixel image can be divided into the following four steps:

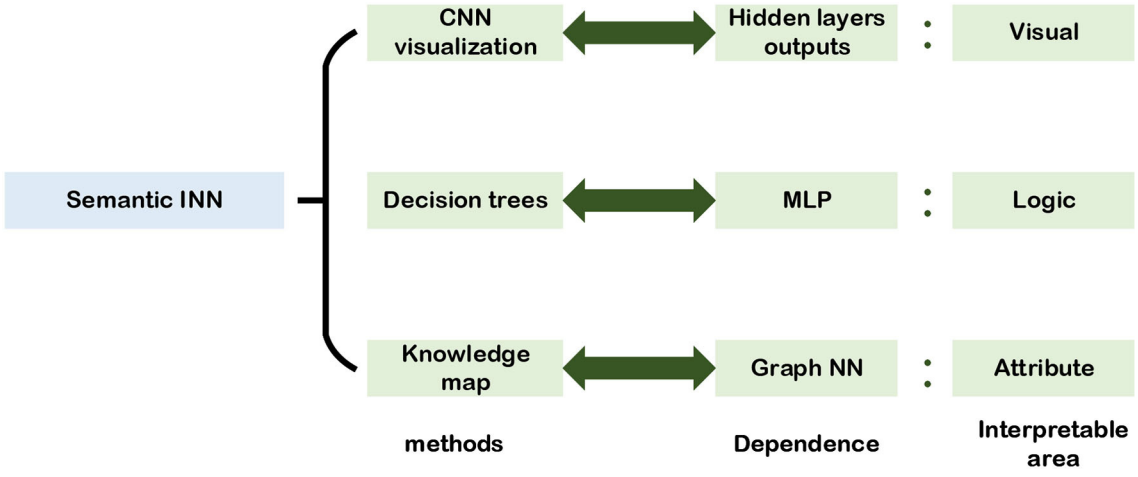


FIGURE 11
Semantic INN.

1. Convolution kernel
   The process of convolution may be thought of as an operation in the field of image filtering, and the size of the filter is proportional to the size of the convolution kernels.
2. Normalization
   Normalization is an equalization operation on each pixel of the feature map, and not all convolutional layers need to be normalized.
3. Activation function
   The activation function ensures the threshold of each feature map. Usually, the feature map of each layer is a positive value, and the Rectified Linear Unit (ReLU) activation function is widely used.
4. Pooling kernel
   Pooling reduces the size of the feature map in the previous step, which is an irreversible down-sampling process.

The fundamental aim of CNN visualization is to combine the feature maps from layers to analyze the influence on input. The feature map deconvolution procedure is the inverse of the pixel image convolution calculation. To correlate to the convolution, the deconvolution procedure is similarly separated into four steps (Zeiler and Fergus, 2014):

1. Up pooling
   The feature map needs to be up-sampled to the same size as the original image of the previous layer, and the up-sampling method is up-pooling. Its basic idea is to record the position of the maximum activation value of the pooling output in the original image, and then only activate this position, while the other positions are zero.
2. Activation function
   The pixel image still needs to keep the pixel value positive, and the activation function in the deconvolution can be consistent with the activation function in the convolution.
3. Denormalization.
   Denormalization is the equalization processing of the entire picture, which can be omitted or multiplied by a fixed intensity.

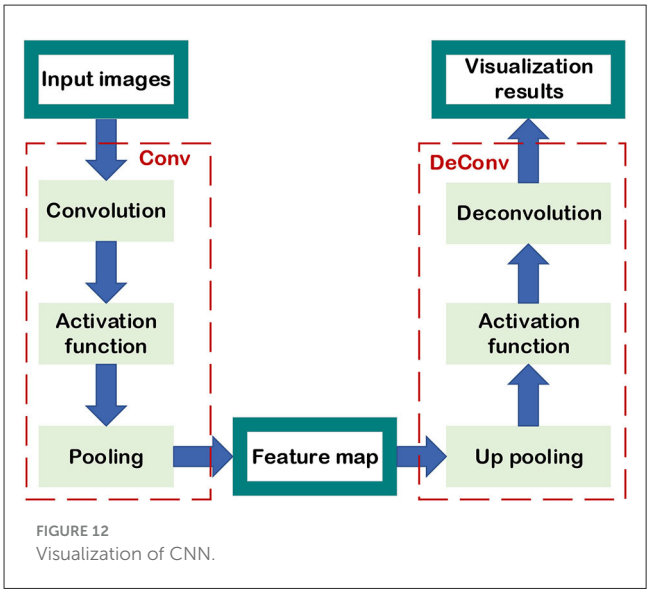


FIGURE 12
Visualization of CNN.

4. Deconvolution
   The deconvolution procedure is the core of CNN visualization, and it is also a filter. It can be achieved by multiplying the feature map with the transpose of the convolution kernel.

As shown in Figure 12, the flowchart of performing a convolution operation on a pixel image to obtain a feature map performing a deconvolution operation with the feature map to obtain an approximate pixel image is given. Through the deconvolution procedure, the information of the particular feature map corresponding to an input image can be visualized, which is used to analyze the different functions between low-level layers and high-level layers to extract pixel image characteristics. Yosinski et al. focused on CNN visualization and pointed out the difference in performance between shallow network and deep network in image feature extraction (Yosinski et al., 2015).

The convolution kernels of the lower layers can be drawn by flattening, and they extract the edge, color, and other macro characteristics of the pixel image. The contribution of shallow structures to the image recognition task can be represented by convolution kernel activated values. Meanwhile, the deep convolution kernel can extract more complex characteristics of the pixel image although the function of the deep structure cannot be judged directly from the convolution kernel shapes and the output of the convolution kernels. Hence, using CNN visualization is one of the most suitable methods to map the output of the deep layers' kernels into the original pixel image, which reflects the texture of the pixel image, and the deeper the feature map, the more specific features are extracted. Besides, it's known that convolution is proposed based on the translation and scaling invariance of the image. Owing to the linear transformation of the image, the edge, and color retrieved by the low-level network will change, while the abstract texture extracted by the high layers will not change. Finally, the visualization of CNN can be applied to not only illustrate the operations of the NN layers but also to verify the role of various convolution kernels in accomplishing tasks. It can further modify and improve the initial structure according to the outcomes of feature map visualization and increase the performance of the NN.

Recently, verification and validation (V&V) of NN is well accepted in the autonomous safety assessment. (Huang et al., 2020; Rajabli et al., 2020). The pipeline of the autonomy system is to use physical sensors to provide image information, and perceptrons to provide image interpretation. As clarified in the standards for autonomous systems (ANSI/UL 4600) (Koopman et al., 2019), when verifying the correctness of classifiers, the classification result can only be accepted if it has been obtained with the consistency of human expectations. Furthermore, the autonomy system needs to provide a mapping of the NN input of an ontology of the operational design domain in addition to the classification result. To this end, using the visualization of CNN to achieve V&V in the autonomy system is reasonable, and the NN will provide a transformation of the NN output of different layers to an activating value of the original images. For example, a pre-trained CNN presents the classification result of the autonomous system, and the CNN visualization approach is used to convert the investigated image to the correct ontology member.

3.2. Generation and extraction of decision trees

From the perspective of human reasoning, it's reasonable to combine the DL methods with decision trees to achieve semantic INN. The decision tree can assist engineers to perform classification tasks by utilizing the meaning of their nodes and edges. In particular, a decision tree contains parent nodes, child nodes, and top-down edges. A parent node can connect to two or more child nodes, and the message transfer on the adjacent edge can only be from the previous parent node to its child nodes. In other words, the decision tree is a top-down structure, which is widely used in classification and regression problems with supervised datasets. For a given dataset, the first step to extracting its decision tree is to encode the labels of targets. For instance, considering a multi-class classification shown in Figures 13, 14, the red color is coded as "0" while the white denotes "1." The sphere is coded as "3," and the cylinder is labeled as "4." Assuming that the decision of the first layer is color, and only the ball is red in the original dataset, the decision from the parent node to the red child node must be a red ball. Keep splitting down until all the decisions are made, and then a standard binary decision tree will be constructed consequently. There are many methods to establish decision trees, and their basic idea is to split nodes from top to bottom, such as CART, ID3, and C4.5 (Charbuty and Abdulazeez, 2021). Because of the logical interpretability of decision trees, it's a reasonable way to combine the DL methods with decision trees to achieve semantic INN, which is the main topic in this section.

The approach of using decision trees to handle data classification and regression issues is consistent with the agent's inductive interpretation and that the decision tree's hierarchical structure is similar to the network layers. Therefore, it is effective to improve the interpretability of NNs by combining the inductive judgment of decision trees. Frosst and Hinton (2017); Wu et al. (2018, 2020) propose a strategy for increasing the interpretability of the trainable network by adding decision tree regularization to the regularized network which can be divided into a global regularization, and a regional regularization network. The objective of adding decision trees is to constrain the network training, and local regularization can better adapt to data changes in data classification issues. The process of combining the decision tree to regularize the network is divided into the following four steps:

1. Data grouping.
   When grouping the input data, the linear segment can be used as a data grouping method.
2. Decision tree extracting.
   The ML method is used to classify each group of data, and construct a decision tree for each group of data.
3. Decision tree regularization network constructing.
   The decision tree is added to the network training as the regularization part of NNs, and the trainable network is constrained by the decision tree.
4. Training a semantic INN.

The decision tree isn't differentiable in step (3), which can't be immediately put into the network training as a regularization function. Hence, it is necessary to construct a map from the decision tree to the trainable structure.

This non-derivable decision process can be achieved by converting it into a layer of a linear transformation, as shown in Figure 13. The strategy adopted by Wu et al. was to convert the decisions in each layer of a decision tree into multi-layer perceptrons (MLPs) (Wu et al., 2018). They use the fully connected layer to realize each round decision, which means that the number of layers in the MLP is consistent with the depth of the decision tree. Then, using a pre-trained network, the decision tree is transformed to the MLP in which the corresponding relationship between the nodes is encoded in the activation values of feature maps. Consider an input image as a parent node, which should be divided into several child nodes at a given level. In the MLP, the comparable procedure is that several feature maps are generated in the input image via a fully connected layer, and these feature maps continue

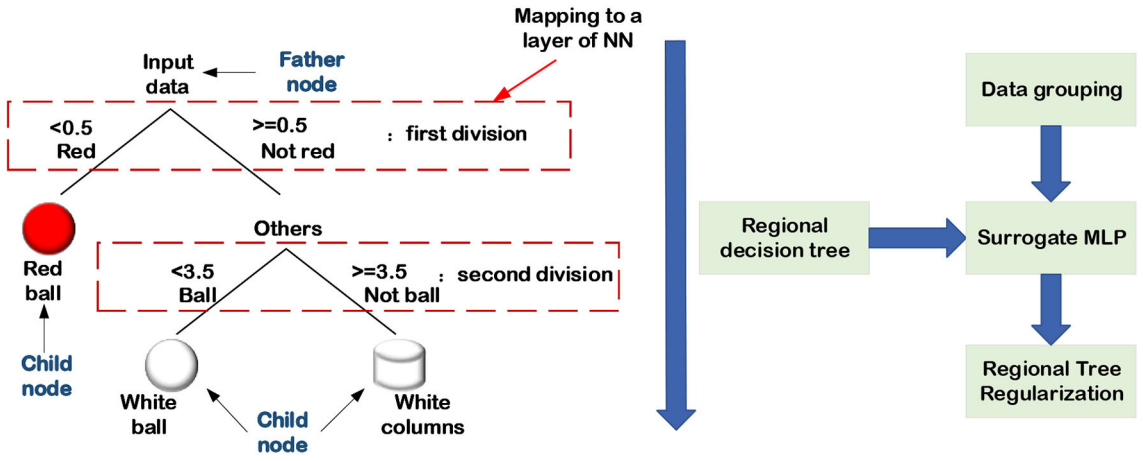


FIGURE 13
Regional decision trees generation.

to split downward as child nodes of the subsequent layer. In general, to employ decision tree regularization to achieve semantic INN combined with the logical level of the agent, it is important to define or train a regularization network in advance.

In addition to constructing the map of decision trees to trainable structures as aforementioned, there are scholars who establish the map from NNs to the explanation graph. Sun et al. (2020) demonstrate the way to use statistical fault location (SFL) techniques from software engineering to provide a high-quality interpretation of DNN's output and propose an algorithm and tool called DEEPCOVER. Their method uses SFL to synthesize a ranking of input features and constructs explanations of DNN decisions based on ranking. Zhang et al. (2017, 2019a) presented the bottom-up technique of using an explanation graph in combination with CNN visualization to extract the hidden semantics of pre-trained CNNs. The essential idea of this technique is to consider the activation peak value of each feature map as a child node, and network layer connections as the edges between the child node and the parent node. It consists mostly of the two phases listed below:

1. Initial decision tree

   Before explaining graph learning, the most crucial step is to initialize the number of activation peaks of each feature map in advance to form an initial decision tree. The shallower layers contain more activation peaks, whereas the deeper layers' feature maps have fewer activation peaks, indicating that several child nodes are linked to a parent node.
2. Fusion nodes

   The initial decision tree will be redundant, and the tree needs to be pruned. That is, the activation peaks that lead to the same result are fused into a single activation peak, and the resulting decision tree is the explaining graph of the pre-trained CNN, as illustrated in Figure 14.

Extracting an explanation graph from a pre-trained network is a *post-hoc* interpretability method that does not have any impact on the network structure.

3.3. Knowledge map aided zero-shot learning

This section merely considers the convolution and interpretability of the semantic map and briefly introduces the structure and optimization method of graph convolution network (GCN) in combination with node features and link properties. It starts with the description of the general explanation graph and subsequently presents the way of establishing the semantic INN to achieve classify task. Firstly, the explanation graph consists of nodes and adjacent edges, which are classified as directed or undirected graphs based on the properties of the adjacent edges. Secondly, according to the attributes contained in nodes, explanation graphs can be divided into probability graphs and semantic graphs. In the probability graph, each node represents the probability of a presented attribute, and the connected nodes represent a joint probability distribution between two different attributes. In the semantic graph, each node represents a feature vector or a kind of semantic message, and the connected nodes indicate that two features or semantics are available in the whole graph at the same time.

Since the semantic message can be a dense matrix or a sentence, it's crucial to convert this kind of semantic message into a feature vector. For the convenience to separate the grid pixel image from the undirected graph composed of edge nodes, the explanation graph in the non-Euclidean space is collectively named the edge-node graph. Considering an image classification task based on edge-node GCN architecture, the input of GCN is usually a form of word embedding (WE), as engineers always utilize a set of words or phrases to describe the attributes of the feature. The operation of the WE is to convert the words or phrases that are semantic messages of the image into a set of feature vectors and these feature vectors consist of the semantic space. Furthermore, the semantic space contains two categories based on the methods used to construct semantic space: engineering semantic space and learning semantic space. The engineering semantic space is artificially designed by engineers. They describe the target in a

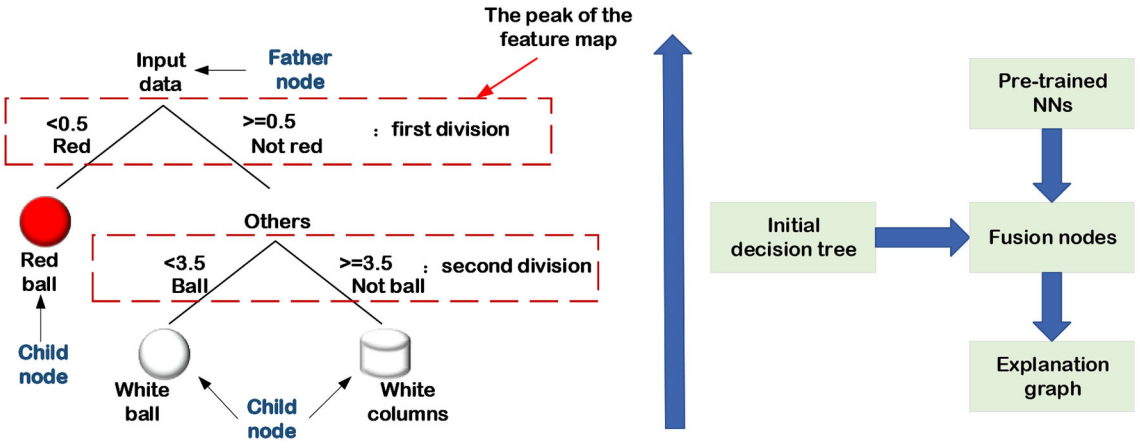


FIGURE 14
Explanation graph extraction.

unified form based on domain expertise, with multiple meanings for each dimension in the semantic space. However, there are several approaches to constructing the engineering semantic space, and the most common method is the attribute semantic space (Lampert et al., 2009; Palatucci et al., 2009). For example, we can design a simple attribute semantic space for a tiger that consists of ears, tails, fur, and forehead lines. Similarly, the animal "cat" can be described by these attributes without forehead lines. Then, the different attributes of an object can be converted into a vector, where the object prototype has this attribute marked as 1, otherwise marked as 0, as shown in Figure 15. On the contrary, there is also learned semantic space that is obtained by ML methods, that is, it's unnecessary for engineers to manually designate features. However, the semantic vector obtained by ML is no longer interpretable and becomes abstract and incomprehensible to humans.

Following the WE, the next step is to perform convolution on the edge-node graph. Since the manifold space of the edge-node graph distribution does not belong to the Euclidean space, the convolution operation in GCN is very different from that in the pixel graph. The GCN is therefore turned into an operation between the feature vectors in semantic space and the adjacency matrix of the edge-node graph, as illustrated in Equation (35) (Kipf and Welling, 2016).

$$H^{l+1} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{l}W^{l}), \tag{35}$$

where $\sigma(\cdot)$ is the activation function, H^l and H^{l+1} represent the feature vectors of the l-th level and the $l+1$ level, respectively. The formula above indicates the message passing of nodes between every two layers of an edge-node graph in multi-layers GCN. Besides, the trainable weight matrix W is used to control the intensity of nodes in each of the two layers, and $\widetilde{D}$ is the degree matrix of the special adjacency matrix $\widetilde{A}$ which can be expressed as

$$\widetilde{A} = A + I. \tag{36}$$

The value in the adjacency matrix A indicates whether there is a link between any two nodes in the edge-node graph. Considering the case of the first-order neighbors of one node, the linked nodes are labeled as 1, while the disconnected nodes are marked as 0. Since there must be two nodes connected to the same edge, the adjacency matrix is also a symmetric matrix. The degree matrix D is a diagonal matrix, and the values on the diagonal represent the degree of each node, which is determined by the number of edges linking the node. Obviously, the GCN is the process of message passing, and the upper layer feature vector H^l exchanges the message with the deeper layer feature vector H^{l+1} via the adjacency matrix A, where the message is transmitted between specific nodes. To incorporate the node's effect on message transmission, the adjacency matrix is transformed into the form specified by Equation (36). Simultaneously, the trainable weight matrix W is employed to manage the process of message transit between two nodes. Following GCN's message traveling through all nodes, the resulting feature vector is denoted as $\widetilde{A}H^lW^l$. Unfortunately, the GCN faces several limitations when it comes to multi-layer transformation. As the number of convolutional layers grows, the output value increases rapidly as well. Thus, in GCN, in addition to the convolution operation, an activation function consistent with DNN is required, and the next level feature vector is denoted by (35). After training the GCN, the linear transformation matrix between layers tends toward a stable value, as shown in Figure 16.

Figure 16 shows the message transmission process of node "1." Its input graph contains 5 nodes and 5 adjacent edges. Nodes "1" and "4" have three edges each, node "3" has two edges, and nodes "2" and "5" have only one edge each. After the GCN operates on the input graph, the resulting graph retains its structure as the input graph, but the feature vectors and the weight matrix corresponding to the information transmission on the edges have been altered. Considering the input graph given in Figure 16, the adjacency matrices A and $\widetilde{A}$ of the given edge-node graph, as well as the degree matrix $\widetilde{D}$, are represented in Table 2. Using the matrix parameters in Table 2, the resulting feature vector after one time of message transmission can be calculated.

GCN is commonly utilized in social networks, molecular investigation, and natural language processing (NLP). At the moment, the zero-shot learning (ZSL) target classification

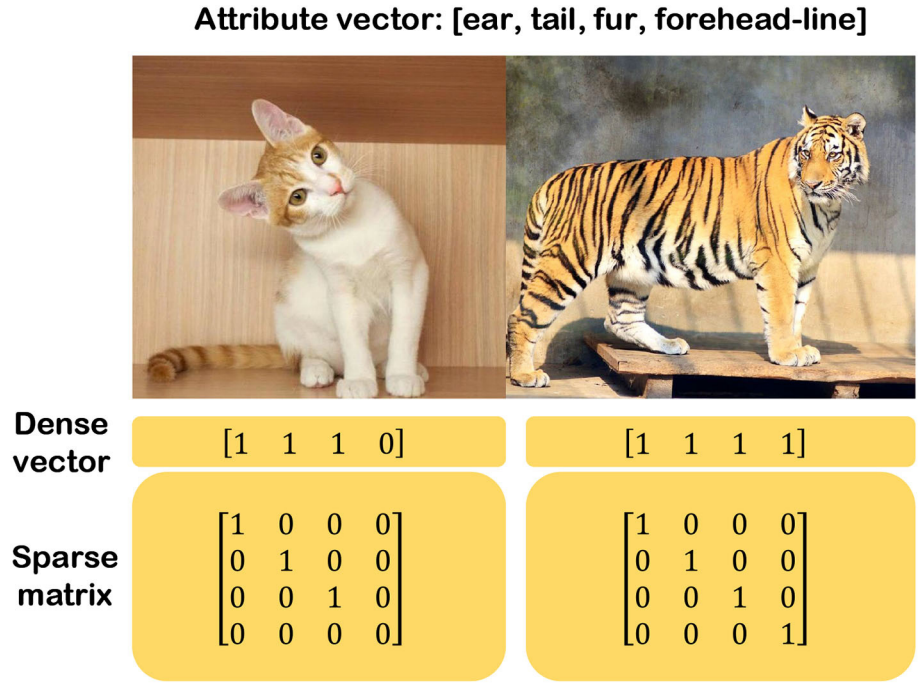


FIGURE 15
Binary attribute space.

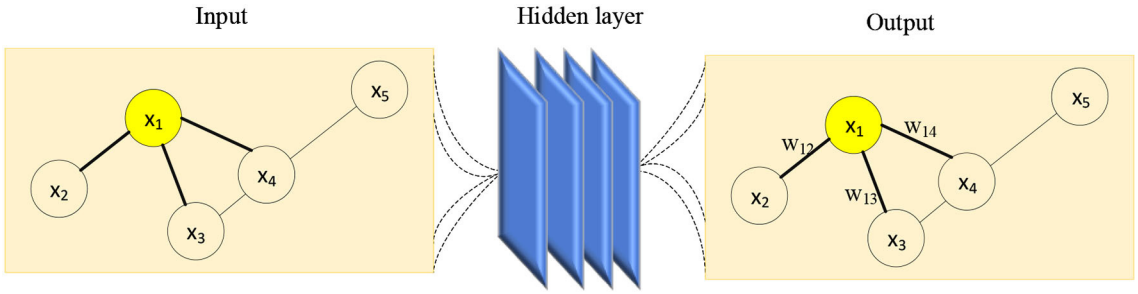


FIGURE 16
The process of message passing in GCN.

technique for pixel images that combines CNN and GCN is still under development. In this, ZSL is to solve the classification problem of image objects without the training data of available classes and only provide the description of classes. In addition, it requires computers to be capable to distinguish new objects by learning the way of humans reason without ever seeing their categories. Its fundamental concept is to utilize a pre-trained CNN to extract features from pixel images, then remove the final classification layer and replace it with a GCN to accomplish target classification, as illustrated in Figure 17.

Before training a GCN, the task-based edge node graph structure to be learned must be manually designed. The GCN is then trained in a supervised way to generate the classifier weight matrix, which will be used to replace the classifier in this classification task. In order to add the interpretability of semantic space, Wang et al. (2018b) add the KG to the GCN and combine the semantic attribute space of the edge node graph with the inference described in the KG to accomplish ZSL for unknown category targets.

TABLE 2 Adjacency matrix and degree matrix examples.

A	$\tilde{A}$	$\tilde{D}$
$\begin{bmatrix} 0&1&1&1&0 \\ 1&0&0&0&0 \\ 1&0&0&1&0 \\ 1&0&1&0&1 \\ 0&0&0&1&0 \end{bmatrix}$	$\begin{bmatrix} 1&1&1&1&0 \\ 1&1&0&0&0 \\ 1&0&1&1&0 \\ 1&0&1&1&1 \\ 0&0&0&1&1 \end{bmatrix}$	$\begin{bmatrix} 4&0&0&0&0 \\ 0&2&0&0&0 \\ 0&0&3&0&0 \\ 0&0&0&4&0 \\ 0&0&0&0&2 \end{bmatrix}$

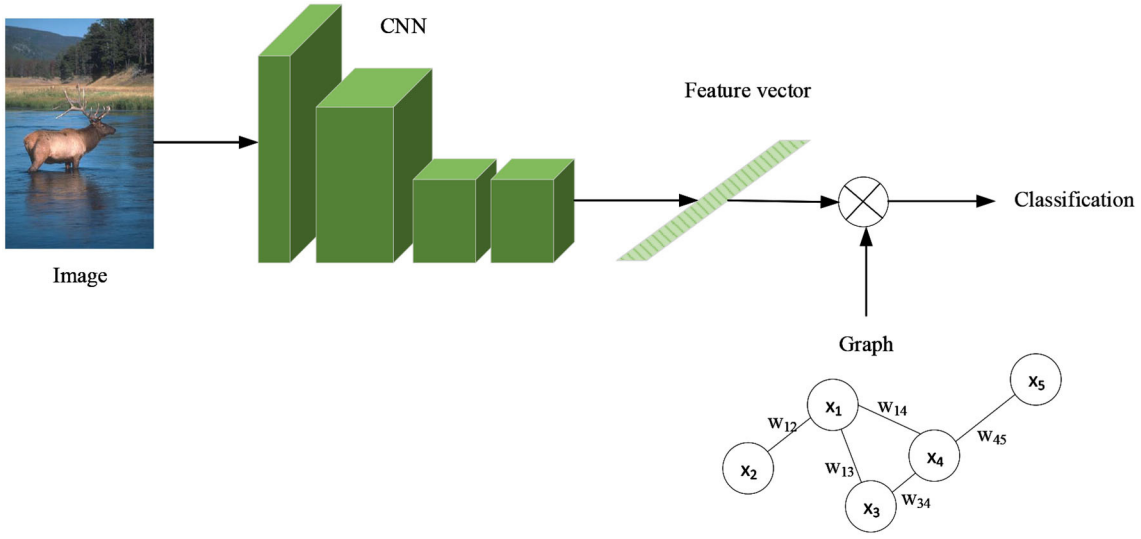


FIGURE 17
CNN+GCN zero-shot learning solves target classification.

4. Electromagnetic neural networks

There has been considerable research on INN in the field of electromagnetic physics, with the goal of balancing the benefits of classical electromagnetic computing algorithms with DL approaches. However, none of their suggested solutions satisfies a real physical problem. Our objective is to apply the previously mentioned physical model-decomposition INN to solve real-world physical issues. This section introduces and defines the electromagnetic neural network (EMNN), outlines our technique for accomplishing actual electromagnetic physics issues, and describes how the EMNN handles forward and inverse electromagnetic problems.

4.1. Demand and challenge of EMNN

In recent decades, researchers have accomplished the forward and inverse electromagnetic tasks by constructing electromagnetic theoretical models. And, their common requirements and challenges are high computational complexity and slow speed. To overcome these obstacles, DL methods have been gradually used, and they are first used in optical images and then transferred to microwave images. However, the image processing algorithms in the electromagnetic field are very different from those in the optical field because of their different frequency properties, as shown in Figure 18. Additionally, it is challenging to obtain high-quality microwave pictures of objects, and the time cost of data acquisition will be greater than for optical images. Therefore, we propose EMNN to address these issues by embedding electromagnetic scattering models within NNs. Finally, our aim is to achieve fast computation, low complexity, high generalization, and interpretability in EMNN. Furthermore, the NNs can be used to accelerate electromagnetic calculations, and the electromagnetic scattering model is used to enhance the generalization of NNs. As a result, EMNN exhibits the result of rigorous logical reasoning and interpretability.

4.2. Definition of EMNN

In comparison to the processing methods of optical images, the electromagnetic neuron theoretical model based on microwave images is developed, which incorporates four critical electromagnetic properties of time, frequency, phase, and polarization. Time is utilized to indicate the time delay T of the echo, and different time delays are used to represent the different relative positions of the neuron in space. The relative positions in the multi-dimensional neuron theoretical model can help to relieve the signal oscillation. Frequency-phase-polarization is the element of the signal emission model that describe the electromagnetic properties of the emitted wave, including the signal frequency f, the initial phase P, and the polarization direction p. The specific form is as follows:

$$E_{neurons} = [T, P, f, p]. \tag{37}$$

Multidimensional neurons regarded as an observation matrix can depict a complex electromagnetic environment, which implies that the transmitter at distinct positions can release polarized electromagnetic waves with a special frequency and phase in space. Similarly, the targets in the scene with varying positions, sizes, and materials in the space will stimulate varying responses in this electromagnetic environment. That is, the echo received by the radar contains the electromagnetic scattering of all the targets in the scene, which can be considered as the measurement matrix of the radar.

Furthermore, the growth of neuron models with electromagnetic characteristics involves the implementation of a novel neural information flow transmission method. In

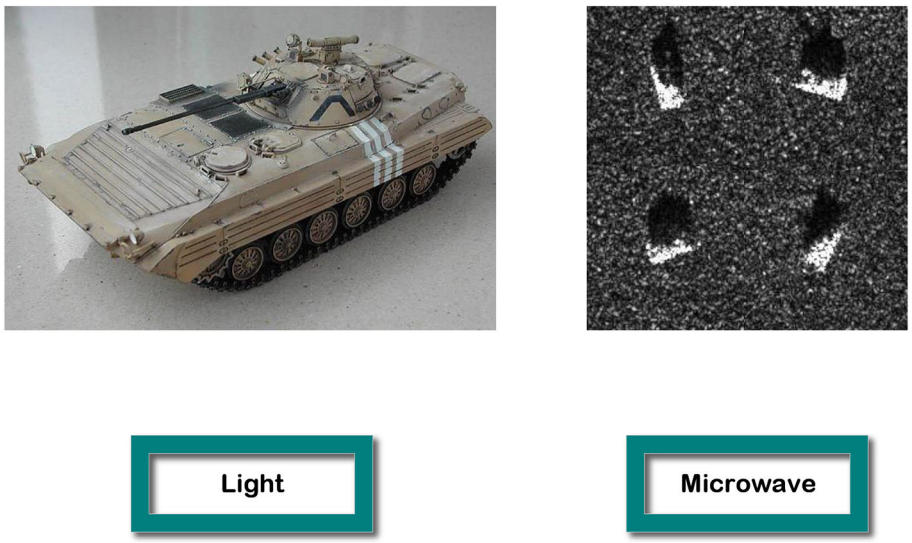


FIGURE 18
The first picture is the optical image of a tank, and the second is four SAR images of a tank in different orientations (Keydel et al., 1996).

comparison to the propagation of electromagnetic waves in free space, the wave function is the fundamental element in the NN's forward propagation process, so Green's function in free space is regarded as the basic solution of the EMNN. By combining it with the expression form of Green's function, the expression of EMNN forward propagation and the gradient descent technique of backward updating can be redefined. Especially compared to standard DL approaches, EMNN can obtain desirable electromagnetic fields or radiation patterns at a faster speed with fewer data.

In this case, the operations of the network layers should correspond to the calculations of the electromagnetic models. For a stricter EMNN, the training parameters in the network correspond to physical properties in the electromagnetic theoretical models. Then, the next step is to decompose the electromagnetic computational algorithm of this problem into iterative steps, which are converted into layers of NNs. Basically, the general electromagnetic model formula is presented as $f(s, \hat{x}_i, \hat{x}_j)$ which is exactly known, but some parameters of this formula are unknown. Then, the iterative steps $F_k(s, \hat{x}_i, \hat{x}_j)$ by solving this EM problem need to be estimated because of the parameters' indeterminacy. These estimated steps can be carried out iteratively by M times of addition, or iteratively by M times of multiplication, and the specific expressions are shown as:

$$f(s, \hat{x}_i, \hat{x}_j) = \sum_{k=1}^{M} F_k(s, \hat{x}_i, \hat{x}_j), \tag{38}$$

$$f(s, \hat{x}_i, \hat{x}_j) = \prod_{k=1}^{M} F_k(s, \hat{x}_i, \hat{x}_j). \tag{39}$$

In the EMNN, it assumes that some iterative steps $F_{k_g}(s, \hat{x}_i, \hat{x}_j)$ are known and others $F_{k_n}(s, \hat{x}_i, \hat{x}_j)$ are unknown, which are related to their front steps $F_{k_p}(s, \hat{x}_i, \hat{x}_j)$. To replace the unknown component of the solution, a special NN is manually designed to fulfill the mapping between the two continuous steps or some layers that can estimate the appropriate physical characteristics. They are written as

$$F_{k_n}\left(s, \hat{x}_i, \hat{x}_j\right) = \sum_{p=1}^{M_p} N_p\left(F_{k_p}\left(s, \hat{x}_i, \hat{x}_j\right)\right), \tag{40}$$

$$F_{k_n}\left(s, \hat{x}_i, \hat{x}_j\right) = \prod_{p=1}^{M_p} N_p\left(F_{k_p}\left(s, \hat{x}_i, \hat{x}_j\right)\right), \tag{41}$$

where M_p is the number of front modules, N_p is a NN whose mathematical meaning is the mapping between the front part $F_{k_p}(s, \hat{x}_i, \hat{x}_j)$ and the unknown part $F_{k_n}(s, \hat{x}_i, \hat{x}_j)$. It's obvious that getting precise front parts is vital for estimating unknown parts. Then, the unknown modules and known parts are restored to the EMNN basic expression, and through numerous iterations, the iterative algorithm to solve the EMNN problems is derived as follows:

$$f\left(s, \hat{x}_i, \hat{x}_j\right) = \sum_{g=1}^{M_g} F_{k_g}\left(s, \hat{x}_i, \hat{x}_j\right) + \sum_{n=1}^{M_n}\sum_{p=1}^{M_p} N_p\left(F_{k_p}\left(s, \hat{x}_i, \hat{x}_j\right)\right), \tag{42}$$

$$f\left(s, \hat{x}_i, \hat{x}_j\right) = \prod_{g=1}^{M_g} F_{k_g}\left(s, \hat{x}_i, \hat{x}_j\right) \prod_{n=1}^{M_n}\prod_{p=1}^{M_p} N_p\left(F_{k_p}\left(s, \hat{x}_i, \hat{x}_j\right)\right). \tag{43}$$

The formulae above provide the final expressions for iterative addition and iterative multiplication algorithms, respectively. If the final decomposed front module M_p and the unknown item M_n have a single item, the EMNN model defined by the iterative addition algorithm can be condensed into the residual form, and the iterative multiplication procedures can be turned into linear equations.

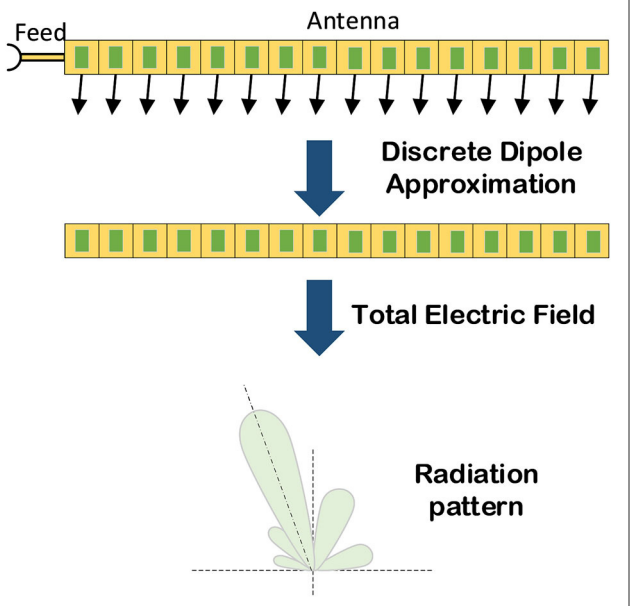


FIGURE 19
The pipeline of calculating coding antenna array radiation pattern prediction.

4.3. Applications of EMNN

The EMNN model is proposed to handle actual forward and inverse electromagnetic issues, and it is appropriate for processing electromagnetic signals because of its robustness, speed, and interpretability (Li et al., 2022b; Liu and Xu, 2022; Zhang et al., 2022). This section will explain how to set up an EMNN to handle the problem of positive radiation pattern prediction using coding antennas.

4.3.1. Coding antennas array radiation pattern prediction based on EMNN

The first step of handling the coding antenna array radiation pattern prediction (CARP) problem is to set up the EMNN model, and the process of accomplishing CARP is shown in Figure 19. In addition, the radiation pattern may be described as the multiplicative EMNN model using the discrete dipole approximation (DDA) method. Then it can be consequently reduced to a linear model because there is only one unknown component, which can be deduced as Liu et al. (2021); Li et al. (2022a):

$$E^{tot} = BAH^{inc}. \tag{44}$$

where B is replaced by the known part $F_g(\theta, x_i, x_j)$, and H_{inc} is replaced by the front part. Then, the transferred matrix A, which is the coupling effect between each two antenna elements can be estimated by NNs. Based on the EMNN model, this CARP problem is formulated as

$$f\left(\theta, x_i, x_j\right) = F_g\left(\theta, x_i, x_j\right) F_n\left(\theta, x_i, x_j\right). \tag{45}$$

In this model, the calculation process of solving the total field is replaced by NN layers, and the expression of the final radiation pattern prediction based on EMNN is shown in Equation (46). This means that it needs to get the incident field as input, and then use some fully connected layers to obtain the total electric field. Finally, the total electric field calculated by the NN is sent to the DDA model to calculate the antenna radiation pattern. The calculation diagram of the EMNN is illustrated in Figure 20.

$$E^{\text{tot}}\left(\theta, x_i, x_j\right) = BN_p\left(H^{\text{inc}}\left(\theta, x_i, x_j\right)\right). \tag{46}$$

5. Discussion

In this review, we introduce how to build the EMNN and provide applications for using INNs to solve real-world physical problems. To begin, this paper discusses the limitations of DL methods and model-based techniques in order to demonstrate the significance and necessity of the emergence of INN. Then, the INN is described in two parts, the model decomposition alternative INN and the semantic INN. The former is to explain the traditional models into NNs, which is achieved by transferring reality constraints and formula constraints into layers of NNs. The latter is mainly the "interpretation" of the agent, which builds and analyzes NNs based on semantic features such as vision, logic, and attributes. Finally, considering electromagnetic problems, this paper introduces how to convert the parameters in the electromagnetic model into the NNs' parameters in detail. Below, the strengths, limitations, and prospects of INNs are discussed.

5.1. The strengths of INNs

Nowadays, INNs still do not guarantee the reliability of specific tasks, but they facilitate the determination of reliability in the following two perspectives. Firstly, from the perspective of constructing model decomposition alternative INN, combining the traditional mathematical-physical model to build an INN can reduce the network parameters and the network layer design. Secondly, from the perspective of semantic INN, the realization of a posteriori INN after pre-training can help engineers correct errors, which means that engineers can find out from the explanation graph where the network operates in classified tasks or other tasks. To sum up, we compare the INNs with prior DL approaches:

1. Generalizability
   INNs can extract information from the theoretical model of the issue or from the laws governing objective facts and incorporate it into the network's architecture. Then, data independence and the generalization performance of INN are better than those of traditional methods.
2. Trustworthiness
   The black box architecture will not inspire trust, but the semantic INN can display the layers and feature maps of the NNs. INN inductively obtains hidden information from NNs and portrays it as a decision tree. Incorporating visual, logical, and semantic descriptions of the agent's attributes into the decision tree aids in the comprehension of how the

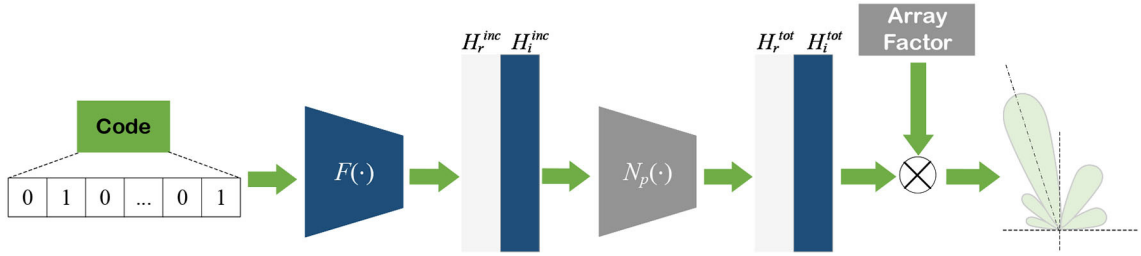


FIGURE 20
The EMNN structure of solving CARP problem.

network operates. Therefore, the trustworthiness of INNs can be enhanced.

3. Interpretability

   Either alternative INNs based on model decomposition or semantic INNs, both emphasize "interpretability." The former interprets the theoretical model as a NN, while the latter interprets the NN as a semantic model.

5.2. The limitations of INNs

INNs also have some shortcomings, which are closely related to the way they are constructed. Here, the limitations of INNs are divided into the following three points:

1. Model Limitations

   Model decomposition alternative INNs are useful for handling linear problems. When decomposing a theoretical model, if it is linear, it implies that the network generated by the model is also linear. For those issues that cannot be immediately reduced to a linear model, they cannot be transferred into model decomposition alternative INNs.
2. Semantic library limitations

   Implementing an INN by extracting or constructing decision trees can only be applied to relatively common issues and tasks that can disentangle between nodes. And, in order to effectively use the semantic information in the network, it is required to build a massive semantic library, which demands a significant amount of personnel to manually design an expert system.
3. Interpretable Definition

   It's unknown whether the layers or elements can be assigned to physical facts and semantics one-to-one. Furthermore, not all intermediary portions are confirmed using ground truth, making the evaluation of the network's interpretable parameters unfeasible.

5.3. The development prospects of INNs

Currently, there is no strict definition of INNs. In this paper, a novel definition of INN is proposed based on a summary of the current research on INNs. A consistent and unambiguous definition may emerge in the future, and the process of creating an INN will be developed progressively. Based on the strengths and limitations of INNs, some future directions are discussed and suggested.

1. Model function expansion

   Constructing a theoretical model that can be represented uniformly. The function of the layer in the NN meets the requirements of the theoretical model calculation while their parameters are unequal.
2. Nonlinear problems expansion

   To decompose the nonlinear issues, this expansion approach starts with standard methods used to address linear problems. The nonlinear model is reduced to these linear formulas that may be substituted by NNs.
3. Semantic extraction expansion

   NNs can automatically extract semantic information from images and generate a semantic library, and the labels associated with these semantic libraries are all visual images that are directly tied to the decision tree. Applying this form of semantic information to INNs will help experts establish a semantic library.

Author contributions

ZL conceived the study and wrote the manuscript with support from the supervisor FX. All authors contributed to all aspects of the preparation and approved the submitted version.

Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under Project U2130202.

Acknowledgments

The authors wish to acknowledge that this work was supported by the Key Lab of Information Science of Electromagnetic Waves.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Akula, A. R., Liu, C., Saba-Sadiya, S., Lu, H., Todorovic, S., Chai, J. Y., et al. (2019). X-tom: Explaining with theory-of-mind for gaining justified human trust. *arXiv preprint arXiv:1909.06907*.

Beck, A., and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci* 2, 183–202. doi: 10.1137/080716542

Belthangady, C., and Royer, L. A. (2019). Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nat. Methods* 16, 1215–1225. doi: 10.1038/s41592-019-0458-z

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* 3, 1–122. doi: 10.1561/2200000016

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.* 113, 3932–3937. doi: 10.1073/pnas.1517384113

Burger, H. C., Schuler, C. J., and Harmeling, S. (2012). "Image denoising: Can plain neural networks compete with bm3d?" in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 2392–2399. doi: 10.1109/CVPR.2012.6247952

Charbuty, B., and Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *J. Appl. Sci. Technol. Trends* 2, 20–28. doi: 10.38094/jastt20165

Chen, S., Eldar, Y. C., and Zhao, L. (2021). Graph unrolling networks: Interpretable neural networks for graph signal denoising. *IEEE Trans. Signal Proc.* 69, 3699–3713. doi: 10.1109/TSP.2021.3087905

Chen, Z.-M., Wei, X.-S., Wang, P., and Guo, Y. (2019). "Multi-label image recognition with graph convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 5177–5186. doi: 10.1109/CVPR.2019.00532

Chien, J.-T., and Lee, C.-H. (2017). Deep unfolding for topic models. *IEEE Trans. Patt. Analy. Mach. Intell.* 40, 318–331. doi: 10.1109/TPAMI.2017.2677439

Clark, H. H. (1969). Linguistic processes in deductive reasoning. *Psychol. Rev.* 76, 387. doi: 10.1037/h0027578

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2011). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Proc.* 20, 30–42. doi: 10.1109/TASL.2011.2134090

Danielyan, A., Katkovnik, V., and Egiazarian, K. (2011). Bm3d frames and variational image deblurring. *IEEE Trans. Image Proc.* 21, 1715–1728. doi: 10.1109/TIP.2011.2176954

Daubechies, I., Defrise, M., and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Mathem.* 57, 1413–1457. doi: 10.1002/cpa.20042

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 248–255. doi: 10.1109/CVPR.2009.5206848

Fan, F.-L., Xiong, J., Li, M., and Wang, G. (2021). On interpretability of artificial neural networks: A survey. *IEEE Trans. Radiat. Plasma Med. Sci.* 5, 741–760. doi: 10.1109/TRPMS.2021.3066428

Fang, Y., Wu, G.-Z., Wang, Y.-Y., and Dai, C.-Q. (2021). Data-driven femtosecond optical soliton excitations and parameters discovery of the high-order nlse using the pinn. *Nonl. Dynam.* 105, 603–616. doi: 10.1007/s11071-021-06550-9

Frosst, N., and Hinton, G. (2017). Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*.

George, D., Lehrach, W., Kansky, K., Lázaro-Gredilla, M., Laan, C., Marthi, B., et al. (2017). A generative vision model that trains with high data efficiency and breaks text-based captchas. *Science* 358, eaag2612. doi: 10.1126/science.aag2612

Geyer, C. J. (1992). Practical markov chain monte carlo. *Stat. Sci.* 473–483. doi: 10.1214/ss/1177011137

Goswami, U. (2011). "Inductive and deductive reasoning," in *The Wiley-Blackwell Handbook of Childhood Cognitive Development* (Wiley-Blackwell), 399–419. doi: 10.1002/9781444325485.ch15

Gregor, K., and LeCun, Y. (2010). "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10* (Madison, WI, USA: Omnipress), 399–406.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Comput. Surv.* 51, 1–42. doi: 10.1145/3236009

Guo, R., Shan, T., Song, X., Li, M., Yang, F., Xu, S., et al. (2021). Physics embedded deep neural network for solving volume integral equation: 2d case. *IEEE Trans. Anten. Propag.* 70, 6135–6147. doi: 10.1109/TAP.2021.3070152

Hager, W. W., and Zhang, H. (2006). A survey of nonlinear conjugate gradient methods. *Pacif. J. Optim.* 2, 35–58.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision* 1026–1034. doi: 10.1109/ICCV.2015.123

Heit, E. (2000). Properties of inductive reasoning. *Psychon. Bull. Rev.* 7, 569–592. doi: 10.3758/BF03212996

Hestenes, M. R., and Stiefel, E. (1952). Methods of conjugate gradients for solving. *J. Res. Natl. Bureau Stand.* 49, 409. doi: 10.6028/jres.049.044

Huang, X., Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E., et al. (2020). A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Comput. Sci. Rev.* 37, 100270. doi: 10.1016/j.cosrev.2020.100270

Hughes, T. W., Williamson, I. A., Minkov, M., and Fan, S. (2019). Wave physics as an analog recurrent neural network. *Sci. Adv.* 5, eaay6946. doi: 10.1126/sciadv.aay6946

Johnson-Laird, P. N. (1999). Deductive reasoning. *Ann. Rev. Psychol.* 50, 109–135. doi: 10.1146/annurev.psych.50.1.109

Keydel, E. R., Lee, S. W., and Moore, J. T. (1996). Mstar extended operating conditions: A tutorial. *Algor. Synthet. Apert. Radar. Imag.* 2757, 228–242.

Kipf, T. N., and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., and Hoyer, S. (2021). Machine learning-accelerated computational fluid dynamics. *Proc. Natl. Acad. Sci.* 118, e2101784118. doi: 10.1073/pnas.2101784118

Koopman, P., Ferrell, U., Fratrik, F., and Wagner, M. (2019). "A safety standard approach for fully autonomous vehicles," in *International Conference on Computer Safety, Reliability, and Security* (Springer), 326–332. doi: 10.1007/978-3-030-26250-1_26

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., and Matas, J. (2018). "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 8183–8192. doi: 10.1109/CVPR.2018.00854

Lampert, C. H., Nickisch, H., and Harmeling, S. (2009). "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 951–958. doi: 10.1109/CVPR.2009.5206594

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nat. Commun.* 10, 1–8. doi: 10.1038/s41467-019-08987-4

Li, L., Wang, L. G., Teixeira, F. L., Liu, C., Nehorai, A., and Cui, T. J. (2018). Deepnis: Deep neural network for nonlinear electromagnetic inverse scattering. *IEEE Trans. Ant. Propag.* 67, 1819–1825. doi: 10.1109/TAP.2018.2885437

Li, S., Liu, Z., Fu, S., Wang, Y., and Xu, F. (2022a). Intelligent beamforming via physics-inspired neural networks on programmable metasurface. *IEEE Trans. Ant. Propag.* 70, 4589–4599. doi: 10.1109/TAP.2022.3140891

Li, S., Liu, Z., Wang, Y., and Xu, F. (2022b). "Programmable metasurface intelligent beamforming," in *2022 16th European Conference on Antennas and Propagation (EuCAP)* (IEEE), 1–3. doi: 10.23919/EuCAP53622.2022.9769249

Li, X., Zhang, G., Qiao, H., Bao, F., Deng, Y., Wu, J., et al. (2021). Unsupervised content-preserving transformation for optical microscopy. *Light: Sci. Applic.* 10, 1–11. doi: 10.1038/s41377-021-00484-y

Li, Y., Tofighi, M., Geng, J., Monga, V., and Eldar, Y. C. (2020). Efficient and interpretable deep blind image deblurring via algorithm unrolling. *IEEE Trans. Comput. Imag.* 6, 666–681. doi: 10.1109/TCI.2020.2964202

Liu, Y., and Storey, C. (1991). Efficient generalized conjugate gradient algorithms, part 1: theory. *J. Optim. Theory Applic.* 69, 129–137. doi: 10.1007/BF00940464

Liu, Z., Li, S., and Xu, F. (2021). "Coded antenna radiation pattern prediction network based on dda algorithm," in *2021 XXXIVth General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)* 1–4. doi: 10.23919/URSIGASS51995.2021.9560632

Liu, Z., and Xu, F. (2022). "Principle and application of physics-inspired neural networks for electromagnetic problems," in *IGARSS 2022–2022 IEEE International Geoscience and Remote Sensing Symposium* (IEEE), 5244–5247. doi: 10.1109/IGARSS46834.2022.9883025

Lohit, S., Liu, D., Mansour, H., and Boufounos, P. T. (2019). "Unrolled projected gradient descent for multi-spectral image fusion," in *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE), 7725–7729. doi: 10.1109/ICASSP.2019.8683124

Lu, Y., Chen, Y., Zhao, D., and Chen, J. (2019). "Graph-fcn for image semantic segmentation," in *International Symposium on Neural Networks* (Springer), 97–105. doi: 10.1007/978-3-030-22796-8_11

McCulloch, W. S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Mathem. Biophys.* 5, 115–133. doi: 10.1007/BF02478259

Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., et al. (2021). A graph placement methodology for fast chip design. *Nature* 594, 207–212. doi: 10.1038/s41586-021-03544-w

Monga, V., Li, Y., and Eldar, Y. C. (2021). Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Proc. Magaz.* 38, 18–44. doi: 10.1109/MSP.2020.3016905

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Proc.* 73, 1–15. doi: 10.1016/j.dsp.2017.10.011

Nah, S., Hyun Kim, T., and Mu Lee, K. (2017). "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 3883–3891. doi: 10.1109/CVPR.2017.35

Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). "Zero-shot learning with semantic output codes," in *Advances in Neural Information Processing Systems* 22.

Pereyra, M., Mieles, L. V., and Zygalakis, K. C. (2020). Accelerating proximal markov chain monte carlo by using an explicit stabilized method. *SIAM J. Imag. Sci.* 13, 905–935. doi: 10.1137/19M1283719

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., et al. (2020). Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*. doi: 10.21203/rs.3.rs-55125/v1

Rajabli, N., Flammini, F., Nardone, R., and Vittorini, V. (2020). Software verification and validation of safe autonomous cars: a systematic literature review. *IEEE Access* 9, 4797–4819. doi: 10.1109/ACCESS.2020.3048047

Ramella, G., and Sanniti di Baja, G. (2007). "Image Segmentation by Non-topological Erosion and Topological Expansion," in *Advances in Mass Data Analysis of Signals and Images in Medicine, Biotechnology and Chemistry*, eds. P. Perner, and O. Salvetti (Berlin, Heidelberg: Springer), 27–36. doi: 10.1007/978-3-540-76300-0_3

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Sci. Adv.* 3, e1602614. doi: 10.1126/sciadv.1602614

Shlezinger, N., Whang, J., Eldar, Y. C., and Dimakis, A. G. (2020). Model-based deep learning. *arXiv preprint arXiv:2012.08405*.

Si, Z., and Zhu, S.-C. (2013). Learning and-or templates for object recognition and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 2189–2205. doi: 10.1109/TPAMI.2013.35

Sirignano, J., and Spiliopoulos, K. (2018). Dgm: A deep learning algorithm for solving partial differential equations. *J. Computat. Phys.* 375, 1339–1364. doi: 10.1016/j.jcp.2018.08.029

Solomon, O., Cohen, R., Zhang, Y., Yang, Y., He, Q., Luo, J., et al. (2019). Deep unfolded robust pca with application to clutter suppression in ultrasound. *IEEE Trans. Med. Imag.* 39, 1051–1063. doi: 10.1109/TMI.2019.2941271

Sternberg, R. J., and Gardner, M. K. (1983). Unities in inductive reasoning. *J. Exper. Psychol.* 112, 80. doi: 10.1037/0096-3445.112.1.80

Sun, Y., Chockler, H., Huang, X., and Kroening, D. (2020). "Explaining image classifiers using statistical fault localization," in A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, *Computer Vision-ECCV 2020–16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII* (Springer), 391–406. doi: 10.1007/978-3-030-58604-1_24

Tao, X., Gao, H., Shen, X., Wang, J., and Jia, J. (2018). "Scale-recurrent network for deep image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 8174–8182. doi: 10.1109/CVPR.2018.00853

Timofte, R., Gu, S., Wu, J., and Van Gool, L. (2018). "Ntire 2018 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* 852–863.

Wang, F., Liu, H., and Cheng, J. (2018a). Visualizing deep neural network by alternately image blurring and deblurring. *Neural Netw.* 97, 162–172. doi: 10.1016/j.neunet.2017.09.007

Wang, X., Ye, Y., and Gupta, A. (2018b). "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 6857–6866. doi: 10.1109/CVPR.2018.00717

Wang, Y., Yang, J., Yin, W., and Zhang, Y. (2008). A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imag. Sci.* 1, 248–272. doi: 10.1137/080724265

Wang, Z., Liu, D., Yang, J., Han, W., and Huang, T. (2015). "Deep networks for image super-resolution with sparse prior," in *Proceedings of the IEEE International Conference on Computer Vision* 370–378. doi: 10.1109/ICCV.2015.50

Wei, Z., and Chen, X. (2019). Physics-inspired convolutional neural network for solving full-wave inverse scattering problems. *IEEE Trans Anten. Propag.* 67, 6138–6148. doi: 10.1109/TAP.2019.2922779

Wu, M., Hughes, M., Parbhoo, S., Zazzi, M., Roth, V., and Doshi-Velez, F. (2018). "Beyond sparsity: Tree regularization of deep models for interpretability," in *Proceedings of the AAAI Conference on Artificial Intelligence* 1. doi: 10.1609/aaai.v32i1.11501

Wu, M., Parbhoo, S., Hughes, M., Kindle, R., Celi, L., Zazzi, M., et al. (2020). "Regional tree regularization for interpretability in deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence* 6413–6421. doi: 10.1609/aaai.v34i04.6112

Xu, K., Wu, L., Ye, X., and Chen, X. (2020). Deep learning-based inversion methods for solving inverse scattering problems with phaseless data. *IEEE Tran. Anten. Propag.* 68, 7457–7470. doi: 10.1109/TAP.2020.2998171

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579.*

Yue, Z., Wang, T., Sun, Q., Hua, X.-S., and Zhang, H. (2021). "Counterfactual zero-shot and open-set visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 15404–15414. doi: 10.1109/CVPR46437.2021.01515

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision* (Springer), 818–833. doi: 10.1007/978-3-319-10590-1_53

Zhang, K., Gool, L. V., and Timofte, R. (2020). "Deep unfolding network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Zhang, Q., Cao, R., Zhang, S., Redmonds, M., Wu, Y. N., and Zhu, S.-C. (2017). Interactively transferring cnn patterns for part localization. *arXiv preprint arXiv:1708.01783.*

Zhang, Q., Wu, Y. N., and Zhu, S.-C. (2018). "Interpretable convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 8827–8836. doi: 10.1109/CVPR.2018.00920

Zhang, Q., Yang, Y., Ma, H., and Wu, Y. N. (2019a). "Interpreting cnns via decision trees," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 6261–6270. doi: 10.1109/CVPR.2019.00642

Zhang, Q. S., and Zhu, S.-C. (2018). Visual interpretability for deep learning: a survey. *Front. Inf. Technol. Electr. Eng.* 19, 27–39. doi: 10.1631/FITEE.1700808

Zhang, X., Wan, J., Liu, Z., and Xu, F. (2022). Rcs optimization of surface geometry with physics inspired neural networks. *IEEE J. Multisc. Multiphys. Comput. Techn.* 7, 126–134. doi: 10.1109/JMMCT.2022.3181606

Zhang, Z., Zheng, L., Qiu, T., and Deng, F. (2019b). Varying-parameter convergent-differential neural solution to time-varying overdetermined system of linear equations. *IEEE Trans. Autom. Control* 65, 874–881. doi: 10.1109/TAC.2019.2921681

Zhou, F., Jin, L., and Jun, D. (2017). "A review of convolutional neural networks," in *2017 International Conference on Communication and Signal Processing (ICCSP)* (IEEE), 0588–0592.

scientific reports

Check for updates

OPEN

Wavelet scattering transform application in classification of retinal abnormalities using OCT images

Zahra Baharlouei[1], Hossein Rabbani[1]✉ & Gerlind Plonka[2]

To assist ophthalmologists in diagnosing retinal abnormalities, Computer Aided Diagnosis has played a significant role. In this paper, a particular Convolutional Neural Network based on Wavelet Scattering Transform (WST) is used to detect one to four retinal abnormalities from Optical Coherence Tomography (OCT) images. Predefined wavelet filters in this network decrease the computation complexity and processing time compared to deep learning methods. We use two layers of the WST network to obtain a direct and efficient model. WST generates a sparse representation of the images which is translation-invariant and stable concerning local deformations. Next, a Principal Component Analysis classifies the extracted features. We evaluate the model using four publicly available datasets to have a comprehensive comparison with the literature. The accuracies of classifying the OCT images of the OCTID dataset into two and five classes were 100% and 82.5%, respectively. We achieved an accuracy of 96.6% in detecting Diabetic Macular Edema from Normal ones using the TOPCON device-based dataset. Heidelberg and Duke datasets contain DME, Age-related Macular Degeneration, and Normal classes, in which we achieved accuracy of 97.1% and 94.4%, respectively. A comparison of our results with the state-of-the-art models shows that our model outperforms these models for some assessments or achieves nearly the best results reported so far while having a much smaller computational complexity.

The retina is the innermost layer in the eye that creates vision. Various diseases have been diagnosed in this sensitive part of the eye, which affect different layers of the retina. In Diabetic Retinopathy (DR); retinal blood vessels can leak or become blocked. Several changes, such as increasing the thickness of retinal layers, are seen in this abnormality. It is a serious cumulative vascular condition that damages retinal cells with no obvious visual symptoms at first but it can progress to a widespread and severe state, and the disease's progression can result in blindness[1]. The changes in DR involve the retinal microvasculature specifically the tight junctions of the endothelial cell wall[2]. Age-related Macular Degeneration (AMD) usually appears with thickness in the Retinal Pigment Epithelium (RPE) layer. AMD originates either from the choroid or, less frequently, from the retinal circulation. The leakage in the aberrant vessels results in fluid accumulation underneath the retina and leads to rapid visual loss[3]. AMD is categorized into three stages as early, intermediate, and late stages. Two ones are non-advanced stages with no fluid or atrophy. The advanced AMD is characterized by the advanced dry stage and advanced exudative stage[4]. Macular Hole (MH) lead to distorted or blurred vision, as well as a decrease in visual acuity. Thickened edges, fluid accumulation, and macular edema are signs of MH. An important factor in the development of MH is parafoveal vitreous detachment. Anteroposterior traction with parafoveal vitreous detachment may be involved in the onset and development of MH[5]. Central Serous Retinopathy (CSR) is an eye condition characterized by the accumulation of fluid under the retina in the central macular area. Leakage of fluid into the retina through an RPE defect is seen in CSR[6]. In this disease, dysfunctional retinal pigment epithelial cells and/or choroid lining the retina lead to the development of sub-retinal fluid[7].

Retinal abnormalities are diagnosed through observation of the retinal images. Optical Coherence Tomography (OCT) is a widely accessible, non-invasive medical imaging technique that uses light to capture pictures at microscopic resolution from the retina[8]. Manual diagnosis of retinal abnormalities is costly and time-consuming and also requires highly trained clinicians to have precision. Early diagnosis of such pathologies can decrease

[1]Medical Image and Signal Processing Research Center, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, Isfahan, Iran. [2]Institute for Numerical and Applied Mathematics, Georg-August-University of Goettingen, Göttingen, Germany. ✉email: h_rabbani@med.mui.ac.ir

the risk of vision loss and the cost of treatment. Recently, computer-aided diagnosis (CAD) in retinal OCT has been considered to assist ophthalmologists in the early detection of retinal pathologies. In this line, new machine learning and deep learning algorithms have been proposed for pre-processing, abnormality diagnosis, segmentation, and classification of OCT images[9–15].

Deep learning-based methods have been shown to generally outperform classical machine learning methods[16]. However, there are also various disadvantages, such as the requirement for using a large training datasets, increasing complexity or processing time; and lack of interpretability[17,18]. Unclear extracted features and decision methods in the network layers may not be helpful for some clinical applications in real life[19]. Furthermore, a high-performance deep learning method that is adjusted for a specific dataset may not be appropriate for different datasets.

To overcome the mentioned shortages in deep learning architectures, a particular CNN was proposed using Wavelet Scattering Transforms (WST) in[20,21]. Since WST contains a cascade of wavelet transform convolutions and nonlinear modulus and averaging operators in each layer, it can be interpreted as a convolutional neural network. Convolution networks cascade convolutions and pooling nonlinearity, which in WST is the modulus of a complex number. The wavelet scattering network provides frequency and time resolutions. This transform preserves high-frequency information for classification, and is invariant to translations. Moreover, it is stable to small local deformations. It takes advantage of CNN while reducing its adverse properties[22].

In this paper, we want to diagnose retinal diseases using OCT images applying WST. We do not employ any pre-processing of the data but rely on the decorrelation property of the wavelet transform. In this way, we obtain an efficient model with essentially decreased computational cost compared to deep learning models. Using only two layers of the WST network, we can already achieve comparable accuracy with the state-of-the-art methods.

To evaluate the model, we use several datasets. We get OCT images from the OCTID dataset[23] with five classes and 572 images, to show the accuracy of this method on a small number of images and the large number of classes as was shown in[24]. We also evaluate the model using the TOPCON (which includes two classes and 57171 images)[25], the Heidelberg (which includes three classes and 4254 images)[26], and the Duke (which includes three classes and 3231 images)[27] datasets to show the generalization of the method which achieves relatively good accuracy on different datasets with different properties such as technologies, the number of images and classes, and with different dimensions. Without using any pre-processing step and with the small number of layers, we propose a very efficient model to classify the OCT images, which is implementable in practice. After data processing using WST, a Principle Component Analysis (PCA) based classifier is implemented for classification. The results show that using WST, good accuracy can be achieved for the classification of OCT images with this simple architecture.

The novelty and the contribution of this work can be summarized as follows:

- For the first time, we use the WST method to detect retinal abnormalities using different OCT datasets.
- To decrease the computational complexity and increase the speed, we don't use any pre-processing on the images. We also use only two layers of WST.
- To show the accuracy of this method on different datasets with different numbers of classes and images, and different technologies, we test the method on four well-known datasets.
- We show that this architecture can achieve an acceptable accuracy with a small amount of data which is important in medical applications.
- We reach accuracies comparable to state-of-the-art methods. In some cases, this method outperforms the others.

The rest of the paper is organized as follows: First, we have a literature review in section "Related works". The section "Materials and Methods" introduces the datasets and describes the method. In section "Results" the experimental results are presented. In the section "Discussion", we summarize the results and analyze them. Section "Conclusion" summarizes the article.

Related works

The results of previous classification methods in the literature differ concerning dataset properties (such as the contrast of images, imaging system, noise level, size of dataset), the network depth, the generality of the algorithm, computational complexity, and processing time. Therefore, the methods cannot be easily compared[19]. For example[28], achieved an accuracy of 88.4% using 2000 images from the EyePACS dataset, while[18] reported an accuracy of 97.93%, using a more complex network and 35,126 images from the same dataset. Authors in[29], used a four layers Convolutional Neural Network (CNN), and reported accuracies of 87.83% using pre-processing and 81.8% without it.

Some papers focus on diagnosing only one particular disease. In He et al.[30], AMD was diagnosed from Normal cases using ResNet-50. The AUC of 0.99, Sensitivity of 95.02%, and Specificity of 95.02 were the reported results. Dry AMD (drusen) versus wet AMD was diagnosed from OCT images using FPN-VGG-16 which lead to 93.4% accuracy[31]. In An et al.[32], AMD with fluid versus AMD without fluid using VGG-16 achieved to the accuracy of 95.1%.

Thomas et al.[33] used Recurrent Neural Network (RNN) for the classification of AMD from Normal ones. Many articles have addressed DR detection, e.g.[34–41]. Several papers used Deep CNN (DCNN) method on various datasets, e.g.[35–38], to detect DR. The obtained accuracy differs from 82.1 to 99.7%.

Some other papers tried to diagnose two and more diseases using different methods and datasets. Rasti et al.[43] recognized AMD, DME, and Normal cases with an accuracy of 98.14%, using a multi-scale convolutional mixture of experts, while[44] diagnosed the same classes with an accuracy of 92.06%, using surrogate CNN. Using

a wavelet-based CNN model, an accuracy of 98.67% was achieved for the three-class classification task in Kafieh et al.[25]. In Elmoufidi et al.[45], Different stages of DR were detected using CNN.

In addition to OCT images, some datasets acquired by other imaging technologies such as Fundus and OCT Angiography (OCTA) are used in the papers. Fundus is preferred for vascular diseases[46–49]. Hacisoftaoglu[47] using smartphone based methods on some datasets with Fundus images achieved to 98.6% of accuracy. Using DCNN, 10-fold cross-validation, an accuracy of 99.28% was achieved in Shankar et al.[48]. Some researchers evaluated their works using both OCT and Fundus images, e.g.[2,49]. OCTA has recently attracted the attention of researchers. It's a non-invasive imaging technique used in ophthalmology to visualize the blood vessels in the retina and choroid (the vascular layer behind the retina). Different studies of classification and segmentations are performed on such images, e.g.[50–53].

A review of the retinal diseases classification results shows that deep learning based methods mostly have higher performance than basic machine learning ones. Basic machine learning methods usually have higher rates. In Sandhu et al.[54], the authors tried to reduce the image dimensions and improve the classification performance, using the feature bagging technique. They achieved an accuracy of 80% with low computational time. In Somasundaram and Ali[55], by extracting wavelet features and using four classification methods, 82% accuracy was obtained. In some basic machine learning models, high accuracy was achieved using special pre-processing techniques. For example, in Ali[56], a novel pre-processing method was proposed, different features were extracted, and five classification methods were implemented to achieve an average accuracy of 98.83%. Compared with[54], improving the accuracy in Ali[56] was in return for increasing the processing time. Most CNN-based methods and specifically, DCNN models, achieved higher accuracy than others. For example[38,48,57], achieved the best accuracy of 99.1%, 99.28%, and 99.73%, respectively in detecting DR grades using DCNN models.

Materials and method

In this work, we aim to diagnose retina diseases from OCT images. We use the Wavelet Scattering Transform (WST) to access a sparse representation of images. Next, we employ a PCA-based classifier to categorize the retina diseases into different classes. We test our model on different OCT datasets to verify the accuracy of the model. We use the OCTID dataset to show the relatively good accuracy of the model to detect diseases from a large number of classes and a small amount of training data. Finally, we also use some well-known datasets involving a different number of images in 2 or 3 classes to compare the accuracy with state-of-the-art models in the literature. The block diagram of the architecture is shown in Fig. 1.

In the rest of this section, we explain the used datasets, the method, and the classification in more detail.

OCT datasets

In this work, four open-access datasets of OCT images are used. In the following, we describe the details of the OCTID[23], TOPCON[25], Duke[27], and Heidelberg[26] datasets.

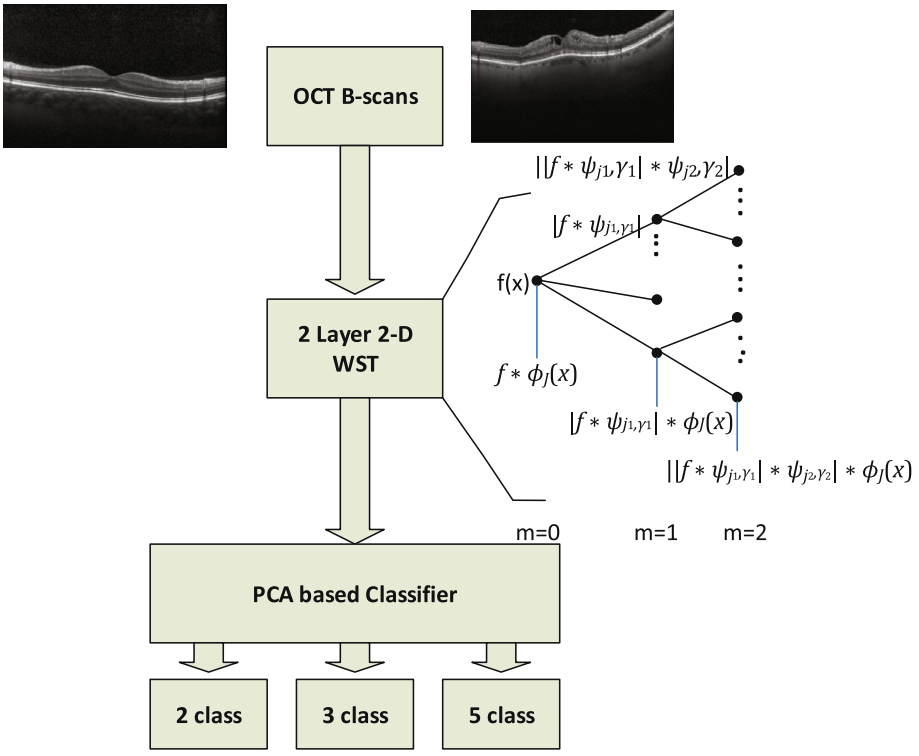


Figure 1. Block diagram of the model.

The OCTID dataset includes 572 OCT images that are categorized into five classes Normal, CSR, MH, AMD, and DR. Images have 586 × 879 pixel resolution and 2 mm scan length, which are obtained from a raster scan protocol using a Cirrus HD-OCT machine[23].

The TOPCON dataset includes 57171 B-scans of DME and Normal images with 650 × 512 resolution obtained from the Topcon 1000 device in the Ophthalmology Dept., Feiz Hospital, Isfahan, Iran.

The Duke-Harvard-Michigan Heidelberg dataset contains 45 cases of AMD, DME, and Normal with a total of 3231 OCT images, which have 496 × 1024 resolution.

The dataset from the Heidelberg device was acquired at Noor Eye hospital in Tehran containing 50 Normal and DME, and 48 AMD cases with a total of 4254 OCT images. The resolution of images is 512 × 496.

A sample of the images in each class of these datasets is presented in Fig. 2 and the properties of the used datasets in this work are listed in Table 1.

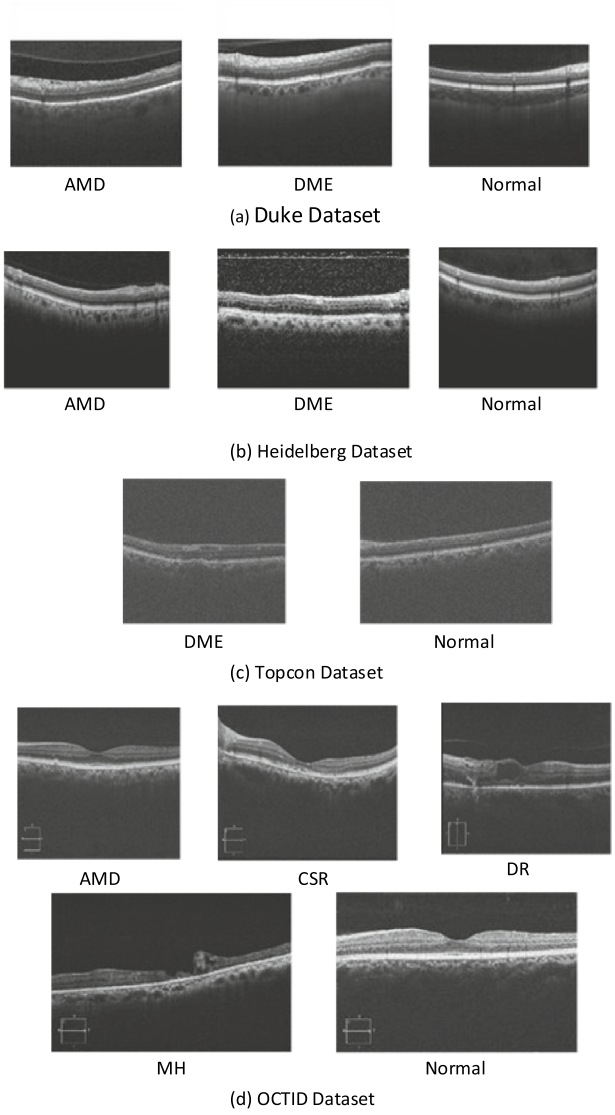


Figure 2. A sample of the OCT images in the datasets.

Dataset	No. Images	Access	Resolution	Classes
OCTID[23]	572	OA	586 × 879	Normal (206), CSR (102), MH (102), AMD (55), and DR (107)
Duke[27]	3231	OA	496 × 1024	Normal (1407), DME (1101), and AMD (723)
Heidelberg[26]	4254	OA	512 × 496	Normal (1585), DME (1104), and AMD (1565)
TOPCON[25]	57171	OA	650 × 512	Normal (33313) and DME (23858)

Table 1. OCT datasets used in this work.

Method

Wavelet scattering transform

We want to use a model with low computational cost and a high classification rate to be implementable in practice for medical tasks. In this model, we use WST to extract the important image features. Unlike deep learning models, the WST can be easily interpreted. The scattering coefficients at each scale and orientation capture different levels of signal information which are crucial for better classification. The WST is designed to be invariant to deformations, rotations, and translations, making it robust to variations in the input signal. This is particularly important in medical imaging applications where the position and orientation of the part being imaged can vary. Moreover, it preserves high-frequency information.

The WST requires fewer training examples than deep learning methods, making it a good choice for applications where labeled data is limited. We need a much smaller amount of training data to achieve clear discrimination of up to five classes.

This method is computationally efficient that can handle large volumes of data. This makes it a good choice for applications where real-time processing is required. Our results show that only two layers in this network are sufficient to achieve very good classification results.

We feed 2-D OCT images, without any pre-processing, to a WST architecture. After transferring the OCT images to the sparse representation, a PCA-based classifier categorizes the retina diseases into different classes

In the following we briefly summarize the WST approach in the continuous setting.

Let $f(\mathbf{x})$ with $\mathbf{x} = (x_1, x_2)^T$ be the two-dimensional signal on a rectangular (image) domain $\Omega \subset \mathbb{R}^2$. In the first step, the image f is filtered by applying convolutions with the scaled Gaussian (low-pass) function ϕ and a scaled and rotated (band-pass) wavelet function ψ. Then we take the modulus of these convolutions and apply a localized averaging by convolution with the scaled Gaussian ϕ. As in Bruna and Mallat[22], let

$$\phi_J(\mathbf{x}) = 2^{-2J}\phi\left(2^{-J}\mathbf{x}\right),$$

where $\phi(\mathbf{x}) := \frac{1}{2\pi\sigma^2}\exp(-|\mathbf{x}|^2/2\sigma^2)$ is the two-dimensional Gaussian window function with $\sigma = 0.85$. Then

$$(\mathscr{S}_{0,J}f)(\mathbf{x}) := (f * \phi_J)(\mathbf{x}) = \int_\Omega f(\mathbf{y})\,\phi_J(\mathbf{x}-\mathbf{y})\,d\mathbf{y}$$

is the zeroth order scattering coefficient representing the low-pass part of f. Next, we consider the two-dimensional Morlet wavelet

$$\psi(\mathbf{x}) := c_1(\mathrm{e}^{3\pi\mathrm{i}\mathbf{x}/4} - c_2)\phi(\mathbf{x}),$$

where c_1 is a normalization factor and c_2 is chosen such that $\int_{\mathbb{R}^2}\psi(\mathbf{x})\,d\mathbf{x} = 0$. In other words, $\psi(\mathbf{x})$ is the difference between a plane wave and a constant, localized by the Gaussian window $\phi(\mathbf{x})$, and can be interpreted as a band-pass filter.

Further, let $\Gamma := \{0, \frac{\pi}{r}, \frac{2\pi}{r}, \ldots, \frac{(r-1)\pi}{r}\}$ be a fixed set of r equidistant rotation angles in $[0, \pi)$ where we usually set $r = 12$ in our experiments. Then the scaled and rotated wavelet functions are determined by

$$\psi_{j,\gamma}(\mathbf{x}) := 2^{-2j}\,\psi(2^{-j}\mathbf{R}_\gamma\mathbf{x}), \qquad j = 0, \ldots, J-1,\ \gamma \in \Gamma,$$

where $\mathbf{R}_\gamma = \begin{pmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{pmatrix}$ denotes the rotation matrix corresponding to $\gamma \in \Gamma$. The vector of scattering coefficients of the first order is now given by

$$\mathscr{S}_{1,J}f(\mathbf{x}) := \{(|f * \psi_{j_1,\gamma_1}| * \phi_J)(\mathbf{x}) : j_1 = 0, \ldots, J-1,\ \gamma_1 \in \Gamma\}.$$

Indeed, the $L^1(\mathbb{R}^2)$-norm $|f * \psi_{j_1,\gamma_1}|_1 = \int_\Omega |(f * \psi_{j_1,\gamma_1})(\mathbf{x})|\,d\mathbf{x}$ is obviously translation-invariant. Employing the convolution with a wide Gaussian window ϕ_J gives a similar result, i.e., we have almost translation invariance, i.e., we have $\mathscr{S}_{1,J}f(\mathbf{x}+\boldsymbol{\tau}) \approx \mathscr{S}_{1,J}f(\mathbf{x})$ if the components of $\boldsymbol{\tau}$ are small enough. The scattering coefficients of the first order are equivalent to the feature vector obtained in the Scale-Invariant Feature Transform (SIFT), a locally invariant image descriptor proposed in Lowe[42]. The convolution of $|f * \psi_{j_1,\gamma_1}(\mathbf{x})|$ with the Gaussian window $\phi(\mathbf{x})$ is a low-pass filtering procedure that causes an information loss. To achieve improved high-frequency information, the vector of scattering coefficients of the second order is computed as

$$\mathscr{S}_{2,J}f(\mathbf{x}) := \{(||f * \psi_{j_1,\gamma_1}| * \psi_{j_2,\gamma_2}| * \phi_J)(\mathbf{x}) : j_1 = 0, \ldots, J-1,\ j_2 = j_1, \ldots, J-1,\ \gamma_1, \gamma_2 \in \Gamma\}.$$

More translation-invariant scattering coefficients can be computed by iterating this procedure, and the energy of the image signal f is propagated across the scattering coefficients. As has been shown in Bruna and Mallat[60], the scattering coefficients of order 0 to 2 in

$$\mathscr{S}_{0,J}f(\mathbf{x}),\ \mathscr{S}_{1,J}f(\mathbf{x}),\ \mathscr{S}_{2,J}f(\mathbf{x})$$

contain usually already more than 98 % of the energy of f. Thus we use only the coefficients in layers 0, 1, 2, which reduces the computational complexity significantly. Figure 3 shows the WST with $m = 2$ used in this work. Observe that in the considered continuous setting the image f as well as all scattering coefficients are still functions on Ω. We set the dimension of the scaling filter, called invariant scale, equal to the minimum dimension of the images for each dataset used in this paper. In practice, we have a given discrete image $\mathbf{f}$ with N pixels and the convolutions have to be discretized. The total number of scattering coefficients in $\mathscr{S}_{1,J}$ is Jr and the number of scattering coefficients in $\mathscr{S}_{2,J}$ is $r^2\frac{J(J-1)}{2}$, where r is the number of considered angles. These functions

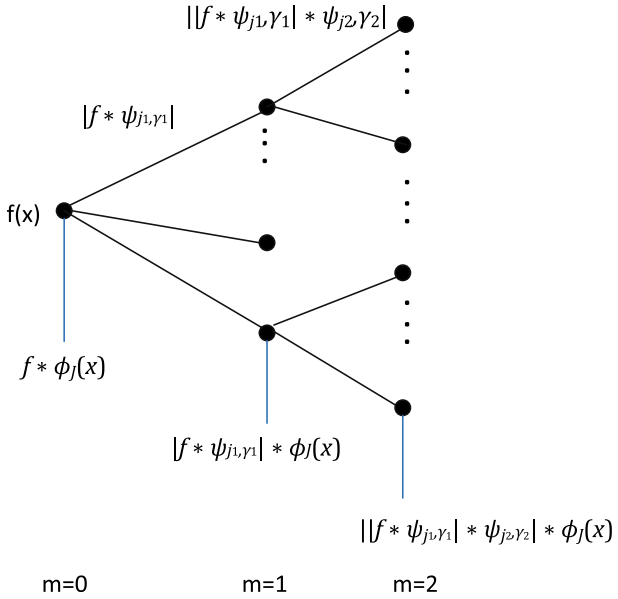


Figure 3. The wavelet scattering network with two layers.

are uniformly sampled with grid size 2^J such that each discretized scattering coefficient has $2^{-2J}N$ coefficients, where N is the number of pixels of the image $\mathbf{f}$. Together, the total number of the discrete feature vectors $S_J\mathbf{f}$ of $\mathbf{f}$ (consisting of components of all feature coefficients of order 0, 1, and 2) is then $N_J := (1 + rJ + r^2\frac{J(J-1)}{2})2^{-2J}N$.

Classifier

We employ a classifier based on PCA applied to a suitable affine space, as proposed in Bruna and Mallat[22]. The classification algorithm directly employs the scattering coefficient vectors $S_J\mathbf{f}$. Assume that we have computed a complete vector of scattering coefficients of length N_J that contains the scattering coefficients of $\mathbf{f}$ of order 0, 1, and 2 at subsampled positions. Each signal class is represented by a random vector $\mathbf{f}_k$, and the realizations of this random vector are the images $\mathbf{f}$ in this class. Let $E(S_J\mathbf{f}_k)$ denote the expected scattering coefficient vector of length N_J of images $\mathbf{f}$ in class k. Further, let $\mathbf{V}_{d,k}$ be the rank-d approximation of the covariance matrix of $S_J\mathbf{f}_k$ of size $N_J \times N_J$ built by the eigenvectors of the covariance matrix corresponding to the largest d eigenvalues. In our experiments, we have used $d = 30$. We obtain the affine approximation space

$$\mathbf{A}_{d,k} = E(S_J\mathbf{f}_k) + \mathbf{V}_{d,k},$$

see also[22]. Having found this affine space, the classifier associates an image $\mathbf{f}$ to the class k (among K classes) if

$$k(\mathbf{f}) = \text{argmin}_{1\le k'\le K}\|S_J\mathbf{f} - P_{\mathbf{A}_{d,k}}(S_J\mathbf{f})\|_2,$$

where $P_{\mathbf{A}_{d,k}}$ denotes the projection onto the affine space $\mathbf{A}_{d,k}$.

The computational effort for the classification is governed by the required singular value decomposition of the covariance matrix of $S_J\mathbf{f}_k$ with $O(N_J^3)$ floating point operations.

Results

To assess the model, we classified the OCT images of the OCTID, TOPCON, Duke, and Heidelberg datasets. These datasets differ in technologies, the number of images and their dimensions, and also the number of classes. The wavelet scattering features are extracted, and a PCA-based classifier is used to diagnose the retinal abnormalities. In this work, a wavelet scattering transform in Matlab was implemented. As mentioned in the Method Section, the energy of signals is significantly decreased as the layers are increased. Using two layers of wavelet filter banks is sufficient for classifying OCT images. For each wavelet filter, different rotations from 6 to 12 in $[0, \pi]$ were considered. The best results were related to 12 rotations for all datasets except for OCTID, in which increasing the rotations number did not have any effect on the results. The spatial support in the row and column dimensions of the scaling filter was considered as half of the minimum dimension of the images for each dataset. To train the network, we used 80% of the data, and the rest of 20% was used to test.

We tested our model to investigate the accuracy of diagnosing five categories in the OCTID dataset. The result is shown in Fig. 4. The accuracy of this classification is 82.5%. Only one work in the literature reported the classification results for five classes in OCTID[61]. In Mishra et al.[61], the accuracy of %93.12(+/ − 8.59) was reported using a CNN model. The model includes 13 convolution layers, 4 Maxpool layers, three fully connected layers, an attention module, and reshape, normalization, flatten softmax, and lost steps. Comparing the process steps and network layers in Mishra et al.[61] with our model shows the trade-off between computational complexity and processing time with accuracy. This is other than the shortages in using black-box CNN models.

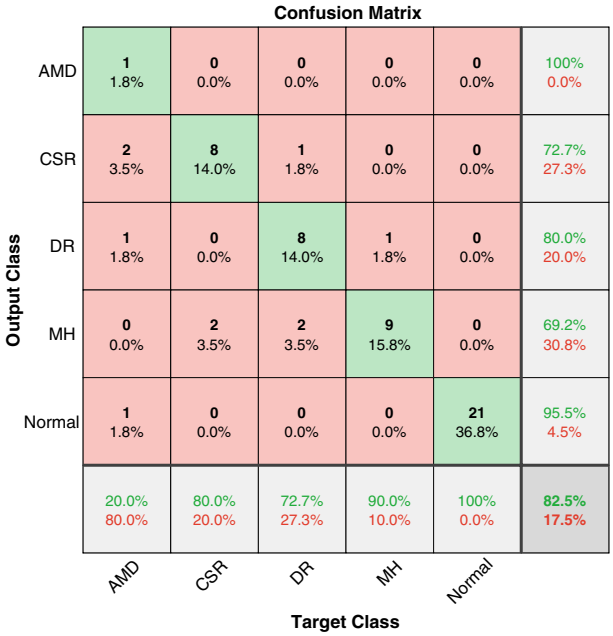


Figure 4. The confusion matrix of WST on OCTID dataset for diagnosing five classes of OCT images.

Most of the classification works on the OCTID dataset investigated the detection accuracy for two classes, one abnormality from Normal ones. We examined our model for detecting DR pathology, which is one of the most common diseases in diabetic patients. Figure 5 shows that our method achieved 100% of accuracy for DR detection. Table 2 compares our result with other works in detecting DR. As seen in the table, this model outperforms other state-of-the-art models.

Next, we tested our model on the TOPCON dataset. An accuracy of 96.6% was achieved in detecting DME from normal ones, as seen in Fig. 6. We listed the best results that have been reported in the literature in Table 3, to compare the results with other works. As seen in the table, most CNN-based works achieved a higher accuracy. The WST-based model in this paper can achieve accuracy close to the complicated architecture of CNN-based models using a simple architecture.

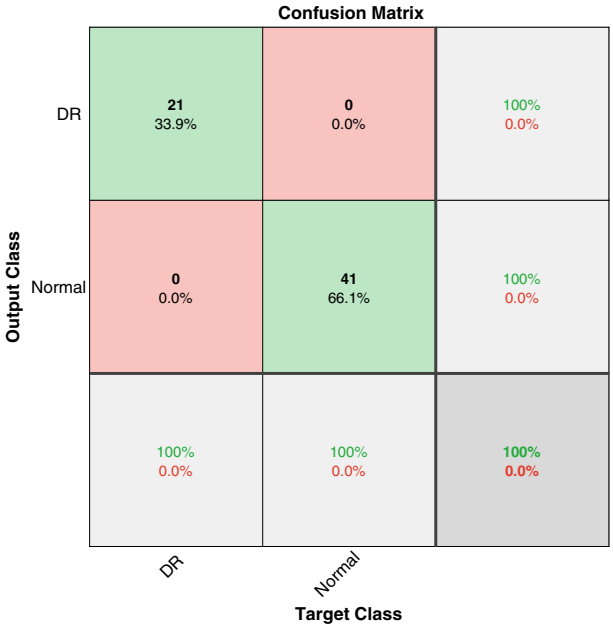


Figure 5. The confusion matrix of WST on OCTID dataset for diagnosing DR from Normal cases.

Paper	Year	Method	Dataset	Accuracy
[47]	2020	Deep learning, Smartphone based method	EyePACS, IDRiD, MESSIDOR, MESSIDOR-2	98.6%
[48]	2020	DCNN, classification based on 10-fold cross validation	MESSIDOR	99.28%
[62]	2021	Transfer learning on Inception-ResNet-V2	Kaggle dataset, MESSIDOR	Kaggle dataset: 72.33%, MESSIDOR: 82.18%
[63]	2021	Hybrid Inductive Machine Learning	CHASE	96.62%
[39]	2022	Graph-CNN	OCTID	98%
[40]	2023	2-stage noninvasive framework	SD-OCT	93.8%
[41]	2023	CNN	DRIL	88.3%
This paper		WST	OCTID	100%

Table 2. Comparing DR detection accuracy in different works.

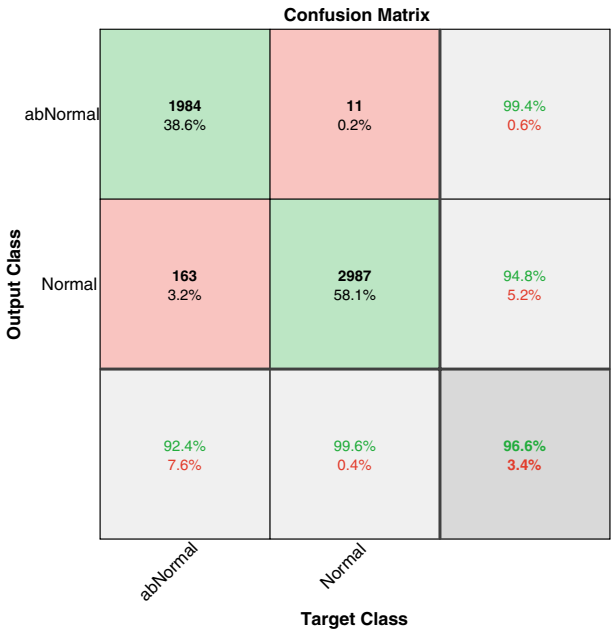


Figure 6. The confusion matrix of WST on the TOPCON dataset for diagnosing DME from Normal cases.

Paper	Year	Method	Dataset	Accuracy (%)
[67]	2011	Multiscale LBP, classification based on 10-fold cross validation	TOPCON	95.9
[27]	2014	Multiscale HOG	TOPCON	96
[64]	2017	CNN (VGG-16)	SERI	87.5
[65]	2018	CNN (VGG-16), classification based on 32-fold cross validation	SERI	93.75
[25]	2018	CNN (WCNN1), classification based on 5-fold cross validation	TOPCON	99.3
[66]	2022	CNN (DeepOCT)	ZhangLab	99.2
This paper		WST	TOPCON	96.6

Table 3. Comparing DME detection accuracy in different works.

To compare the performance of the work with the research on other well-known datasets, we tested our model on the Duke and Heidelberg datasets to diagnose DME and AMD from Normal ones. We achieved the accuracy of 97.1% and 94.4%, respectively. The results are shown in Figs. 7 and 8.

The best results reported in the literature on Duke and Heidelberg datasets are compared in Tables 4 and 5. The results show that we achieved the best accuracy in classifying on the Duke dataset. Since most of the works on the Duke dataset used k-fold cross-validation, we also implemented 10-fold validation to have a fair comparison. We achieved 96.7% of accuracy which is the best result reported in the literature and equal to the one in

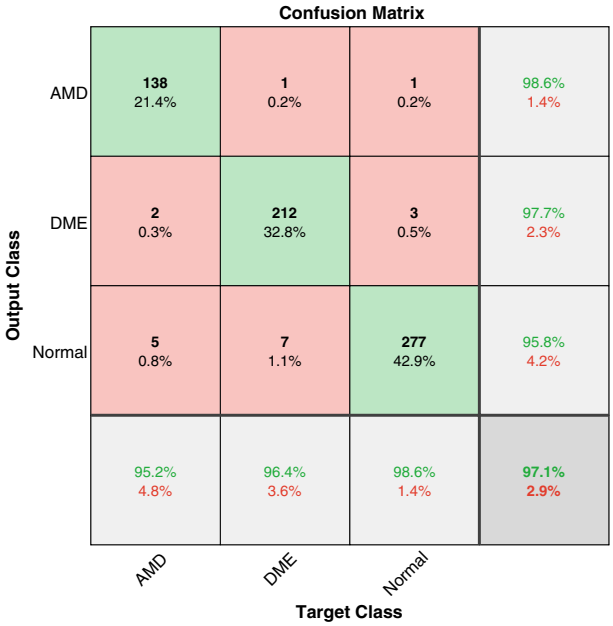


Figure 7. The confusion matrix of WST on the Duke dataset.

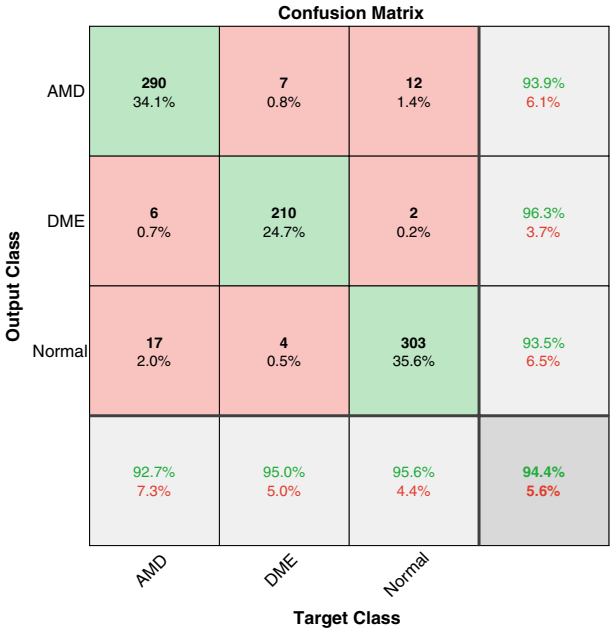


Figure 8. The confusion matrix of WST on the Heidelberg dataset.

Thomas et al.[33]. The classification accuracy of this work on the Heidelberg dataset is close to the best results in the literature but less than some. An overall view of the results on different datasets shows that this model achieves similarly good classification results as the other state-of-the-art models, specifically the CNN-based ones.

Discussion

In this article, we used the WST-based method to diagnose retinal diseases from OCT images. We achieved different accuracies for the four databases used. Comparing the accuracy obtained in this method with other methods in Tables 2, 3, 4 and 5 shows that this method is generally comparable with state-of-the-art and highly accurate methods. As mentioned, the presented results are using two layers of the WST. We have shown the effect of using fewer layers on the results in Supplementary Appendix 1. In the appendix, we have also discussed the cause of failure cases in the classification.

Paper	Year	Method	Classes	Accuracy
[33]	2021	Multipath CNN, classification based on 10-fold cross validation	AMD, Normal	96.7%
[68]	2021	Statistical method	AMD, Normal	96.6%
[70]	2021	DL (VGG-16)	AMD,DME, Normal	94.2%
[69]	2022	Classical ML (n-gram), classification based on 10-fold cross validation	AMD, DME, Normal	AMD:86.7% DME:93.3% Normal: 93.3%
This paper		WST	AMD DME Normal	97.1%
		WST, classification based on 10-fold cross validation		96.7%

Table 4. Comparing DME, AMD, and Normal detection accuracy in different works using the Duke dataset.

Paper	Year	Method	Classes	Accuracy (%)
[43]	2018	CNN (MCME), classification based on 5-fold cross validation	AMD, DME, Normal	99.01
[73]	2019	DL CliqueNet based	AMD, DME, Normal	98
[72]	2021	CNN	AMD, DME, Normal	93.87
[33]	2021	Multipath CNN, classification based on 10-fold cross validation	AMD, Normal	98.9
[71]	2022	DL(fine-tuned)	AMD, DME, Normal	96.5
This paper		WST	AMD, DME, Normal	94.4

Table 5. Comparing DME, AMD, and Normal detection accuracy in different works using the Heidelberg dataset.

Among the advantages of this method over deep learning methods is short processing time. The computational cost for the WST only depends on the input size of the image, the chosen predefined scale 2^J and the number of angles r, and can be given as $O(N_J \log(N))$ for an image with N pixels. This means, the effort to perform the WST is even smaller than the needed cost to compute the low rank approximation of the correlation matrix of size $N_J \times N_J$ for classification.

In Table 6 we report all the obtained accuracies in this paper. Considering accuracy, our method outperforms previous research in DR detection using the OCTID dataset (with a very small amount of data) and on the Duke dataset. In other cases, the accuracy of our method is not much different from the best results obtained.

We also calculated AUC (Area under the ROC Curve). According to Table 6, our method has the best AUC on the Duke dataset, but this result is lower compared to previous research reports, which mostly reached an AUC above 0.9.

Using ANOVA statistical testing, we calculated the P-value for the experiments. The best results were achieved in the experiment performed on the OCTID dataset with five classes and on the TOPCON dataset, as seen in the table.

Conclusions

Various retinal diseases can be diagnosed using OCT images. To overcome some shortages in manual diagnosing, such as mistakes and costs, computer-aided manners have been considered today. Various classical machine learning and deep learning methods have been proposed in this field. Although deep learning techniques, specifically CNN-based methods, can achieve high accuracies in detecting different abnormalities, some shortages make them often impractical. Application problems in practice include the high computation complexity, long processing time, requirement of large datasets, and unclear interpretability.

In this paper, we implemented the wavelet scattering network to diagnose retinal abnormalities using OCT images. This transformation overcomes some mentioned shortages of CNN methods. In particular, the CNN of

Dataset	Numer of Images	Number of classes	Acc	AUC	P-value
OCTID	572	5	82.5%	0.87	0.0109
OCTID	313	2	100%	0.78	0.333
Duke	3231	3	97.1%	0.88	0.2385
Heidelberg	4254	3	94.4%	0.82	0.94
Topcon	57171	2	96.6%	0.68	0.0344

Table 6. The experimental results of using the WST on four OCT datasets.

the WST is based on predefined wavelet filters. Employing only two layers of the WST, we achieved an efficient model with low computational complexity.

This is the first time that WST was used on OCT images. In previous research, WST-based methods have been proposed for the classification of EEG and ECG signals, and in most cases, good results have been achieved compared to other methods. In this article, using this method and without pre-processing, we categorized retinal diseases using several OCT databases to obtain an evaluation of the different numbers of image classes, technologies, and sizes of images. We performed a comprehensive assessment and comparison of the method.

The accuracies of classifying the OCT images of the OCTID dataset into five and two classes were 82.5% and 100%, respectively. We achieved an accuracy of 96.6% in diagnosing DME from Normal ones using the TOPCON device-based dataset. The Heidelberg and the Duke datasets contain DME, AMD, and Normal classes, where we achieved 97.1% and 94.4%, respectively.

Comparing our results with the state-of-the-art models in the literature shows that this model outperforms the compared models in detecting DR in the OCTID and the Duke dataset with three classes. In other cases, our results are comparable with other works, specifically with CNN-based techniques. An acceptable decrease in accuracy of some assessments was seen comparing the best results that have been reported in the literature, in return for an essential decrease of the computational complexity and processing time which are essential factors in practice.

Although the classification results with this method are generally good, it still needs to be improved. In future works, we aim to upgrade the method by finding more proper wavelet filters that are particularly adapted to the special features of OCT images and which can increase the performance of diagnosing retinal disease. We also examine the effectiveness of this method to detect real samples.

Data availability

The authors declare that the data supporting the findings of this study are available at the links below:
The OCTID dataset is available at: https://borealisdata.ca/dataverse/OCTID?q=&types=datasets&sort=dateSort&order=desc&page=1. The TOPCON dataset is available at: https://misp.mui.ac.ir/en/topcon-3d-oct-diabetic-data-denoising-0. The Douck dataset is available at: https://people.duke.edu/~sf59/Srinivasan_BOE_2014_dataset.htm. The Heidelberg dataset is available at: https://misp.mui.ac.ir/en/dataset-oct-classification-50-normal-48-amd-50-dme-0.

Received: 10 October 2022; Accepted: 29 October 2023
Published online: 03 November 2023

References

1. Elgafi, M. *et al.* Detection of diabetic retinopathy using extracted 3D features from OCT images. *Sensors* **22**(20), 7833 (2022).
2. Pavithra, K. C., Kumar, P., Geetha, M., & Bhandary, S. V. Computer aided diagnosis of diabetic macular edema in retinal fundus and OCT images: A review. *Biocybern. Biomed. Eng.* (2023).
3. Pawloff M., Gerendas, B. S., Deak, G., Bogunovic, H., Gruber, A. & Schmidt-Erfurth U. Performance of retinal fluid monitoring in OCT imaging by automated deep learning versus human expert grading in neovascular AMD. *Eye* 1–8 (2023).
4. Moradi, M., Chen, Y., Du, X. & Seddon, J. M. Deep ensemble learning for automated non-advanced AMD classification using optimized retinal layer segmentation and SD-OCT scans. *Comput. Biol. Med.* **154**, 106512 (2023).
5. Sakaguchi, H. *et al.* Relationship between full-thickness macular hole onset and posterior vitreous detachment: A temporal onset theory. *Ophthalmol. Sci.* **3**(4), 1003–39 (2023).
6. Nicholson, B., Noble, J., Forooghian, F. & Meyerle, C. Central serous chorioretinopathy: Update on pathophysiology and treatment. *Survey ophthal.* **58**, 103–26 (2013).
7. Patel, G., Edirisooriya, M., Dey, M. & Parkar, R. Bilateral multifocal central serous retinopathy due to management of metastatic melanoma with BRAf MEK inhibitors: Case report. *Curr. Probl. Cancer Case Rep.* **9**, 1002–08 (2023).
8. Fujimoto, J. G., Drexler, W., Schuman, J. S. & Hitzenberger, C. K. Optical coherence tomography (OCT) in ophthalmology: Introduction. *Opt. Express* **17**, 3978–3979 (2009).
9. Amini, Z. & Rabbani, H. Statistical modeling of retinal optical coherence tomography. *IEEE TMI* **35**, 1544–1554 (2016).
10. Rabbani, H., Sonka, M. & Abramoff, M. OCT noise reduction using anisotropic local bivariate gaussian mixture prior in 3D complex wavelet domain. *Int. J. Biomed. Imaging* **22** (2013).
11. Esmaeili, M., Mehri, A., Rabbani, H. & Hajizadeh, F. 3D segmentation of retinal cysts from SD-OCT images by the use of 3D curvelet based K-SVD. *JMSS* **6**, 166–171 (2016).
12. Huang, L. *et al.* Automatic classification of retinal optical coherence tomography images with layer guided convolutional neural network. *IEEE Signal Proc. Lett.* **26**, 1026–1030 (2019).
13. Rasti, R., Mehridehnavi, A., Rabbani, H. & Hajizadeh, F. Convolutional mixture of experts model: A comparative study on automatic macular diagnosis in retinal optical coherence tomography imaging. *J. Med. Signals Sens.* **9**(1), 1 (2019).
14. Jalili, J., Rabbani, H., Dehnavi, A. M., Kafieh, R. & Akhlaghi, M. Forming optimal projection images from intra-retinal layers using curvelet-based image fusion method. *J. Med. Signals Sens.* **10**(2), 76 (2020).
15. Majumder, S., Elloumi, Y., Akil, M., Kachouri, R. & Kehtarnavaz, N. A deep learning-based smartphone application for real-time detection of five stages of diabetic retinopathy, in *Real-Time Image Processing and Deep Learning 2020*, Vol. 11, 106–114 (2020).
16. Skouta, A. *et al.* Deep learning for diabetic retinopathy assessments: A literature review. *Multimedia Tools Appl.* 1–6 (2023).
17. Gadekallu, T. R. *et al.* Early detection of diabetic retinopathy using PCA-firefly based deep learning model. *Electron* **9**, 274 (2020).
18. Mansour, R. F. Deep-learning-based automatic computer-aided diagnosis system for diabetic retinopathy. *Biomed. Eng. Lett.* **8**, 41–57 (2017).
19. Lakshminarayanan, V., Kheradfallah, H., Sarkar, A. & Jothi Balaji, J. Automated detection and diagnosis of diabetic retinopathy: A comprehensive survey. *J. Imaging* **7**, 165 (2021).
20. Mallat, S. Recursive interferometric representation, in *Proc. of EUSICO Conference, Danemark* (2010).
21. Mallat, S. Group invariant scattering. *Commun. Pure Appl. Math.* **65**, 1331–1398 (2012).
22. Bruna, J. & Mallat, S. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1872–1886 (2013).
23. Gholami, P., Roy, P., Parthasarathy, M. K. & Lakshminarayanan, V. OCTID: Optical coherence tomography image database. *Comput. Electr. Eng.* **81**, 106532 (2020).

24. Baharlouei, Z., Rabbani, H. & Plonka, G. Detection of retinal abnormalities in OCT images using wavelet scattering network, in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 3862–3865 (2022).
25. Kafieh, R., Rabbani, H. & Selesnick, I. Three dimensional data-driven multi scale atomic representation on optical coherence tomography. *IEEE Trans. Med. Imaging* **34**(5), 1042–62 (2015).
26. Rasti, R., Rabbani, H., Mehridehnavi, A. & Hajizadeh, F. Macular OCT classification using a multi-scale convolutional neural network ensemble. *IEEE Trans. Med. Imaging* **37**, 1024–1034 (2017).
27. Srinivasan, P. P. *et al.* Fully automated detection of diabetic macular edema and dry age-related macular degeneration from optical coherence tomography images. *Biomed. Opt. Express* **5**, 3568–3577 (2014).
28. Sayres, R. *et al.* Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy. *Ophthalmology* **126**, 552–564 (2019).
29. Pao, S. I. *et al.* Detection of diabetic retinopathy using bichannel convolutional neural network. *J. Ophthalmology.* **2020**, 1–7 (2020).
30. He, T., Zhou, Q. & Zou, Y. Automatic detection of age-related macular degeneration based on deep learning and local outlier factor algorithm. *Diagnostics (Basel)* **12**(2), 53 (2022).
31. Sotoudeh-Paima, S., Jodeiri, A., Hajizadeh, F. & Solta-nian-Zadeh, H. Multi-scale convolutional neural network for automated AMD classification using retinal OCT images. *Comput. Biol. Med.* **144**, 105368 (2022).
32. An, G., Akiba, M., Yokota, H. *et al.* Deep learning classification models built with two-step transfer learning for age related macular degeneration diagnosis, in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2049-52 (2019).
33. Thomas, A. *et al.* A novel multiscale and multipath convolutional neural network based age-related macular degeneration detection using OCT images. *Comput. Methods Programs Biomed.* **209**, 106294 (2021).
34. Abdelmaksoud, E., El-Sappagh, S., Barakat, S., AbuHmed, T. & Elmogy, M. Automatic diabetic retinopathy grading system based on detecting multiple retinal lesions. *J. IEEE Access.* **9**, 15939–15960 (2021).
35. Gangwar, A.K. & Ravi, V. Diabetic retinopathy detection using transfer learning and deep learning, in *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020)*, 679-689 (Springer, 2020).
36. He, A., Li, T., Li, N., Wang, K. & Fu, H. CABNet: Category attention block for imbalanced diabetic retinopathy grading. *IEEE TMI* **40**, 143–153 (2021).
37. Khan, Z. *et al.* Diabetic retinopathy detection using VGG-NIN a deep learning architecture. *J. IEEE Access* **9**, 61408–61416 (2021).
38. Saeed, F., Hussain, M. & Aboalsamh, H. A. Automatic diabetic retinopathy diagnosis using adaptive fine-tuned convolu-tional neural network. *J. IEEE Access* **9**, 41344–44359 (2021).
39. Sunija, A. *et al.* Multi-scale directed acyclic graph-CNN for automated classification of diabetic retinopathy from OCT images. *Biomed. Eng. Appl. Basis Commun.* **34**(05), 2250025 (2022).
40. Pour, K. *et al.* Automated machine learning-based classification of proliferative and non-proliferative diabetic retinopathy using optical coherence tomography angiography vascular density maps. *Graefes Arch. Clin. Exp. Ophthalmol.* **261**, 391–9 (2023).
41. Singh, R. *et al.* Deep learning algorithm detects presence of disorganization of retinal inner layers (DRIL)-an early imaging biomarker in diabetic retinopathy. *Transl. Vis. Sci. Technol.* **12**, 6–20 (2023).
42. Lowe, D. G. Distinctive image features from scale invariant key points. *Int. J. Comput. Vis.* **60**, 91–110 (2004).
43. Rasti, R. *et al.* Convolutional mixture of experts model: A comparative study on automatic macular diagnosis in retinal OCT imaging. *JMSS.* **9**, 1–14 (2019).
44. Rong, Y. *et al.* Surrogate-assisted retinal OCT image classification based on convolutional neural networks. *IEEE J. Biomed. Health Inf.* **23**, 253–263 (2018).
45. Elmoufidi, A. *et al.* Diabetic retinopathy prevention using EfficientNet B3 architecture and fundus photography. *SN Comput. Sci.* **4**(1), 1–9 (2023).
46. Skouta, A. *et al.* Hemorrhage semantic segmentation in fundus images for the diagnosis of diabetic retinopathy by using a convolutional neural network. *J. Big Data* **9**(1), 1–24 (2022).
47. Hacisoftaoglu, R. E., Karakaya, M. & Sallam, A. B. Deep learning frameworks for diabetic retinopathy detection with smartphone-based retinal imaging systems. *Pattern Recog. Lett.* **135**, 409–417 (2020).
48. Shankar, K. *et al.* Automated detection and classification of fundus diabetic retinopathy images using synergic deep learning model. *Pattern Recog. Lett.* **133**, 210–216 (2020).
49. Mahmudi, T., Kafieh, R., Rabbani, H., Mehri, A. & Akhlaghi, M. R. Evaluation of asymmetry in right and left eyes of normal individuals using extracted features from optical coherence tomography and fundus images. *J. Med. Signals Sens.* **11**(1), 12 (2021).
50. Liu, X., Zhang, D., Yao, J. & Tang, J. Transformer and convolutional based dual branch network for retinal vessel segmentation in OCTA images. *Biomed. Signal Process. Control* **83**, 104604 (2023).
51. Xie, J., Yi, Q. *et al.* Deep segmentation of OCTA for evaluation and association of changes of retinal microvasculature with Alzheimer's disease and mild cognitive impairment. *Br. J. Ophthalmol.* (2023).
52. Tan, X. *et al.* OCT2Former: A retinal OCT-angiography vessel segmentation transformer. *Comput. Methods Programs Biomed.* **233**, 107454 (2023).
53. Lang, Y. *et al.* Retinal structural and microvascular changes in myelin oligodendrocyte glycoprotein antibody disease and neuromyelitis optica spectrum disorder: An OCT/OCTA study. *Front. Immunol.* **14**, 1029124 (2023).
54. Sandhu, H. S. *et al.* Automated diagnosis of diabetic retinopathy using clinical biomarkers, optical coherence tomography, and optical coherence tomography angiography. *Am. J. Ophthalmol.* **216**, 201–206 (2020).
55. Somasundaram, S. K. & Ali, P. A machine learning ensemble classifier for early prediction of diabetic retinopathy. *J. Med. Syst.* **41**, 201 (2017).
56. Ali, A. *et al.* Machine learning based automated segmentation and hybrid feature analysis for diabetic retinopathy classification using fundus image. *Entropy* **22**, 567 (2020).
57. Hsieh, Y. T. *et al.* Application of deep learning image assessment software VeriSee for diabetic retinopathy screening. *J. Formos. Med. Assoc.* **120**, 165–171 (2021).
58. Anden, J. & Mallat, S. Multiscale scattering for audio classification, in *Int. Society Music Inf. Retrieval Conf. USA*, 657-662 (2011).
59. Leonarduzzi, R., Liu, H. & Wang, Y. Scattering transform and sparse linear classifiers for art authentication. *Signal Proc.* **150**, 11–19 (2018).
60. Bruna, J. & Mallat, S. Classification with scattering operators, in *Comp. Vision Pattern Recog.*, 1561–1566 (2011).
61. Mishra, S. S., Mandal, B. & Puhan, N. B. MacularNet: Towards fully automated attention-based deep CNN for macular disease classification. *SN Comput. Sci.* **3**, 142 (2022).
62. Gangwar, A.K. & Vadlamani, R. Diabetic retinopathy detection using transfer learning and deep learning, in *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020)*, 679–689 (2021).
63. Mahmoud, M. H. *et al.* An automatic detection system of diabetic retinopathy using a hybrid inductive machine learning algorithm. *Pers. Ubiquitous Comput.* 1–15 (2021).
64. Awais, M., Muller, H., Tang, T.B. & Meriaudeau, F. Classification of SD-OCT images using a deep learning approach, in *ICSIPA*, 489–492 (2017).
65. Perdomo, O., Otalora, S., Gonzalez, F. A., Meriaudeau, F. & Muller, H. OCT-NET: A convolutional network for automatic classification of normal and diabetic macular edema using sd-oct volumes, in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* (2018).

66. Altan, G. DeepOCT: An explainable deep learning architecture to analyze macular edema on OCT images. *Int. J. Eng. Sci. Tech.* **34**, 101091 (2022).
67. Liu, Y. Y. *et al.* Automated macular pathology diagnosis in retinal OCT images using multi-scale spatial pyramid and local binary patterns in texture and shape encoding. *Med. Image Anal.* **15**, 748–759 (2011).
68. Thomas, A. *et al.* RPE layer detection and baseline estimation using statistical methods and randomization for classification of AMD from retinal OCT. *Comput. Methods Programs Biomed.* **200**, 105822 (2021).
69. Wang, G., Chen, X., Tian, G. & Yang, J. A novel-gram-based image classification model and its applications in diagnosing thyroid nodule and retinal OCT images. *CMMM* **2** (2022).
70. Luo, Y. *et al.* Automatic detection of retinopathy with optical coherence tomography images via a semi-supervised deep learning method. *Biomed. Opt. Exp.* **12**, 2684–2702 (2021).
71. Khan, A. M., Hassan, T., Akram, M. U., Alghamdi, N. S. & Werghi, N. Continual learning objective for analyzing complex knowledge representations. *Sensor* **22**, 1667 (2022).
72. Thomas, A., Harikrishnan, P. M., Krishna, A. K., Ponnusamy, P. & Gopi, V. P. Automated detection of age-related macular degeneration from OCT images using multipath CNN. *J. Comput. Sci. Eng.* **15**(1), 34–46 (2021).
73. Wang, D. & Wang, L. On OCT image classification via deep learning. *IEEE Photonics J.* **11**(5), 1–14 (2019).
74. Gangwar, A.K. & Ravi, V. Diabetic retinopathy detection using transfer learning and deep learning, in *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020)*, 679–689 (2020).
75. Aldahami, M. & Alqasemi, U. Classification of oct images for detecting diabetic retinopathy disease using machine learning. *europepmc* (2020).
76. Huang, Y. P. *et al.* A fuzzy approach to determining critical factors of diabetic retinopathy and enhancing data classification accuracy. *Int. J. Fuzzy Syst.* **21**, 1844–57 (2019).
77. Ryu, G., Lee, K., Park, D., Park, S. H. & Sagong, M. A deep learning model for identifying diabetic retinopathy using optical coherence tomography angiography. *Sci. Rep.* **11**, 1–9 (2021).
78. Sabi, S., Varun, P. & Gopi, P. A dual-path CNN based age-related macular degeneration detection, in *Proc. Int. Conf. Electrical, Computer and Comm. Tech.* (2021).
79. Hassan, S. A. *et al.* Recent developments in detection of central serous retinopathy through imaging and artificial intelligence techniques-A review. *IEEE Access* **9**, 168731–168748 (2021).

Acknowledgements

This work is supported by Isfahan University of Medical Sciences (Grant No. 2400206 and No. 2401156).

Author contributions

Z.B. designed/implemented the final method and wrote the main manuscript. H.R. and G.P. designed/modified the main method and evaluated the final results. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-023-46200-1.

Correspondence and requests for materials should be addressed to H.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mechanical Systems and Signal Processing 195 (2023) 110314

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

Interpretable convolutional neural network with multilayer wavelet for Noise-Robust Machinery fault diagnosis

Huan Wang [a,b], Zhiliang Liu [a,*], Dandan Peng [c], Ming J. Zuo [a,d,e]

[a] School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] Department of Industrial Engineering, Tsinghua University, Beijing 100084, China
[c] Department of Mechanical Engineering, KU Leuven, Leuven 3000, Belgium
[d] Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta T6G 1H9, Canada
[e] Qingdao International Academician Park Research Institute, Qingdao 266041, China

ARTICLE INFO

Keywords:
Fault diagnosis
Wavelet transform
Convolutional neural network
Attention mechanism

ABSTRACT

Convolutional neural networks (CNNs) are being utilized for mechanical fault diagnosis, due to its excellent automatic discriminative feature learning ability. However, the poor interpretability and noise robustness of CNNs have plagued both academia and industry. Since traditional signal analysis technology has a sound theoretical basis and physical meaning, it motivates us to use signal processing theory to improve the interpretability and performance of the CNN algorithm. To this end, this paper proposes a multilayer wavelet attention convolutional neural network (MWA-CNN) for noise-robust machinery fault diagnosis. This framework aims to learn discriminative fault features from the wavelet domain, which allows the model to obtain better interpretability and superior performance than conventional time-domain-based CNNs. The proposed Discrete Wavelet Attention Layer (DWA-Layer) is used to map time domain signals to wavelet space, and obtain valuable information through the learnable convolutional layer. By alternately using DWA-Layer and convolutional layer for signal decomposition and feature learning, the proposed framework actually embeds a similar multi-resolution analysis algorithm in CNN. This helps integrate physics-based knowledge into the CNN. Finally, the frequency attention mechanism is proposed to enhance the ability of MWA-CNN to obtain fault-related features from different frequency components. Experiments on high-speed aeronautical bearing and motor bearing datasets prove that the proposed method has excellent fault diagnosis ability and noise robustness. The visual analysis of the attention mechanism contributes to the interpretability of CNN in the field of fault diagnosis.

1. Introduction

Modern mechanical systems generally have complex mechanical structures and work in harsh environments. Some key mechanical components (such as bearings and gears) are widely used in important mechanical systems, including wind turbines, high-speed trains, and aero engines [1]. However, during operation of mechanical equipment, the bearing inevitably appears fatigue degradation, cracks, deformation, spalling and other failures. If faulty parts are not repaired or replaced in time, it may cause serious damage to whole mechanical system. Therefore, in order to ensure normal operation of mechanical systems, it is very necessary to monitor health status

* Corresponding author.
E-mail address: zhiliang_liu@uestc.edu.cn (Z. Liu).

https://doi.org/10.1016/j.ymssp.2023.110314
Received 6 July 2022; Received in revised form 7 February 2023; Accepted 19 March 2023
Available online 29 March 2023

of key components of mechanical systems [2].

Machinery condition monitoring based on vibration signal analysis is the current mainstream technology. However, signals collected by the sensor contain a lot of noise, which brings challenges to signal analysis [3]. This is because mechanical equipment is a complex system composing of multiple components, and shocks and vibrations caused by other factors are transmitted among different components. In addition, mechanical systems also face interference from the outside world. In the initial stage of the fault, the fault characteristic information is weak, and it is easy to be overwhelmed by noise and difficult to detect [4]. To address these challenges, deep learning-based intelligent diagnostic techniques have received more and more attention from academia and industry. The advantage of deep learning is that it can learn a set of good fault-related features from big dataset and can automatically diagnose the health status of mechanical equipment. In recent years, a variety of deep learning algorithms have been extensively studied and applied to machinery fault diagnosis [5–8]. For instance, Liu et al. [5] proposed an improved autoencoder based on recurrent neural network (RNN) for bearing fault diagnosis. Zhou et al. [8] proposed an improved generative adversarial network to solve the problem of sample imbalance. In particular, the convolutional neural network (CNN) stands out among these algorithms and obtains the state-of-the-art performance in a variety of fault diagnosis tasks [9–12]. For instance, Liu et al. [13] proposed a lightweight multi-task CNN architecture for fault diagnosis and condition monitoring of wheelset bearings. Chen et al. [14] Combined CNN with extreme learning machine, and achieved good results on multiple datasets. Han et al. [15] proposed a CNN architecture for vibration signal denoising. This method has good denoising performance on bearing dataset and can improve the diagnostic performance of the CNN model in noisy environments.

Although CNN has achieved good performance in many tasks, it still has the following problems.

1) CNN mainly consists of finite filters, and they are learned by a stochastic gradient descent algorithm under random initial conditions. Compared with methods such as wavelet transform, CNN is quite rough in terms of signal processing. This rough method requires a large number of parameters to obtain sufficient feature expression abilities.
2) It is becoming more and more important in practical industrial applications to understand how an algorithm learns and what it learns. However, CNN has a large number of parameters that are difficult to analyze, so it is still used as a black-box model and difficult for CNN to perform an in-depth interpretability analysis.
3) In practical applications, noise is inevitable. According to our experiments and literature reports [15], the CNN algorithm is susceptible to noise, that is, noise seriously affects the performance of the CNN model. The worthless noise will conceal the valuable information of the signal, causing the algorithm to overfit the invalid features.

To enhance the interpretability of CNN in intelligent diagnosis, many efforts have been made. Wang et al. [16] used the attention method to explore the feature learning mechanism of CNN, and similarly, Yang et al. [17] combined gate recurrent unit (GRU) with the attention mechanism and then analyzed the interpretability of the neural network through attention. Zhou et al. [18] proposed a partially interpretable neural network for fault diagnosis of gas turbines. Since traditional signal analysis methods have a sound theoretical basis and clear physical meaning, we believe that using them to improve the interpretability of CNNs is a very feasible solution. Li et al. [19] designed a continuous wavelet convolutional layer to replace the first convolutional layer of the standard CNN, which enabled the CNN to gradually incorporate the advantages of wavelet transform and its good interpretability. Further, our study deeply explores the fusion architecture of CNN and wavelet transform and proposes a deep hybrid architecture of wavelet and convolution. The proposed method is no longer limited to the first layer of the network; a hybrid architecture of wavelet and convolutional layer runs through the entire neural network framework. Convolutional layer and wavelet transform are alternately performed layer by layer.

Based on the above discussion, this paper aims to rethink the feature learning of CNN architecture from the wavelet domain. Compared with the time domain, the frequency domain or wavelet domain has always been an excellent space for signal processing problems in the field of fault diagnosis, such as signal denoising and fault feature extraction [20]. Compared with randomly initialized convolution kernels, wavelet analysis has excellent feature learning efficiency and interpretability. Wavelet analysis is also a powerful tool for signal denoising. Therefore, we believe that integrating the discrete wavelet transform (DWT) in the whole network can not only improve the model's shortcomings in signal analysis and reduce the number of its parameters, but also make the model have the ability to filter noise and invalid information and have good noise robustness. In addition, the excellent theoretical foundation of wavelet analysis also lays a good foundation for the interpretability of the model.

Therefore, this paper attempts to expand the learning space from the time domain space to the more useful wavelet domain space, so as to break through the current bottleneck of CNN in performance and interpretability. To this end, this paper proposes a discrete wavelet attention layer (DWA-Layer). DWA-Layer extends the feature learning space to the wavelet domain space, which uses discrete wavelet transform to decompose the signal into multiple frequency components. The frequency attention mechanism (FAM) is proposed to further enhance the ability of DWA-Layer to learn valuable information and filter irrelevant noise. DWA-Layer can be embedded in any position of the CNN architecture to jointly participate in model training and gradient update. Inspired by wavelet packet analysis algorithms, CNN-based feature learning and DWT-based signal decomposition are performed alternately, and the fault-related features hidden in the time domain and wavelet domain are learned layer by layer. This is equivalent to embedding a similar wavelet packet decomposition algorithm in the CNN architecture, which is usually used for multi-resolution analysis. Finally, a novel CNN framework based on multilayer wavelet attention (MWA-CNN) for mechanical fault diagnosis is proposed. MWA-CNN is dedicated to learning discriminative fault features from the wavelet domain. It uses efficient wavelet transform to improve the deficiencies of CNN in signal processing; relies on wavelet analysis with domain knowledge to significantly enhance the interpretability of the algorithm; adaptively filters irrelevant information through frequency attention mechanism to retain valuable features.

Additionally, one research approach is to integrate CNN with an independent signal preprocessing step for fault diagnosis [21–24]. Although these methods can improve the performance of CNN to some extent, they cannot solve the shortcomings of deep models. The ideas of this study are fundamentally different from them. In this study, wavelet transform and CNN are deeply fused, and they are alternately integrated in the whole network. This means that the wavelet transform and convolutional layers jointly participate in model training and gradient update. The update of CNN parameters is affected by the result of wavelet transform, and the result of wavelet transform depends on the current parameters of CNN. They promote each other during model training and jointly learn valuable information from the signals.

The proposed method is verified on the high-speed aeronautical (HSA) bearing and motor bearing datasets, and the experimental results show that it significantly improves the diagnostic performance of the CNN model. The proposed MWA-CNN has good interpretability and has great practical application potential.

The contributions of this paper are summarized as follows:

1) This paper explores the expansion of the feature learning space of the diagnostic model to the wavelet domain space. The wavelet domain space can provide inherent advantages that the time domain space does not have, and brings new insights into the feature learning and interpretability of CNN.
2) This paper implements a wavelet-transform-based layer (DWA-Layer), which uses DWT to map the time domain space to the wavelet domain space, and proposes an attention mechanism to learn valuable frequency domain information. DWA-Layer can be embedded in any position of the deep learning model, and jointly participate in model training and gradient update.
3) This paper proposes a signal-processing-based CNN framework (MWA-CNN) that embeds DWT into deep learning model for machinery fault diagnosis. The framework deeply integrates DWT and convolutional layers, and the feature flow is processed and learned in the network model in a novel form.
4) This paper analyzes the interpretability of the proposed method, and the results show that it learns fault-related features layer by layer and suppresses irrelevant information.

The paper is organized as follows. Section II introduces the basic theory. Section III describes the DWA-Layer and MWA-CNN in detail. In Section IV, the effectiveness and superiority of DWA-Layer and MWA-CNN is verified. Section V discusses four aspects of MWA-CNN. Section VI analyzes the interpretability of the proposed method. Section VII summarizes this paper.

2. Basic theory

2.1. Wavelet transform

Let $\psi(t)$ be a square integrable function, that is,$\psi(t) \in L^2(R)$. A family of functions can be obtained from $\psi(t)$ after scaling and translation transformation.

$$\psi_{\tau,\upsilon}(t) = \frac{1}{\sqrt{\tau}}\psi(\frac{t-\upsilon}{\tau}),\ \tau,\upsilon \in R,\ \tau \neq 0 \tag{1}$$

$\{\psi_{\tau,\upsilon}\}$ is called continuous wavelet, ψ is basic wavelet or mother wavelet and has $\widehat{\psi}(t=0) = 0$. τ is the scaling factor, and υ is the translation factor. The scaling factor changes the shape of the continuous wavelet. The translation factor changes the displacement of the continuous wavelet.

For the initial signal $x(t) \in L^2(R)$, its continuous wavelet transform (CWT) is defined as:

$$W_x(\tau,\upsilon) = \langle x, \psi_{\tau,\upsilon}\rangle = \frac{1}{\sqrt{\tau}}\int_{-\infty}^{+\infty} x(t)\overline{\psi}(\frac{t-\upsilon}{\tau})dt \tag{2}$$

where $\overline{\psi}(t)$ represents the complex conjugate of $\psi(t)$. Both τ and υ are continuous variables. It can be seen that $W_x(\tau,\upsilon)$ represents the projection of the signal $x(t)$ on the wavelet basis function $\psi(t)$.

By discretizing the scaling factor τ and the translation factor υ, the DWT can be obtained. In general, τ and υ are defined as:

$$\tau = \tau_0^i,\ \upsilon = j\upsilon_0\tau_0^i \tag{3}$$

where i and j are integers. The discrete wavelet is defined as:

$$\psi_{\tau,\upsilon} = \frac{1}{\sqrt{\tau_0^i}}\psi(\frac{t-j\upsilon_0\tau_0^i}{\tau_0^i}) = \tau_0^{-i/2}\psi(\tau_0^{-i}t - j\upsilon_0) \tag{4}$$

The corresponding DWT is:

$$W_x(\tau_0^i, j\upsilon_0\tau_0^i) = \tau_0^{-i/2}\int_{-\infty}^{+\infty} x(t)\psi(\tau_0^{-i}t - j\upsilon_0)dt \tag{5}$$

2.2. Multi-Resolution analysis and wavelet packet analysis

Multi-resolution analysis was first proposed by Mallat [25] in 1989, and it provides a new way to implement wavelet analysis algorithms. Let $X(n)$ be the discrete sequence of signal $x(t)$, where $n = 1, 2, \ldots, N$. If the decomposition scale $\eta = 0$, the signal can be expressed as $X(n) = A_0(n)$. Then the decomposition algorithm can be expressed as:

$$\begin{cases} A_\eta(n) = \sum_{k\in Z} h(k-2n) A_{\eta-1}(k) \\ D_\eta(n) = \sum_{k\in Z} g(k-2n) A_{\eta-1}(k) \end{cases} \tag{6}$$

where $h(n)$ and $g(n)$ are the filter coefficients determined by the wavelet basis function. $h(n)$ and $g(n)$ have low-pass and high-pass properties, respectively. η is the number of decomposed layers. The wavelet decomposition tree for multi-resolution analysis is shown in Fig. 1(a).

The discrete sequence A_0 is decomposed into two signal components through a low-pass filter and a high-pass filter, and then sampled at intervals, and finally the low-frequency component A_1 and the high-frequency component D_1 of the signal are obtained. A_1 is selected to repeat the decomposition steps described above to complete the multi-resolution decomposition of the signal. It can be seen that multi-resolution analysis is actually equivalent to multiple band-pass filters. Only low-frequency components are decomposed in each layer, and high-frequency components are not considered. One advantage of wavelet packet analysis is that it decomposes the frequency of the signal in multiple levels. It not only decomposes the low-frequency components of each layer, but also decomposes the high-frequency components of each layer. The result of a 3-layer wavelet packet decomposition is shown in Fig. 1(b). It can be seen that D_1 is also decomposed into its low frequency component AD_2 and high frequency component DD_2. Then, AD_2 and DD_2 are further decomposed.

3. Multilayer wavelet attention CNN

3.1. Discrete wavelet attention Layer

DWT is an effective signal processing technique, which can decompose a signal into two wavelet coefficients, namely low-frequency component and high-frequency component. Further decomposing these two wavelet coefficients can display the information of different frequency bands of the signal. This paper proposes a novel discrete wavelet attention layer (DWA-Layer). DWA-Layer introduces advantages of wavelet transform to the CNN model, so that the improved CNN model can not only learn the information hidden in the time domain and frequency domain, but also has good interpretability. In order to facilitate understanding, we will first introduce the proposed discrete wavelet transform layer (DW-Layer), and then introduce its improved version: DWA-Layer. Finally, the wavelet-based error propagation is analyzed.

3.2. Discrete wavelet transform Layer

In the CNN model, the data stream is generally characterized in the form of feature maps. Assuming that $M \in \Re^{C\times L}$ is the feature map of one-dimensional CNN (1DCNN), then M is a two-dimensional matrix with length L and width C. C represents the number of channels, and L represents the length of feature signals. That is, for the feature map M, it contains C feature signals. Based on the given wavelet basis function, a low-pass filter $h(n)$ and a high-pass filter $g(n)$ can be obtained, and $g(k) = (-1)^k h(1-k)$. It can be seen from Eq. (8) that after each signal is filtered by $h(n)$ and $g(n)$, its low-frequency and high-frequency components can be finally obtained. In 1DCNN model, DW-Layer decomposes each feature signal independently. Assuming that the feature map M is the input of DW-Layer, the output of DW-Layer is the low-frequency feature map $LM \in \Re^{C\times L'}$ and the high-frequency feature map $HM \in \Re^{C\times L'}$ respectively. Generally, $L' = L/2$, and the value of L' varies slightly when different wavelet basis functions are used. Subsequently, LM and HM are spliced into a whole according to the channel to obtain the mixed feature $LG \in \Re^{2C\times L'}$. Compared with learning from the feature map M, the CNN model can learn richer feature information from the wavelet domain feature map.

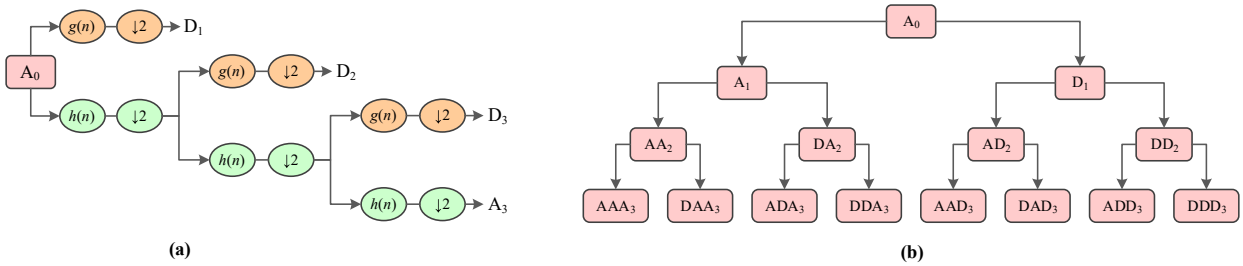


Fig. 1. The decomposition structure of multi-resolution analysis (a) and wavelet packet analysis (b).

Dropout, as an effective regularization method to suppress over-fitting, is usually used in the fully connected layer. We found that dropout technology also has excellent anti-overfitting ability in DW-Layer. Because the existence of DW-layer makes it easier for CNN model to learn features from the signal, the overfitting phenomenon of CNN model becomes serious. Dropout technology can solve this problem well. Therefore, after each DW-Layer, Dropout is used to further improve the generalization performance of the CNN model.

Although DW-Layer can decompose the input feature signal, filtering out the fault-related feature from the obtained results is still a complex problem. In conventional signal analysis methods, the recognition of features can be achieved through some indicators, but this is difficult to achieve in the CNN architecture. Therefore, this paper proposes DWA-Layer to make the CNN model automatically pay attention to important feature information and reduce the model's attention to irrelevant information (noise, etc.). DWA-Layer introduces the frequency attention mechanism (FAM) to aggregate the global information of the mixed feature *LG*, and then encodes the relative importance of different channel features to guide the feature learning of the CNN model.

3.3. Frequency attention mechanism

As shown in Fig. 2, DWA-Layer is mainly composed of a DW-Layer and a FAM. The core idea of FAM is to use the self-learning ability of CNN to filter out the signal components that are useful for the diagnosis task from the mixed feature $LG \in \Re^{2C\times L'}$. It replaces manual feature selection with an automatic valuable feature attention mechanism similar to human visual attention. First, FAM introduces a global average pooling layer (GAP) [26] to compress the global information of the mixed feature *LG* into a channel representation vector $z \in \Re^{2C\times 1}$. The *i*-th element of **z** is expressed as:

$$z_i = \mathrm{GAP}(LG_i) = \frac{1}{L'}\sum_{j}^{L'} LG_i(j) \tag{7}$$

LG_i represents the *i*-th feature signal of *LG*. Then, FAM uses a simple encoding and decoding mechanism to capture the importance of these channel signals. The channel representation vector is first compressed into a hidden layer vector with a dimension of $C \times 1$, and then decoded back to the original dimension. Finally, the channel weight vector $z' \in \Re^{2C\times 1}$ is output. The encoding and decoding operations of FAM are completed by two convolutional layers respectively. The first convolutional layer uses the ReLU function to provide non-linear transformation capabilities. The second convolutional layer uses the Sigmoid function, which is mainly used to map the obtained feature vector to an interval range of 0 to 1, thereby generating a weight vector z'. The size of the element of z' represents the importance of the corresponding channel feature. This can be expressed as:

$$\begin{aligned} \hat{z} &= \mathrm{ReLU}(\boldsymbol{W}_1 * z + b_1),\ \hat{z} \in \Re^{C\times 1}, \\ z' &= \mathrm{Sigmoid}(\boldsymbol{W}_2 * \hat{z} + b_2),\ z' \in \Re^{2C\times 1}. \end{aligned} \tag{8}$$

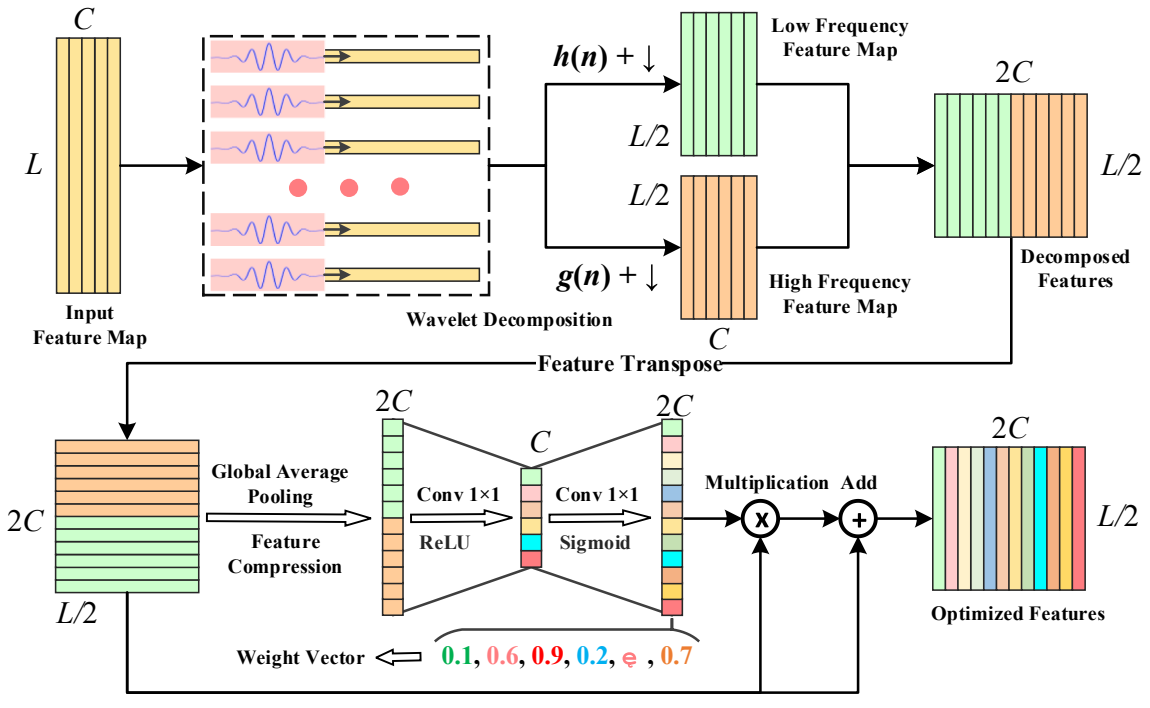


Fig. 2. The architecture details of the proposed DWA-Layer, which mainly consists of a DW-layer and a frequency attention mechanism.

Finally, FAM uses matrix multiplication to embed this weight information into the CNN model to guide the feature learning of the CNN model. In addition, residual connections are added in FAM to optimize the gradient propagation of the network and prevent feature responses from being too small.

3.4. Wavelet-Based backpropagation

This work deeply integrates DWT and convolution to jointly participate in optimization and gradient propagation. Wavelet-driven backpropagation is described below. Assume the error passed to the output of LG is $\xi_{LG}^{\varphi} \in \mathbb{R}^{2C\times L'_2}$, and the low-pass and high-pass filters for the DWT are h^{η},g^{η}, φ denotes the φ_{th} layer. Since in the forward operation, the output of the DWT is concatenated according to their channels, then the backpropagation firstly needs to spilt ξ_{LG}^{φ} into two parts:ξ_{LM}^{φ}, $\xi_{HM}^{\varphi} \in \mathbb{R}^{C\times L'_2}$. ξ_{LM}^{φ} and ξ_{HM}^{φ} correspond to the low-frequency and high-frequency output of the DWT. Note that the split is needed to take the same sequence as the forward concatenation.

Before error is back-propagated, ξ_{LM}^{φ} and ξ_{HM}^{φ} are firstly up-sampled by a factor of two, since the forward computation is performed by a convolution with a step of two:

$$\widetilde{\xi}_{LM,HM}^{\varphi}(n) = \begin{cases} \xi_{LM,HM}^{\varphi}(n/2), & n/2 \text{ are integers} \\ 0, & otherwise \end{cases} \tag{9}$$

Then error is passed to the input of DWT by:

$$\xi_{\Im}^{\varphi-1} = \widetilde{\xi}_{LM}^{\varphi} * reverse(h^{\eta}) + \widetilde{\xi}_{HM}^{\varphi} * reverse(g^{\eta}) \tag{10}$$

3.5. Multilayer wavelet attention convolutional neural network

The CNN model has good automatic feature learning ability, and DWT has good signal decomposition ability. The effectiveness of these two technologies in fault diagnosis tasks has been verified by numerous studies. This paper attempts to combine the advantages of these two technologies to propose a novel machinery fault diagnosis model. Fig. 3 shows the proposed Multilayer Wavelet Attention CNN. MWA-CNN is mainly composed of multiple convolutional layers and DWA-Layer. The collected signal is first decomposed by

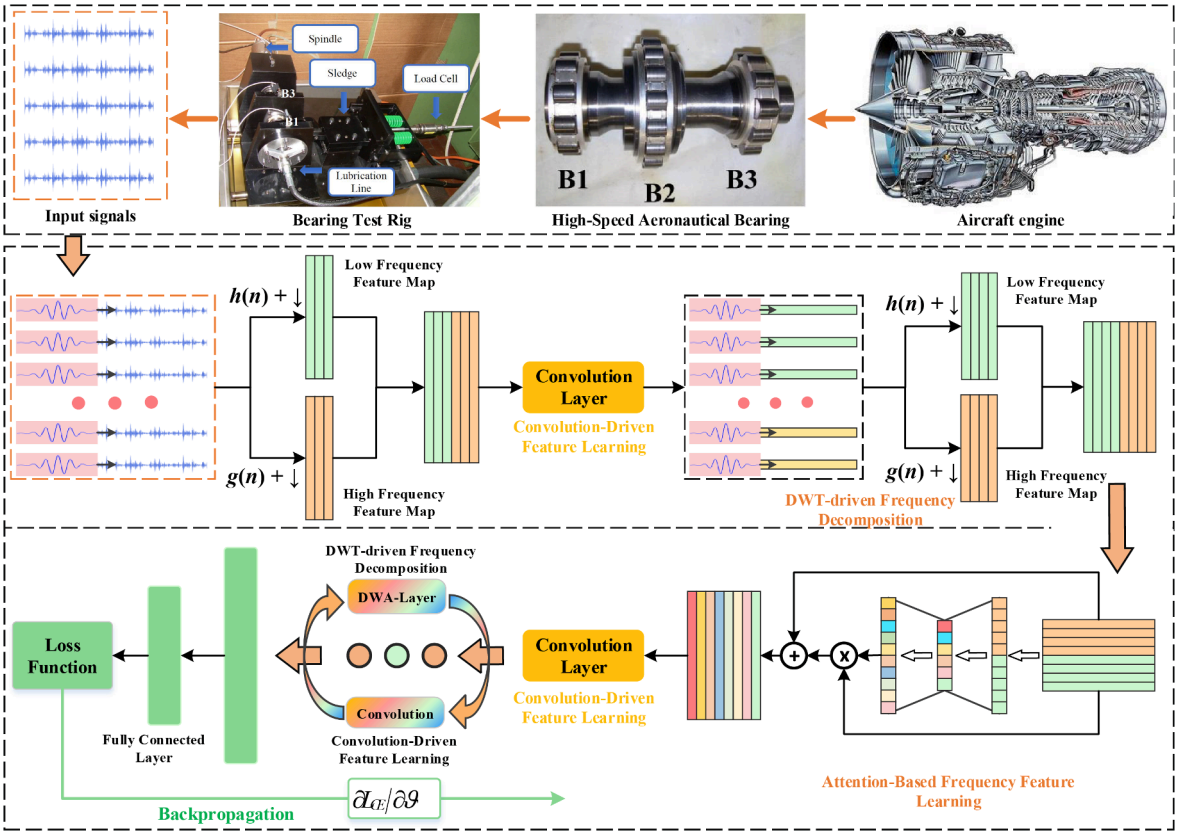


Fig. 3. The detailed network architecture of the proposed multilayer wavelet attention CNN (MWA-CNN).

DWA-Layer, and then input into the convolutional layer for automatic feature learning. In the entire MWA-CNN, the above operations are performed multiple times. This means that the input signal is decomposed layer by layer into multiple frequency components, and the required feature information is also learned layer by layer by the convolutional layer. In addition, this paper recommends using the proposed DWA-Layer to replace MaxPooling. DWA-Layer performs DWT decomposition of the signal while performing feature downsampling, so as to display the hidden features in the time domain and frequency domain to the convolutional layer. This paper uses the group normalization (GN) layer [27] to replace the batch normalization (BN) layer. The GN layer divides the channels into groups and calculates the mean and variance within each group for normalization. Although the performance of GN is similar to that of BN when using a large batch size, the performance of GN is significantly better than BN when using a small batch size. In fault diagnosis, obtaining enough labeled samples to train the model in many scenarios is difficult. In this scenario, GN will perform better than BN. Therefore, adopting GN can give the proposed method broad applicability. The training algorithm of MWA-CNN is shown in **Algorithm. 1**. $D = \left\{\Gamma^i, j^i\right\}_{i=1}^{N}$ represents the dataset, Γ^i represents the data sample, and its corresponding label is.

Generally, the fault diagnosis task can be regarded as a multi-classification task. Therefore, Softmax is adopted, which transforms the input features into a probability distribution whose sum is 1. Assuming that the output of the fully connected layer is a logit vector ϖ, χ represents the input signal of the CNN model, and the total number of fault categories is S.

$$p(s|\chi) = \frac{\exp(\varpi_s)}{\sum_{s=1}^{S}\exp(\varpi_s)} \tag{11}$$

$p(s|\chi)$ is the predicted probability that χ belongs to category s. Suppose $q(s|\chi)$ is the true probability that χ belongs to category s. The cross-entropy loss function is used to measure the distance between the prediction and the true label, which can be expressed as:

$$L_{CE} = -\sum_{s=1}^{S} q(s|\chi)\log p(s|\chi). \tag{12}$$

Algorithm 1 Multilayer Wavelet Attention Convolutional Neural Network

Input: Dataset $D = \left\{\Gamma^i, j^i\right\}_{i=1}^{N}$ trained by mini-batch
Output: Optimized Predictor $F(v)$
1: Set $e = 0$ and epoch E; set H.
2: Initialize the ϑ; Normalize the dataset D
3: **while** $e \le E$ **do**
4: Sample a batch of data B from D; set $\hbar = 1$.
5: Calculate DWT according to Eq. (8), obtained A_1 and D_1, then splice them.
6: Perform CNN to extract features:$F_1^C = \mathrm{Re}LU\left(W^T(A_1 + D_1) + b\right)$
7: **while** $\hbar \le H$ **do**
8: Calculate DWT according to Eq. (8), obtained A_h and D_h
9: Get mixed frequency features:$LG = (A_h + D_h)$
10: Perform attention to filter frequency features:$z^{'} \otimes LG + LG$
11: Perform CNN to extract features:$F_h^C = \mathrm{Re}LU\left(W^T(z^{'} \otimes LG + LG) + b\right)$
12: Set $\hbar = \hbar + 1$
13: Calculate the output of
14: Calculate, update by back propagation
15: If an epoch is completed, set $e = e + 1$
16: **end while**

Table 1
The Parameter Configuration of MWA-CNN.

Layer	Type	Kernel	Channel	Output
1	Input Layer	N/A	N/A	4096 × 1
2	DWA-Layer	N/A	N/A	2048 × 2
3	CNN-Layer	3 × 1	12	2048 × 12
4	DWA-Layer	N/A	N/A	1024 × 24
5	CNN-Layer	3 × 1	24	1024 × 24
6	DWA-Layer	N/A	N/A	512 × 48
7	CNN-Layer	3 × 1	48	512 × 48
8	DWA-Layer	N/A	N/A	256 × 96
9	CNN-Layer	3 × 1	96	256 × 96
10	DWA-Layer	N/A	N/A	128 × 192
11	CNN-Layer	3 × 1	192	128 × 192
12	DWA-Layer	N/A	N/A	64 × 384
13	CNN-Layer	3 × 1	384	64 × 384
14	Global Average Pooling			384
15	FC + Softmax			7

Table 1 shows the parameter configuration of MWA-CNN. When MWA-CNN is applied to the HSA bearing dataset, the input dimension of MWA-CNN is 4096 × 1. DWA-Layer is used to replace the MaxPooling layer to complete feature dimensionality reduction. Not only that, DWA-Layer can also perform multi-resolution signal decomposition on the input signal. The output dimension of each layer varies slightly depending on the wavelet basis function and the padding. Assuming the Haar wavelet is used, the output dimension of the DWA-Layer is half of the input dimension. MWA-CNN consists of six CNN-Layers and six DWA-Layer. The number of CNN-Layers and DWA-Layer can be simply adjusted to suit different tasks. A CNN-Layer contains a convolutional layer, a GN layer, and a ReLU activation function. The kernel of the convolutional layer is set to 3 × 1, the number of channels gradually increases from 12 to 384, and the padding is set to 0. The classification layer of MWA-CNN consists of a GAP and a fully connected layer with a Softmax function. The open-source code of the proposed method can be found at ***https://github.com/PHM-Code/MWA-CNN***.

3.6. Architecture design combining wavelet and CNN

In this study, we deeply integrate wavelet transform and CNN, so that they can learn from each other and jointly promote each other in a unified model. In the proposed MWA-CNN, convolutional layers and wavelet transform are performed alternately to jointly build a deep neural architecture. This actually builds a deep frequency feature decomposition and feature learning mechanism. With the deepening of the network depth, the frequency features are gradually decomposed in detail, and valuable features are also learned layer by layer by the convolution layer.

To highlight the characteristics of the proposed method, Fig. 4 shows different architectural design methods combining wavelet transform and CNN. **(1) Ordinary CNN Network.** It follows the end-to-end design concept, and its network architecture is stacked by multiple convolutional layers. The approach does not use signal analysis methods. **(2) CNN with Signal Analysis** [21–24]. It tried to combine CNN with signal analysis methods such as wavelet transform or Fourier transform. This method first uses signal analysis techniques to preprocess the signal and then uses the CNN model for feature learning and fault identification. This method only uses wavelet transform as a data preprocessing method, and CNN and wavelet transform are not perfectly combined. Most existing methods combining wavelets and CNNs can be classified into this category [28,29]. **(3) Wavelet-Kernel-Net** [19]. It designs a continuous wavelet convolution (CWConv) layer to replace the first convolutional layer of standard CNN. This enables the first CWConv layer to discover more meaningful kernels. This method fully integrates CNN and wavelet transform, but is limited to the first layer of the model. **(4) Our Proposed Method.** It deeply integrates wavelet transform and convolutional layers, which enables the proposed method to perfectly fuse the advantages of both to achieve effective fine-grained signal decomposition and fault-related feature learning. Moreover, an attention mechanism is proposed to enable the model to distinguish valuable frequency features and ignore irrelevant information.

As shown in Fig. 5, from the perspective of signal analysis, our architecture design draws on the ideas of multi-resolution analysis and wavelet packet decomposition. Moreover, unlike wavelet packet decomposition, MWA-CNN has self-learning ability. The convolutional layer is placed after the signal decomposition layer to learn fault-related features. The convolutional layer is actually used as a feature selector, which sends the useful feature signals of the previous layer to the next layer (DWA-Layer) for more detailed decomposition. As the number of layers increases, the signal is decomposed more finely by DWA-Layer, and the convolutional layer can learn the feature information hidden in the time domain and frequency domain well. Through layer-by-layer signal decomposition and feature learning, irrelevant information (noise, etc.) is gradually removed, so that discriminative fault features can be learned by MWA-CNN.

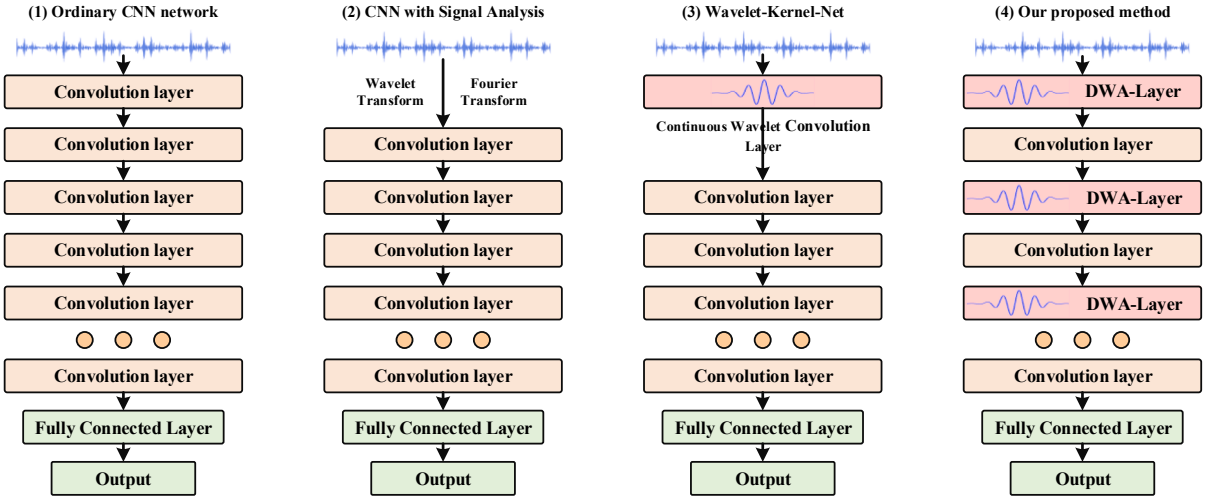


Fig. 4. The architectural design methods combining wavelet transform and CNN.

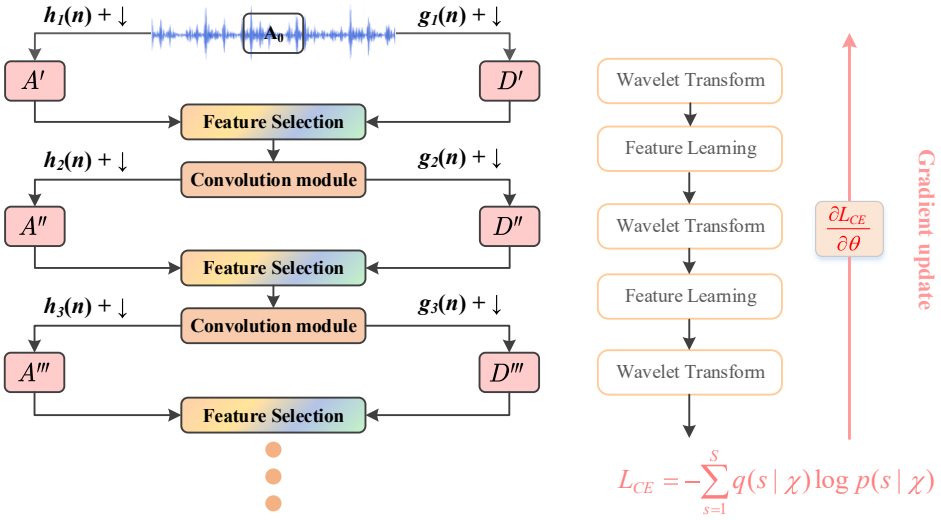


Fig. 5. This illustrates the architectural design idea of the proposed method.

4. Experimental validation

In this section, the effectiveness of the proposed DWA-Layer and MWA-CNN is verified on the HSA bearing dataset and the motor bearing dataset. In addition, MWA-CNN is compared with cutting-edge deep learning methods. We also proved that MWA-CNN has good noise robustness.

4.1. Experimental setup

The proposed method is implemented by deep learning framework Pytorch, python 3.7 and pytorch_wavelets [30]. All models are trained and tested on a server with NVIDIA GeForce RTX 3090 GPU. The server has 32 GB memory and uses Intel 10900 K CPU. Z-score normalization is used to normalize all data samples so that the training of the model becomes stable. In the training process, the Adam optimization algorithm is adopted, which has the advantages of fast calculation efficiency and small memory requirements. Adam can also accelerate the convergence speed of the network model. The batch size is set to 64 and the learning rate is set to 0.0001. Since Daubechies (db) wavelet is widely used in fault diagnosis tasks, this paper takes db16 wavelet as an example for experiment. It is worth

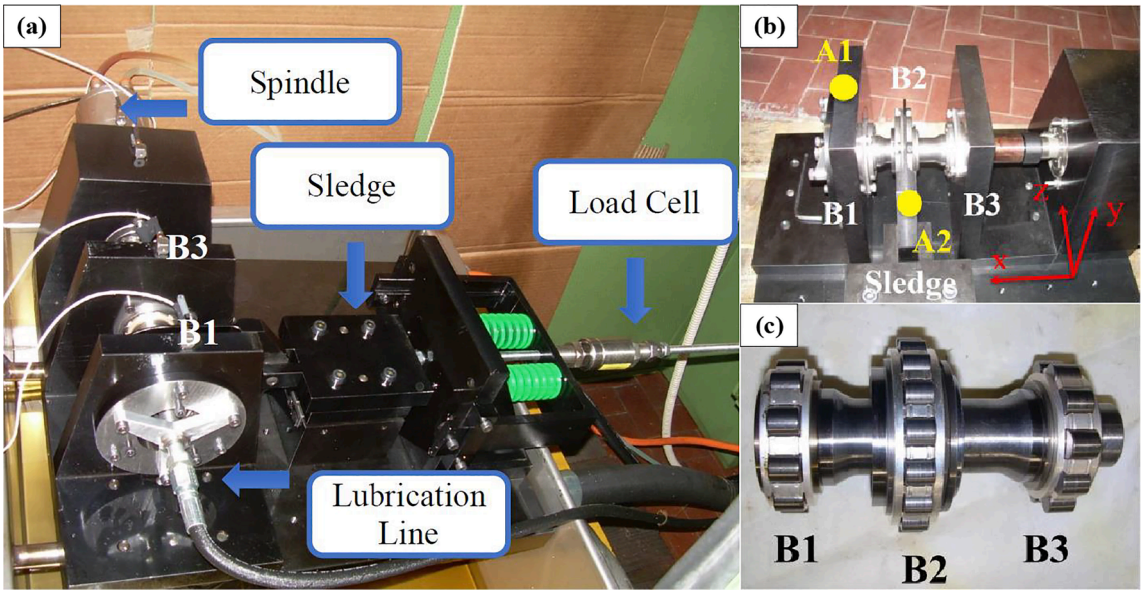


Fig. 6. The HSA bearings test rig a) general view of the test rig; b) positions of the accelerometers and the reference system; c) the shaft with its three roller bearings.

noting that the developed method is also applicable to other wavelets, such as Haar, Biorthogonal, Dmeyer, etc. This is discussed in Section V. This paper uses accuracy to evaluate the performance of the network model. Each experiment was repeated four times, and the mean and standard deviation were used as the final experimental results.

The collected vibration signal already contains a certain degree of noise. However, in real situations, the vibration signal may contain higher levels of noise. In order to verify the performance of MWA-CNN in different noise environments, we add additional Gaussian white noise to the vibration signal. In this paper, the Gaussian white noise with SNR = 4 dB, 0 dB, −2dB, and − 4 dB is added to the signal, respectively.

B. Case 1. ***C. High-Speed Aeronautical Bearings Fault Diagnosis.***

4.2. Data Description

The experimental data comes from the HSA bearing signal acquisition test rig [31]. Fig. 6 shows the structure of the test rig, the test bearing and the location of the acceleration sensor. The test rig is mainly composed of a high-speed spindle, which is used to drive the rotation of the shaft. As shown in Fig. 6(b), a triaxal IEPE accelerometer is installed at A1 and A2, and the sampling frequency is set to 51200 Hz. As shown in Fig. 6(c), the inner rings of these bearings (B1, B2, and B3) are connected to a very short and tick hollow shaft, specifically designed for speeds up to 35,000 rpm. As shown in Table 2, when collecting signals, seven health states are set on the B1 bearing. A total of one health state and six failure states. There are mainly two types of faults: inner ring fault and roller fault. These two kinds of faults have three different fault sizes, and their fault diameters are 150, 250 and 450 μm respectively. The experiment used HSA bearings, so the experiment was carried out under different loads and speeds. See Table 2 for details, where 100 Hz means 6000 rpm. In this experiment, we use the sliding segmentation method for data enhancement. The length of the signal sample is set to 4096 × 1, and a total of 22,134 training samples and 7259 test samples are obtained.

4.3. The effectiveness of DWA-Layer

This experiment compared four different network architectures. 1DCNN: this architecture only uses six convolutional layers and a classification layer with GAP. The parameter configuration of the convolutional layer and classification layer is the same as that of MWA-CNN. W-CNN: this architecture adds a DW-Layer before 1DCNN. MW-CNN: the parameter configuration of this architecture is the same as that of MWA-CNN, but all attention modules are removed. MWA-CNN: the method described in Section III. In order to illustrate the performance of these methods under different noise conditions, the experiment was carried out under four noise conditions (4 dB, 0 dB, −2dB and − 4 dB). Table 3 shows their experimental results.

As shown in Table 3, 1**DCNN** obtains an accuracy of 91.16% when SNR = 4 dB, however, when SNR = − 4 dB, it only obtains an accuracy of 50.38%. This shows that the ordinary CNN model can obtain high accuracy when there is no noise or less noise, but when the noise is strong, the performance of the ordinary CNN model is greatly reduced. This phenomenon has been reported by many studies. Therefore, improving the anti-noise ability of the CNN model is also the focus of this paper. After adding a DW-Layer before 1DCNN, we found that the performance of the model is slightly improved under all noise conditions. This shows that adding DWT is helpful to improve the performance of the CNN model. Furthermore, we add multiple DW-layers to the 1DCNN model, which is MW-CNN. The diagnostic performance of MW-CNN has been greatly improved compared to 1DCNN and W-CNN. For example, when SNR = 4 dB, MW-CNN obtains a diagnosis accuracy of 95.08%, which is 3.92% higher than 1DCNN. When SNR = − 4 dB, the diagnostic accuracy of MW-CNN is 70.66%, which is 20.28% higher than 1DCNN. This shows that the proposed method is effective, it integrates the advantages of CNN method and DWT technology, so that the network model has very good noise robustness. Then, we use the self-learning ability of the CNN model and introduce the attention mechanism to make the network automatically select useful frequency component information. Finally, MWA-CNN is proposed. MWA-CNN demonstrates excellent fault diagnosis performance. For example, when SNR = 4 dB, MWA-CNN obtains an accuracy of 98.75%. When SNR = − 4 dB, MWA-CNN still achieves 87.61% accuracy, which is 16.95% higher than MW-CNN. This shows that using attention to make the CNN model focus on useful feature information and ignore irrelevant information (such as noise) can greatly improve the anti-interference ability of the CNN model.

Fig. 7 and Fig. 8 show the training curves of 1DCNN, MW-CNN and MWA-CNN when SNR = 4 dB and SNR = − 4 dB. When SNR = 4 dB, the training accuracy and validation accuracy of these three network models increase rapidly, and then tend to be stable when epoch greater than 80. However, the validation accuracy of MW-CNN and 1DCNN is significantly lower than the training accuracy, indicating that there is a serious overfitting. When SNR = − 4 dB, the convergence speed of MW-CNN and 1DCNN is obviously slower,

Table 2
Description of the HSA Bearing Dataset Information.

Defect	Dimension	Load	Speed	Label
No defect	–	0 N-1800 N	100 Hz-500 Hz	F1
Diameter of an indentation on the inner ring	450 μm	0 N-1800 N	100 Hz-500 Hz	F2
Diameter of an indentation on the inner ring	250 μm	0 N-1800 N	100 Hz-500 Hz	F3
Diameter of an indentation on the inner ring	150 μm	0 N-1800 N	100 Hz-500 Hz	F4
Diameter of an indentation on a roller	450 μm	0 N-1800 N	100 Hz-500 Hz	F5
Diameter of an indentation on a roller	250 μm	0 N-1800 N	100 Hz-500 Hz	F6
Diameter of an indentation on a roller	150 μm	0 N-1800 N	100 Hz-500 Hz	F7

Table 3
The Experimental Results of 1DCNN, W-CNN, MW-CNN, and MWA-CNN Under Four Kinds of Noise.

Noise	1DCNN	W-CNN	MW-CNN	MWA-CNN
4 dB	91.16 ± 0.32	91.41 ± 0.96	95.08 ± 0.46	**98.75 ± 0.20**
0 dB	78.54 ± 0.29	79.28 ± 0.48	88.28 ± 0.83	**96.48 ± 0.20**
−2 dB	65.04 ± 1.61	68.45 ± 0.73	80.66 ± 0.33	**93.62 ± 0.38**
−4 dB	50.38 ± 0.64	54.01 ± 1.31	70.66 ± 1.10	**87.61 ± 0.85**

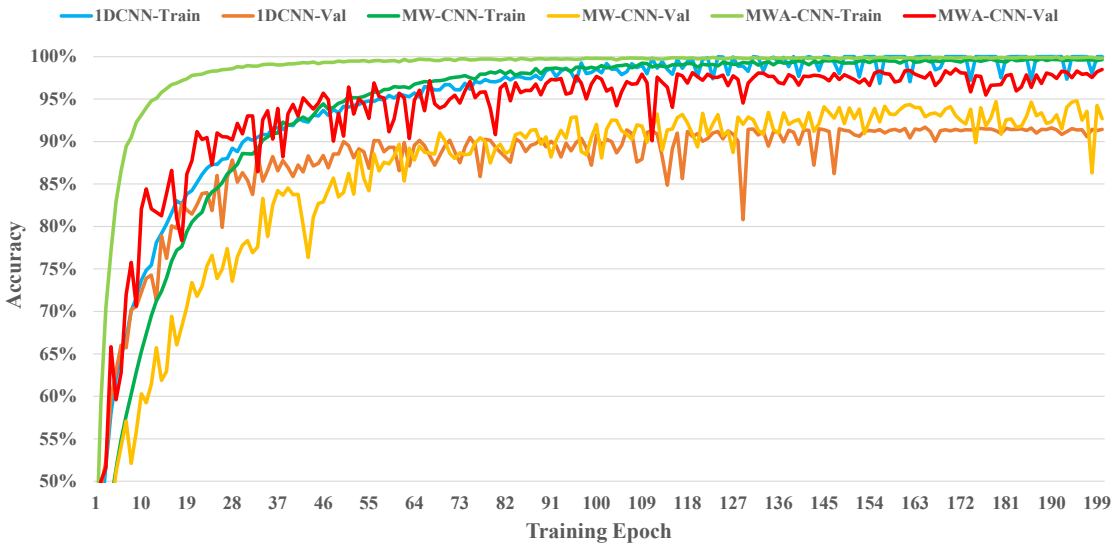


Fig. 7. The training accuracy curves and validation accuracy curves of 1DCNN, MW-CNN and MWA-CNN on the HSA bearing dataset (SNR = 4 dB).

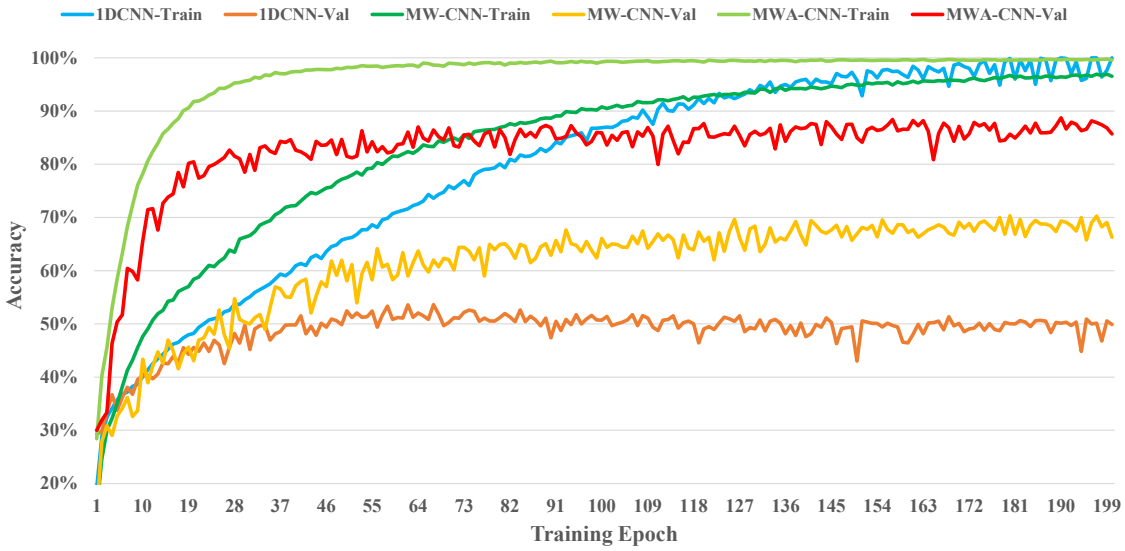


Fig. 8. The training accuracy curves and validation accuracy curves of 1DCNN, MW-CNN and MWA-CNN on the HSA bearing dataset (SNR = −4 dB).

which indicates that it is a great challenge to learn useful features from vibration signals in strong noise environment. The over-fitting phenomenon of MW-CNN and 1DCNN has also become more serious. When epoch = 180, the training accuracy of these two networks can reach more than 95%, and the validation accuracy is about 70% and 50%, respectively. In addition, MWA-CNN also has over-fitting phenomenon, but its situation is much better than MW-CNN and 1DCNN. Its validation accuracy can reach more than 85%.

In addition, we also show the visualization of the features of the fully connected layers of 1DCNN, MW-CNN and MWA-CNN in 2D

space. The visualization result is shown in Fig. 9, which is realized by T-SNE technology. Different colors represent different categories. When SNR = 4 dB, the features of MWA-CNN have very good distinguishability, and different categories are clearly distinguished. The features of 1DCNN and MW-CNN have poor distinguishability. When SNR = − 4 dB, the features of 1DCNN are completely indistinguishable, and samples of different fault categories are mixed together. This leads to poor fault diagnosis performance of the network. This also shows that it is difficult for 1DCNN to learn useful information from the signal under a strong noisy environment. MW-CNN performs better than 1DCNN, but its distinguishability is also poor. The features of MWA-CNN also have good distinguishability, indicating that the attention mechanism can effectively help MWA-CNN learn useful features from the signal.

4.4. Performance comparison

This experiment compares five existing deep learning methods to verify the superiority of the proposed method under noise conditions. These methods are MA1DCNN, WenCNN, ResNet-18, VGG-16 and LSTM. MA1DCNN was proposed by Wang et al. [16] for fault diagnosis of wheelset bearings. MA1DCNN is composed of multiple joint attention modules and convolutional layers, which can optimize the extracted features. WenCNN was proposed by Wen et al. [32] for fault diagnosis of motor bearings. WenCNN converts the time domain signal into a 2D image, and then uses a 2D CNN to extract feature information from the image. ResNet-18 [33] and VGG-16 [34] are two excellent network architectures for image recognition. We modify their 2D convolution into 1D convolution for machinery fault diagnosis. LSTM is an excellent network model, which has excellent long-term feature learning capabilities. These methods all adopt the same training strategy, and the experimental results under four kinds of noise are shown in Table 4.

As Table 4, MWA-CNN has the best performance under the four noise conditions. For example, when SNR = 4 dB, the diagnostic accuracy of MWA-CNN is 98.75%, which is 6.69% higher than MA1DCNN. When SNR = -4dB, the diagnostic accuracy of MWA-CNN is 87.61%, which is 22.8% higher than MA1DCNN. Thanks to the help of the attention mechanism, MA1DCNN has achieved the best performance among the five comparison methods, but there is still a big gap compared to MWA-CNN. In addition, these five comparison methods are very susceptible to noise, resulting in a significant performance degradation. For example, when SNR = 4 dB, MA1DCNN can obtain 94.06% accuracy, but when SNR = − 4 dB, its accuracy is only 64.81%. When SNR = 4 dB, VGG-16 can obtain 91.55% accuracy, but when SNR = − 4 dB, its accuracy is only 48.61%. In particular, when the SNR changes from 4 dB to 0 dB, the accuracy of these five comparison methods all drop by more than 10%. This shows that the anti-noise performance of these methods is insufficient, and it is difficult for them to obtain enough useful feature information from the noisy signal. MWA-CNN uses DWA-Layer to decompose the signal, making it more convenient to obtain useful fault features from the signal, and then obtain excellent diagnostic performance.

Fig. 10 and Fig. 11 show the confusion matrix of MWA-CNN when SNR = 4 dB and SNR = − 4 dB, respectively. The value on the diagonal indicates the number of samples that are correctly classified for each category. The last column indicates the number of test samples for each category. When the noise intensity is small (SNR = 4 dB), good diagnosis results can be obtained for most categories. The diagnosis recall of F1, F2, F3, F6 and F7 are all above 98%, and some are even close to 100%. When the SNR changes from 4 dB to − 4 dB, the recall and precision of each category are greatly reduced. For example, the recall of F1 dropped from 98.94% to 89.30%. The precision of F1 dropped from 99.03% to 88.70%. In particular, the recall of F4 dropped from 94.41% to 81.00%. This shows that this category is easily affected by noise. This may be because the fault diameter of F4 is only 150 μm, which is smaller than that of F2

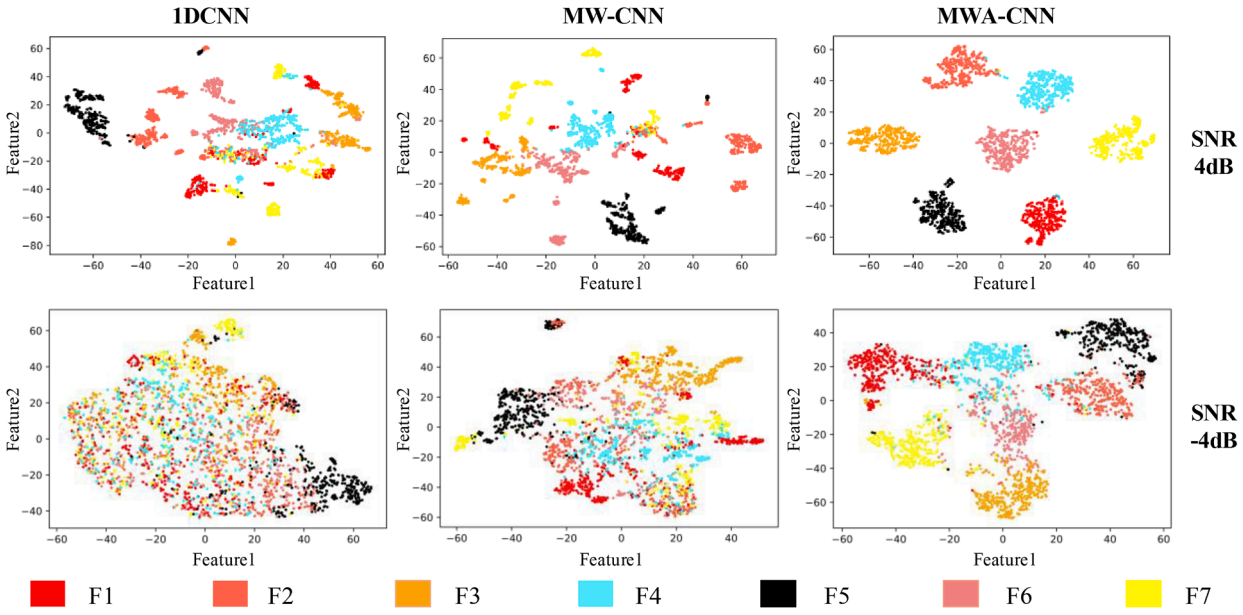


Fig. 9. The visualization of the features of the fully connected layer of 1DCNN, MW-CNN and MWA-CNN in 2D space.

Table 4
The Experimental Results of MWA-CNN and Five Comparison Methods Under Four Kinds of Noise.

Noise	MWA-CNN	MA1DCNN	WenCNN	ResNet-18	VGG-16	LSTM
4 dB	**98.75 ± 0.20**	94.06 ± 0.77	90.44 ± 0.96	85.31 ± 0.67	91.55 ± 0.48	87.23 ± 0.23
0 dB	**96.48 ± 0.20**	84.11 ± 0.90	78.91 ± 0.76	63.58 ± 0.56	79.18 ± 0.42	74.59 ± 1.02
−2 dB	**93.62 ± 0.38**	76.72 ± 0.89	72.10 ± 0.54	48.06 ± 1.03	65.50 ± 0.82	63.43 ± 2.28
−4 dB	**87.61 ± 0.85**	64.81 ± 1.13	61.18 ± 2.06	38.50 ± 0.87	48.61 ± 1.45	57.28 ± 0.93

True Label \ Predicted Label	F1	F2	F3	F4	F5	F6	F7	Recall	Test Number
F1	1026	0	0	8	0	3	0	**98.94%**	1037
F2	2	1028	0	5	0	1	1	**99.13%**	1037
F3	1	0	1033	0	0	3	0	**99.61%**	1037
F4	7	26	1	979	0	24	0	**94.41%**	1037
F5	0	0	0	23	1014	0	0	**97.78%**	1037
F6	0	0	7	9	0	1021	0	**98.46%**	1037
F7	0	0	0	1	0	0	1036	**99.90%**	1037
Precision	**99.03%**	**97.53%**	**99.23%**	**95.51%**	**100%**	**97.05%**	**99.90%**	—	7259

Fig. 10. The Confusion matrix of MWA-CNN on HSA bearing dataset (SNR = 4 dB).

True Label \ Predicted Label	F1	F2	F3	F4	F5	F6	F7	Recall	Test Number
F1	926	26	30	28	1	15	11	**89.30%**	1037
F2	21	921	10	40	12	10	23	**88.81%**	1037
F3	1	0	1000	2	0	14	20	**96.43%**	1037
F4	47	42	5	840	3	65	35	**81.00%**	1037
F5	4	29	0	26	975	1	2	**94.02%**	1037
F6	37	20	103	99	3	737	38	**71.07%**	1037
F7	8	12	6	11	7	25	968	**93.35%**	1037
Precision	**88.70%**	**88.71%**	**86.66%**	**80.31%**	**97.40%**	**85.01%**	**88.24%**	—	7259

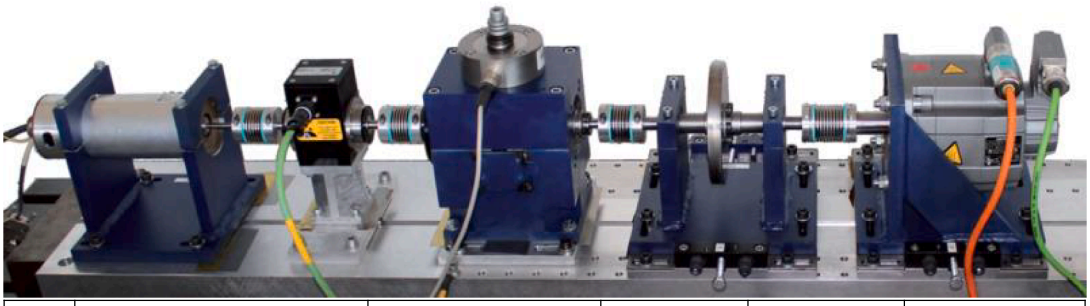
Fig. 11. The Confusion matrix of MWA-CNN on HSA bearing dataset (SNR = − 4 dB).

and F3. This leads to weaker fault features, which is easily overwhelmed by noise. In addition, category F6 is also easily affected by noise, which causes the model to incorrectly predict it into categories F3, F4, F7, etc.

G. Case 2. *H. **Motor Bearing Fault Diagnosis.***

4.5. Data Description

This dataset comes from the motor bearing signal acquisition experiment platform [35]. The test rig used is shown in Fig. 12. In addition, Fig. 12 also shows the fault information and working status information of the experimental bearing. The test rig consists of several modules: an electric motor, a torque-measurement shaft, a rolling bearing test module, a flywheel and a load motor.



Label	Damage	Bearing Element	Radial Force	Load Torque	Rotational Speed
H1	Normal	–	400N,1000N	0.1Nm,0.7Nm	900rpm,1500rpm
H2	Fatigue: Pitting	Outer Ring	400N,1000N	0.1Nm,0.7Nm	900rpm,1500rpm
H3	Plastic Deform.: Indentations	Outer Ring	400N,1000N	0.1Nm,0.7Nm	900rpm,1500rpm
H4	Fatigue: Pitting	Outer Ring; Inner Ring	400N,1000N	0.1Nm,0.7Nm	900rpm,1500rpm
H5	Plastic Deform.: Indentations	Outer Ring; Inner Ring	400N,1000N	0.1Nm,0.7Nm	900rpm,1500rpm
H6	Fatigue: Pitting	Inner Ring	400N,1000N	0.1Nm,0.7Nm	900rpm,1500rpm

Fig. 12. The motor bearing test rig and the fault information of experimental bearings.

Experimental bearings with different failure modes are installed in the bearing test module to obtain experimental data. The failures of the experimental bearings are real and not artificial. The fault location is mainly divided into the inner ring and outer ring, and the failure mode is mainly fatigue pitting and plastic deform. As shown in Fig. 12, the dataset is divided into five different fault categories, plus the normal category, the dataset has a total of six categories. This experiment only uses vibration signal data, and its sampling frequency is 64 kHz. The sliding segmentation method is also used to increase the signal samples, and the length of each sample is 5120 $\times$ 1. A total of 36,053 training samples and 12,017 test samples were obtained.

4.6. *Performance comparison*

This section conducts experiments on MWA-CNN and five existing methods on the motor bearing dataset to show the performance of these methods on different datasets. The parameter configuration of these methods has been introduced in Section IV.B. Similarly, we also conducted experiments on these methods under 4 dB, 0 dB, −2dB and − 4 dB noise. The results are shown in Table 5.

Similarly, MWA-CNN performs better than five comparison methods under four kinds of noise conditions. When SNR = 4 dB, MWA-CNN achieved a diagnostic accuracy of 99.70%. In particular, when SNR = − 4 dB, MWA-CNN still achieved 98.71% accuracy. When the noise changes from 4 dB to − 4 dB, the accuracy of MWA-CNN only drops by less than 1%. This shows that MWA-CNN has very good noise robustness on this dataset, and it can obtain fault features from noisy signals very well. It can be found that the motor bearing dataset is relatively simple, and the performance of all methods is higher than that on the HSA bearing dataset. When SNR = − 4 dB, the accuracy of these five comparison methods are all over 85%. However, the performance of these methods has dropped significantly as the noise increases. In addition, different methods have different performance on different datasets. For example, VGG-16 does not perform well on the HSA bearing dataset, but obtains good results on the motor bearing dataset. MWA-CNN has obtained excellent results on these two datasets, which shows that MWA-CNN has good adaptability.

Fig. 13 and Fig. 14 show the confusion matrix of MWA-CNN when SNR = 4 dB and SNR = − 4 dB. On the motor bearing dataset, MWA-CNN can achieve good performance under these two noise conditions. In particular, when SNR = 4 dB, the precision of categories H1, H2 and H4 are all 100%. When the SNR changed from 4 dB to − 4 dB, only the Recall of category H6 showed a significant drop, and its Recall changed from 99.70% to 91.55%. The performance of other categories has not changed much. This shows that MWA-CNN has a very good performance on this dataset, and its performance does not drop significantly in a strong noise environment.

Table 5
The Experimental Results of MWA-CNN and Five Comparison Methods Under Four Kinds of Noise.

Noise	MWA-CNN	MA1DCNN	WenCNN	ResNet-18	VGG-16	LSTM
4 dB	**99.70 ± 0.03**	99.33 ± 0.38	95.07 ± 0.63	99.29 ± 0.21	99.65 ± 0.13	97.93 ± 0.25
0 dB	**99.65 ± 0.11**	98.17 ± 0.27	90.79 ± 1.10	97.91 ± 0.35	99.47 ± 0.20	95.98 ± 0.70
−2 dB	**98.99 ± 0.14**	89.16 ± 0.79	88.57 ± 0.38	95.58 ± 0.89	98.16 ± 0.47	94.22 ± 0.55
−4 dB	**98.71 ± 0.21**	86.58 ± 2.44	85.47 ± 0.94	91.62 ± 0.99	96.50 ± 0.24	92.97 ± 0.88

True Label \ Predicted Label	H1	H2	H3	H4	H5	H6	Recall	Test Number
H1	1994	0	0	0	7	0	**99.65%**	2001
H2	0	2979	0	0	4	29	**98.90%**	3012
H3	0	0	1999	0	0	0	**100%**	1999
H4	0	0	2	2004	0	0	**99.90%**	2006
H5	0	0	0	0	999	0	**100%**	999
H6	0	0	0	0	6	1994	**99.70%**	2000
Precision	**100%**	**100%**	**99.90%**	**100%**	**98.33%**	**98.57%**	—	12017

Fig. 13. The Confusion matrix of MWA-CNN on motor bearing dataset (SNR = 4 dB).

True Label \ Predicted Label	H1	H2	H3	H4	H5	H6	Recall	Test Number
H1	1994	3	0	0	4	0	**99.65%**	2001
H2	0	2988	0	2	1	21	**99.20%**	3012
H3	0	0	1999	0	0	0	**100%**	1999
H4	0	0	0	2006	0	0	**100%**	2006
H5	7	0	0	0	992	0	**99.30%**	999
H6	32	100	0	0	37	1831	**91.55%**	2000
Precision	**98.08%**	**96.67%**	**100%**	**99.90%**	**95.94%**	**98.97%**	—	12017

Fig. 14. The Confusion matrix of MWA-CNN on motor bearing dataset (SNR = − 4 dB).

4.7. Computational burden analysis

This section analyzes the computational burden of the proposed MWA-CNN and six other network models. Table 6 shows the parameters and Multiply-Accumulate Operations (MACs) of these methods. 1DCNN has the same network architecture as MWA-CNN but does not have the DWA-Layer. In other words, 1DCNN does not include the wavelet and attention mechanism. It can be seen that adding wavelet transform and frequency attention mechanism to the CNN model greatly increases the computational burden of the model. For example, the MACs of 1DCNN are only 0.02 G, and the MACs of MWA-CNN are 0.17 G. Although the computational burden has increased significantly, the benefits of adding DWA-Layers are also apparent. For example, on the HSA dataset with 0 dB noise, the performance of MWA-CNN is nearly 18% higher than that of 1DCNN. Moreover, the MACs of ResNet-18 and VGG-16 reach 0.44 G and 0.62 G, respectively. Compared with ResNet-18 and VGG-16, the computational burden of MWA-CNN is acceptable. The diagnostic performance of MWA-CNN is also significantly better than ResNet-18 and VGG-16. These results show that the proposed DWA-Layer can significantly improve the model's diagnostic performance, and the computational burden increase is acceptable.

Table 6
The Parameters and Computational Burden of MWA-CNN and the Six Comparison Methods.

-	MWA-CNN	1DCNN	MA1DCNN	WenCNN	ResNet-18	VGG-16	LSTM
Parameters	1.39 M	3.01 M	0.29 M	0.90 M	3.85 M	50.97 M	1.38 M
MACs	0.17 G	0.02 G	0.07 G	0.08 G	0.44 G	0.62 G	0.11 G

5. Discussions

5.1. Discuss the number of DWA-Layer

To demonstrate the effect of the number of DWA-Layer on model performance, we conduct experiments on the HSA bearing dataset with 4 dB noise. Seven different network architectures are experimented, namely MWA-CNN-2, MWA-CNN-3,…, MWA-CNN-8, where 2,3,…,8 represent the number of DWA-Layer. The experimental results are shown in Fig. 15. As the number of DWA-Layer increases, the performance of the diagnostic model also increases gradually. For example, the performance of MWA-CNN-2 is below 80%, while the performance of MWA-CNN-5 exceeds 95%. This shows that with the increase of DWA-Layer, the signal is decomposed more finely, and more valuable features are learned. When the number of DWA-Layer exceeds six, the performance of the diagnostic model does not increase but slightly decreases, which indicates that the model is over-fitting and no additional DWA-Layer is needed. Therefore, in this study, the number of DWA-Layer is set to six. In addition, for more complex and challenging tasks, the number of DWA-Layer can be increased to obtain better learning ability and diagnostic performance.

5.2. Comparing interpretable Space-Based approaches

The proposed method automatically obtains valuable features from the signal through a deep fusion architecture of wavelet transform and CNN. In addition, CNN can also learn valuable features from other interpretable feature spaces to complete fault prediction. For example, we can perform empirical mode decomposition (EMD) on the signal to get multiple intrinsic mode functions (IMFs) or perform fast Fourier transform (FFT) on the signal to get the frequency domain distribution. In addition, Wavelet-Kernel-Net combines continuous wavelet transform with the first convolutional layer to obtain valuable time–frequency information. To this end, this experiment conducts experimental analysis on MWA-CNN, EMD-CNN, FFT-CNN, Wavelet-Kernel-Net (Laplace-ResNet) [19], and 1DCNN, respectively. 1DCNN is the baseline model of the above network, mainly composed of six 1D convolutional layers. For EMD-CNN, the signal is first decomposed by EMD, and then 1DCNN is used to learn from multiple IMFs to complete fault prediction. For FFT-CNN, the signal is first decomposed by FFT, and then 1DCNN is used to learn from time and frequency domains to complete fault prediction. The backbone network of Laplace-ResNet is the standard ResNet-18 and uses the Laplace wavelet. The experiment used the motor bearing dataset and the results are shown in Table 7.

The results show that adopting an interpretable feature space can improve the performance of diagnostic models. This confirms that the feature space constructed by EMD, FFT, or wavelet is beneficial for CNN to learn valuable fault-related features. For example, compared with 1DCNN, the diagnostic performance of EMD-CNN and FFT-CNN is improved by 5.86% and 5.01% under −4 dB noise, respectively. The diagnostic performance of Laplace-ResNet is comparable to that of EMD-CNN. Through the deep integration of wavelet and CNN, the proposed MWA-CNN achieves the best performance, showing that it fully exploits the advantages of CNN and wavelet.

5.3. The influence of wavelet basis functions

In the experiment of Section IV, we only used the DB wavelet. In order to verify that the proposed method can also be applied to other wavelet basis functions, we further discussed the performance of other six wavelet basis functions on the HSA bearing dataset. The six wavelet basis functions are Haar, Dmeyer (Dmey), Symlet (sym8), Biorthogonal (bior3.1), Reverse Biorthogonal (rbio3.1) and Coiflet (coif8). The network architecture adopts MWA-CNN, and their experimental results under four kinds of noise conditions are shown in Fig. 16.

Obviously, using different wavelet basis functions, MWA-CNN shows different fault diagnosis performance. We found that db16 wavelet, sym8 wavelet, rbio3.1 wavelet and coif8 wavelet have relatively better performance on the HSA bearing dataset. For example, when SNR $= -4$ dB, the four wavelets mentioned above can achieve more than 85% diagnostic accuracy, while the accuracy of the other three wavelets is less than 85%. Although the diagnostic performance of the model changes when different wavelets are used, the magnitude of this change is not large. In general, these methods show good fault diagnosis ability. For example, when SNR $= 4$ dB, these methods have achieved a diagnostic accuracy of more than 97%. As the noise intensity increases, the performance gap among these wavelets gradually becomes obvious. When SNR $= -4$db, the worst-performing haar wavelet and bior3.1 wavelet still get more than 80% accuracy. Through the above experimental analysis, we suggest to adopt those wavelet basis functions with excellent performance, such as db16 wavelet, coif8 wavelet and rbio3.1 wavelet. Different wavelets have different characteristics. In future work, a multi-wavelet fusion strategy can be considered to make the model perform better.

5.4. The influence of the DB vanishing moments

In this section, we take dbN wavelet as an example to discuss the effect of the value of vanishing moments on network performance. We set a total of 18 network models, they use MWA-CNN and dbN wavelet, N is set to 2, 4, 6, 8, 10,…, 34, 36. The dataset uses the HSA bearing dataset. The experimental results under four different noise conditions are shown in Fig. 17.

Obviously, if different N is set for db wavelet, the model shows different diagnostic performance. When the noise is weak, these methods have good performance. For example, when SNR $= 4$ dB, the diagnostic accuracy of all models are higher than 97%. However, in the case of strong noise, and when N is less than 8, the model shows poor performance. For example, when SNR $= -4$ dB, the diagnostic accuracy of db2 is only 82%, which is significantly lower than the 87.3% accuracy of db10. When N is greater than 8, the

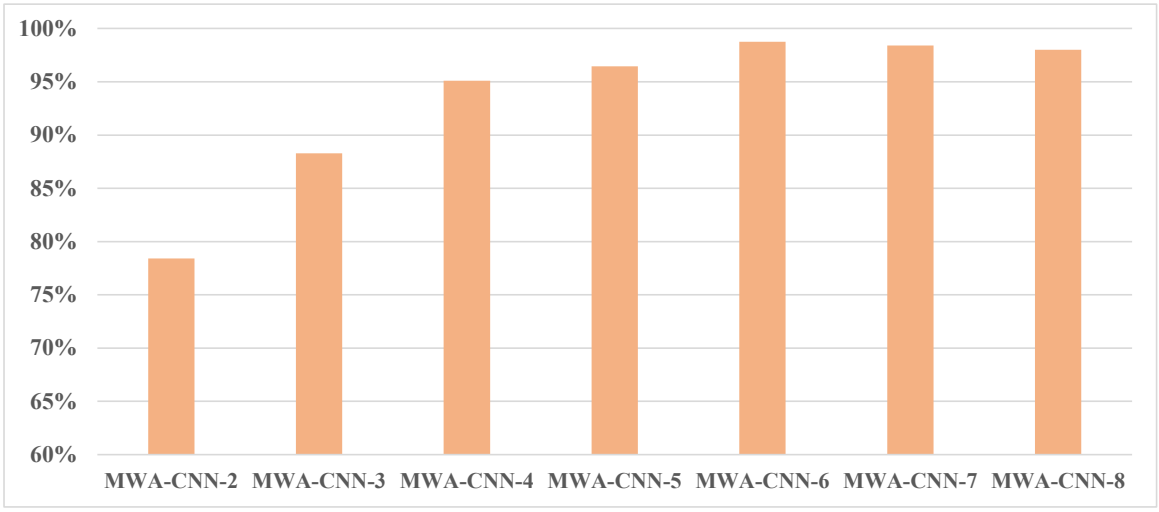


Fig. 15. Experimental results of the effect of the number of DWA-Layer on model performance.

Table 7
The Results of MWA-CNN and Interpretable Space-Based Methods Under Four Kinds of Noise.

Noise	MWA-CNN	Wavelet-Kernel-Net (Laplace-ResNet)	EMD-CNN	FFT-CNN	1DCNN
4 dB	**99.70 ± 0.03**	99.46 ± 0.21	99.58 ± 0.13	99.57 ± 0.15	98.10 ± 0.054
0 dB	**99.65 ± 0.11**	98.89 ± 0.34	99.05 ± 0.23	98.83 ± 0.29	96.53 ± 0.68
−2 dB	**98.99 ± 0.14**	97.04 ± 0.37	97.97 ± 0.52	96.66 ± 0.53	92.49 ± 0.65
−4 dB	**98.71 ± 0.21**	94.73 ± 0.63	95.15 ± 0.84	94.30 ± 0.59	89.29 ± 1.68

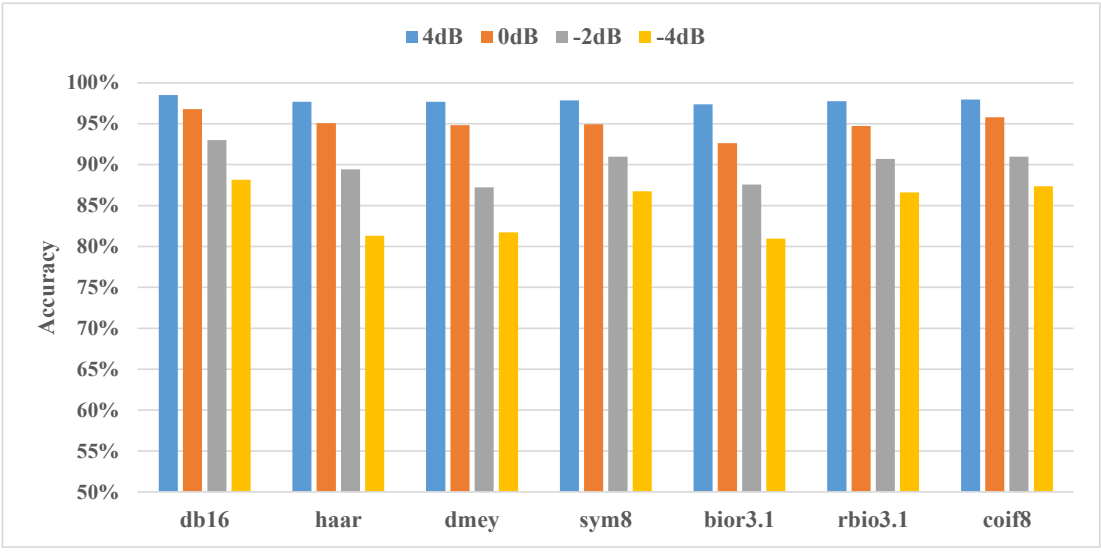


Fig. 16. The Experimental results of MWA-CNN with different wavelet basis functions (SNR = 4 dB, 0 dB, −2dB and − 4 dB).

diagnostic performance of most methods is between 86% and 88%. In general, when the value of N is around 16, the model can obtain relatively good performance.

6. Interpretability analysis

This section explores the feature learning mechanism of the proposed method. Specifically, we first analyze the model's feature

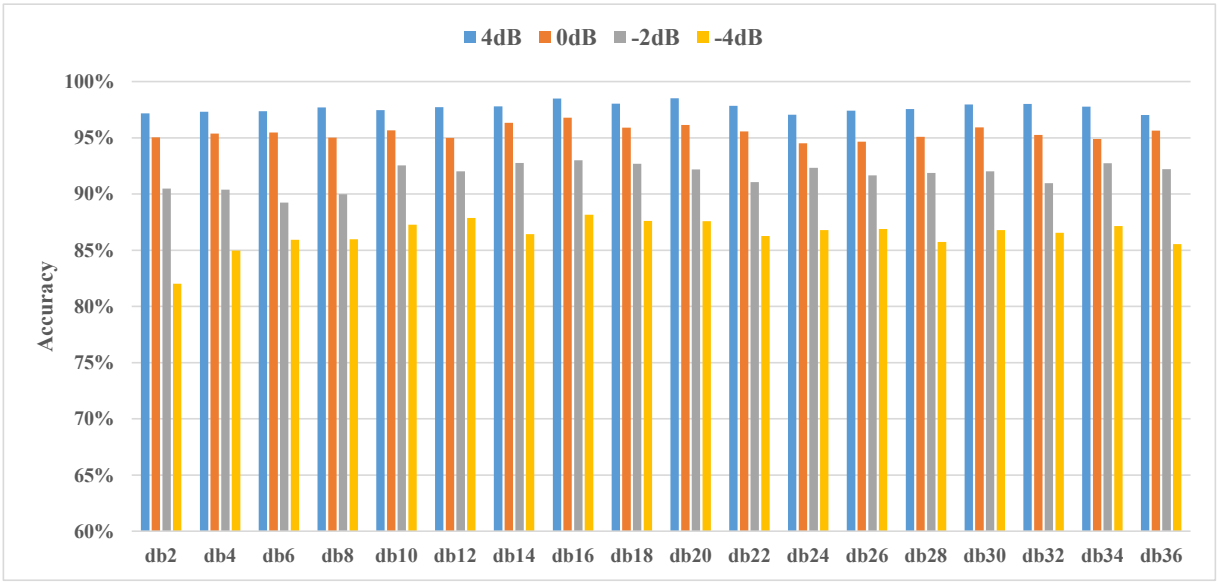


Fig. 17. The Experimental results of MWA-CNN with different db*N* (SNR = 4 dB, 0 dB, −2dB and − 4 dB).

learning preference by visualizing attention weight vectors. Subsequently, the time-domain diagrams, frequency-domain diagrams, and squared envelope spectra of the learned features are shown to explore the learning process of the proposed model.

6.1. *Frequency attention weights visualization*

Fig. 18 shows the MWA-CNN attention weights for two signal samples of category F3 and F4, the experiment uses the HSA bearing dataset and SNR = 4 dB. In MWA-CNN, a total of five FAMs are used, and the weight vectors ($z'_1, z'_2, \ldots, z'_5$) of these FAMs are shown in Fig. 18. Each small grid represents a weight value. The weight value is represented by a color, and the brighter the color, the greater the value. The weight value represents the importance of the corresponding frequency feature. The left side represents the attention weights of low-frequency feature components, and the right side represents the attention weights of high-frequency feature components.

Obviously, the attention module gives different weights to different feature signals, which shows that the attention module is trying to distinguish which information is important and which information is not. According to the bearing fault mechanism, when a bearing has a local fault, the faulty part and other components produce a periodic short-term impact and encourage the bearing system to perform high-frequency free attenuation vibration according to its resonance frequency. In addition, different bearing fault categories will produce different low-frequency fault characteristic frequencies, which are the most critical indicators for distinguishing bearing

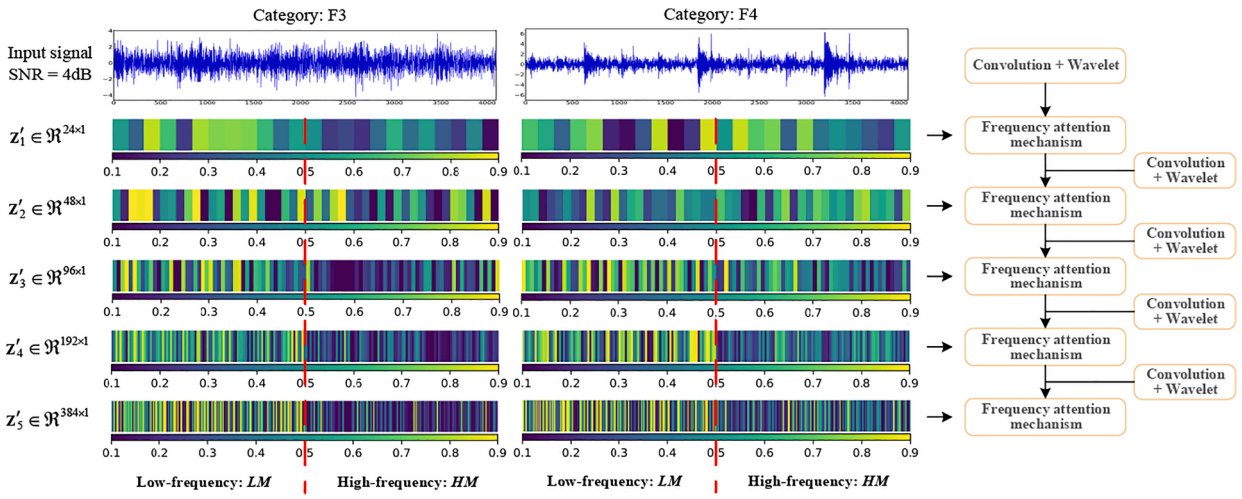


Fig. 18. The attention weights of MWA-CNN for two signal samples of two different classes.

faults. Based on this, we found that CNNs tend to learn fault-related low-frequency features and, when necessary, learn a small number of high-frequency features to achieve accurate fault identification. Generally, the shallow layer of the model is mainly responsible for filtering irrelevant information and retaining important information, and the deep layer of the model is responsible for modeling high-level abstract features. This shows that the network gradually filters and learns the high and low frequencies of the signal in the shallow layers, and then models the fault-related low-frequency features in the deep layers.

6.2. *Visualization of learned features*

To further explore the feature learning mechanism of MWA-CNN, we perform the FFT on the features learned in MWA-CNN and obtain their square envelope spectra. Fig. 19 shows the time-domain and frequency-domain diagrams of the learned features, and their squared envelope spectra. ①, ②, ③, and ④ represent the feature signals obtained from the corresponding positions of the network, respectively. ① indicates the input signal sample.② represents the feature signal output by the first layer convolution module. ③ represents the feature signal output after wavelet transform. ④ is obtained by the inverse wavelet transform of the feature signals output of the first DWA-Layer. The Case Western Reserve University bearing dataset is used to facilitate the visualization of fault characteristic frequencies on the squared envelope spectrum. Due to the ReLU function, the time domain diagram only contains positive values.

Features ② and ④ represent the input and output features of the first DWA-Layer, respectively. As shown in Fig. 19, from the time-domain and frequency-domain diagrams, there is little difference between feature ② and feature ④. The difference between Features ② and ④ is evident in the squared envelope spectrum. The fault characteristic frequency in the square envelope spectrum of feature ④ is clearly visible, but the fault characteristic frequency in the square envelope spectrum of feature ② is not clearly displayed. This shows that through wavelet decomposition and frequency attention learning, important fault-related features are effectively learned and highlighted. This again confirms the usefulness of the proposed method.

To explore the learning process inside DWA-Layer, we have visualized feature ③. Fig. 20 shows the squared envelope spectra of the four feature signals and their corresponding attention weights. These four feature signals are all selected from the low-frequency part. As shown in Fig. 20, the feature signal with a large attention weight generally has a relatively obvious fault characteristic frequency. For example, for feature signals with attention weights of 0.8082, 0.5202, and 0.5965, their fault characteristic frequencies are clearly visible. For the feature signal with a weight of 0.2783, its fault characteristic frequency is not clearly displayed. This shows that DWA-Layer enhances valuable information and suppresses useless information through frequency decomposition and attention mechanisms. In the visualization experiment, the attention mechanism will occasionally give greater weight to the feature signal with an unclear fault characteristic frequency, which may be that the signal contains unknown but essential features. This needs to be further explored in future work.

7. Conclusions

This study aims to deeply integrate the DWT and CNN models to maximize the advantages of these two technologies. To this end, a

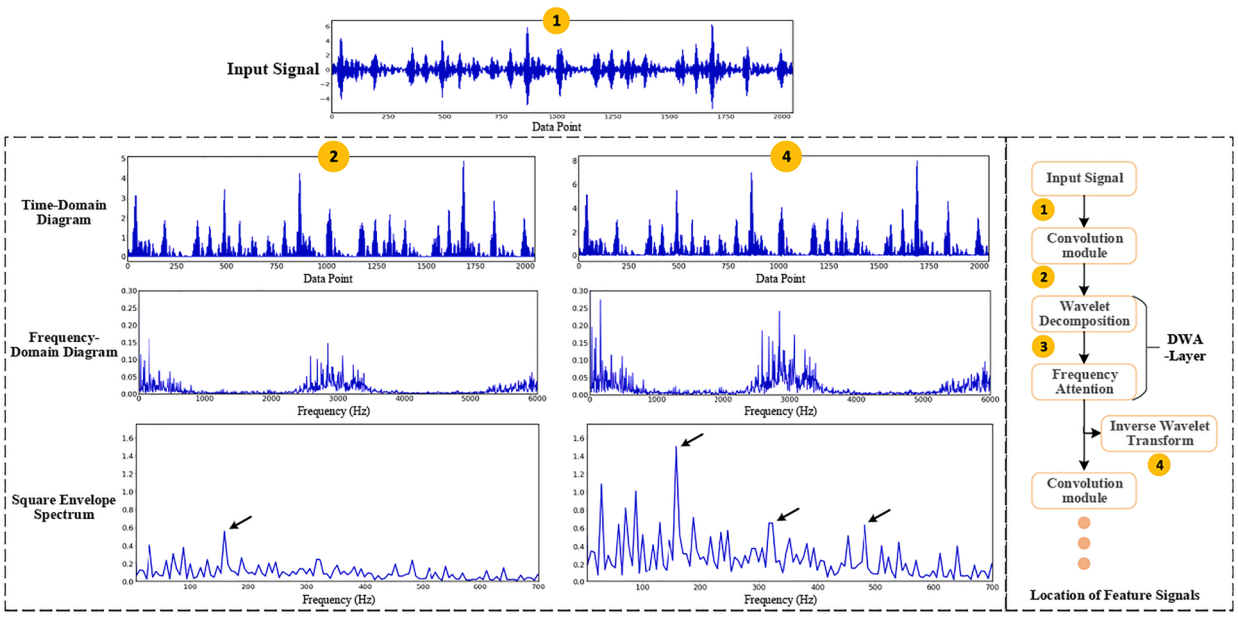


Fig. 19. The time-domain and frequency-domain diagrams of the learned features, and their squared envelope spectra (arrows highlight the fault characteristic frequency).

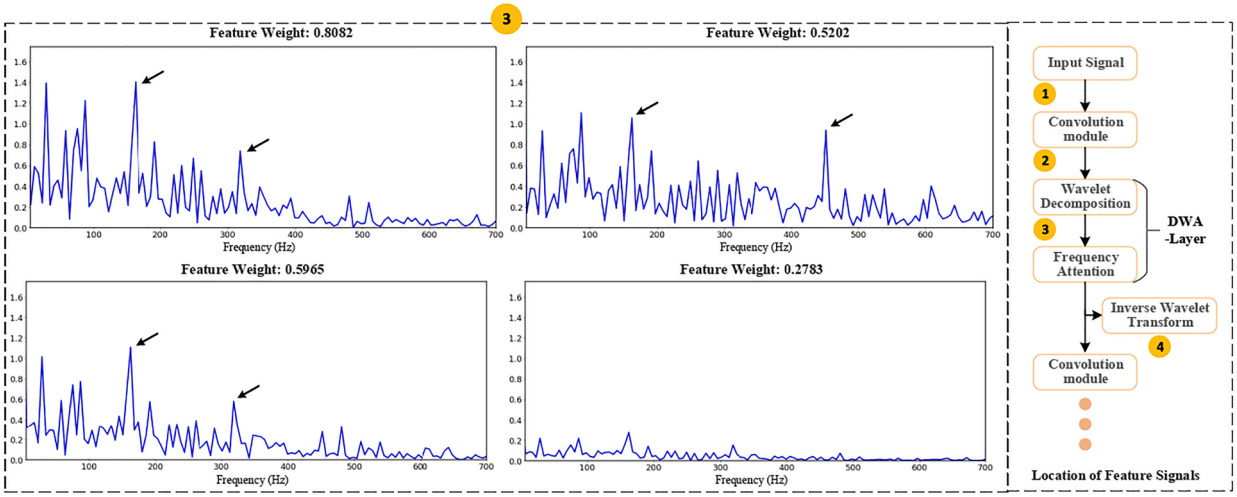


Fig. 20. The squared envelope spectra of the four feature signals and their corresponding attention weights (arrows highlight the fault characteristic frequency).

novel multi-layer wavelet attention CNN for machinery fault diagnosis is proposed. MWA-CNN is mainly composed of multiple DWA-Layers and multiple convolutional layers. DWA-Layer and convolutional layers are stacked alternately. In DWA-Layer, the DWT algorithm is used to decompose the signal into multiple frequency components, and the attention mechanism is used to filter out useful frequency information. The convolutional layer is mainly used to automatically learn useful information from the obtained frequency components. Experimental results show that DWA-Layer can significantly improve the fault diagnosis performance and noise robustness of the CNN model. MWA-CNN has better performance than other deep learning methods. Especially in a strong noise environment, MWA-CNN has excellent performance. Based on DWT, we have analyzed the feature learning mechanism of CNN from the perspective of frequency domain with the help of the attention mechanism, which also provides a direction for the interpretability analysis of the CNN model. In future work, multi-wavelet fusion strategies can be explored and studied to improve the performance of the CNN model further.

Declaration of Competing Interest

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

Data availability

We have open-sourced our code.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1702400, and in part by Sichuan Province Key Research and Development Program under Grant 23ZDYF0212.

References

[1] M. Cerrada, R. Sánchez, C. Li, F. Pacheco, D. Cabrera, J. Valente de Oliveira, et al., A review on data-driven fault severity assessment in rolling bearings, Mech. Syst. Sig. Process. 99 (2018) 169–196.
[2] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A.K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mech. Syst. Sig. Process. 138 (2020), 106587.
[3] W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load, Mech. Syst. Sig. Process. 100 (2018) 439–453.
[4] D. Peng, H. Wang, Z. Liu, W. Zhang, M.J. Zuo, J. Chen, Multibranch and Multiscale CNN for Fault Diagnosis of Wheelset Bearings Under Strong Noise and Variable Load Condition, IEEE Trans. Ind. Inf. 16 (7) (2020) 4949–4960.
[5] H. Liu, J. Zhou, Y. Zheng, W. Jiang, Y. Zhang, Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders, ISA Trans. 77 (2018) 167–178.
[6] J. Shi, D. Peng, Z. Peng, Z. Zhang, K. Goebel, D. Wu, Planetary gearbox fault diagnosis using bidirectional-convolutional LSTM networks, Mech. Syst. Sig. Process. 162 (2022), 107996.
[7] T. de Bruin, K. Verbert, R. Babuška, Railway Track Circuit Fault Diagnosis Using Recurrent Neural Networks, IEEE Trans. Neural Networks Learn. Syst. 28 (3) (2017) 523–533.
[8] F. Zhou, S. Yang, H. Fujita, D. Chen, C. Wen, Deep learning fault diagnosis method based on global optimization GAN for unbalanced data, Knowl.-Based Syst. 187 (2020), 104837.

[9] H. Wang, Z. Liu, D. Peng, M. Yang, Y. Qin, Feature-Level Attention-Guided Multitask CNN for Fault Diagnosis and Working Conditions Identification of Rolling Bearing, IEEE Trans. Neural Networks Learn. Syst. 33 (9) (2021) 4757–4769.
[10] B. Zhao, X. Zhang, H. Li, Z. Yang, Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions, Knowl.-Based Syst. 199 (2020), 105971.
[11] X. Wang, D. Mao, X. Li, Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network, Measurement 173 (2021), 108518.
[12] T. Jin, C. Yan, C. Chen, Z. Yang, H. Tian, S. Wang, Light neural network with fewer parameters based on CNN for fault diagnosis of rotating machinery, Measurement 181 (2021), 109639.
[13] Z. Liu, H. Wang, J. Liu, Y. Qin, D. Peng, Multitask Learning Based on Lightweight 1DCNN for Fault Diagnosis of Wheelset Bearings, IEEE Trans. Instrum. Meas. 70 (2021) 1–11.
[14] Z. Chen, K. Gryllias, W. Li, Mechanical fault diagnosis using Convolutional Neural Networks and Extreme Learning Machine, Mech. Syst. Sig. Process. 133 (2019), 106272.
[15] H. Han, H. Wang, Z. Liu, J. Wang, Intelligent vibration signal denoising method based on non-local fully convolutional neural network for rolling bearings, ISA Trans. 122 (2022) 13–23.
[16] H. Wang, Z. Liu, D. Peng, Y. Qin, Understanding and learning discriminant features based on multiattention 1DCNN for wheelset bearing fault diagnosis, IEEE Trans. Ind. Inf. 16 (9) (2020) 5735–5745.
[17] Z. Yang, J. Zhang, Z. Zhao, Z. Zhai, X. Chen, Interpreting network knowledge with attention mechanism for bearing fault diagnosis, Appl. Soft Comput. 97 (2020), 106829.
[18] D. Zhou, Q. Yao, H. Wu, S. Ma, H. Zhang, Fault diagnosis of gas turbine based on partly interpretable convolutional neural networks, Energy 200 (2020), 117467.
[19] T. Li, Z. Zhao, C. Sun, L. Cheng, X. Chen, R. Yan, et al., WaveletKernelNet: an interpretable deep neural network for industrial intelligent diagnosis, IEEE Transactions on Systems, Man, and Cybernetics: Systems 52 (4) (2022) 2302–2312.
[20] J. Chen, Z. Li, J. Pan, G. Chen, Y. Zi, J. Yuan, et al., Wavelet transform based on inner product in fault diagnosis of rotating machinery: A review, Mech. Syst. Sig. Process. 70–71 (2016) 1–35.
[21] M.M.M. Islam, J. Kim, Automated Bearing Fault Diagnosis Scheme Using 2D Representation of Wavelet Packet Transform and Deep Convolutional Neural Network, Comput. Ind. 106 (2019) 142–153.
[22] R. Chen, X. Huang, L. Yang, X. Xu, X. Zhang, Y. Zhang, Intelligent fault diagnosis method of planetary gearboxes based on convolution neural network and discrete wavelet transform, Comput. Ind. 106 (2019) 48–59.
[23] P. Liang, C. Deng, J. Wu, Z. Yang, J. Zhu, Z. Zhang, Compound fault diagnosis of gearboxes via multi-label convolutional neural network and wavelet transform, Comput. Ind. 113 (2019), 103132.
[24] Y. Zhang, K. Xing, R. Bai, D. Sun, Z. Meng, An enhanced convolutional neural network for bearing fault diagnosis based on time–frequency image, Measurement 157 (2020), 107667.
[25] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 674–693.
[26] M. Lin, Q. Chen and S. Yan, "Network In Network," [Online]. Available: https://arxiv.org/abs/1312.4400.
[27] Y. Wu and K. He, "Group Normalization," in Proc. ECCV, 2018, pp. 3-19.
[28] M. Reza Asadi Asad Abad, H. Ahmadi, A. Moosavian, M. Khazaee, M. Ranjbar Kohan and M. Mohammadi, "Discrete wavelet transform and artificial neural network for gearbox fault detection based on acoustic signals," Journal of Vibroengineering, vol. 15, no. 1, pp. 459-463, 2013.
[29] O.N. Oyelade, A.E. Ezugwu, A novel wavelet decomposition and transformation convolutional neural network with data augmentation for breast cancer detection using digital mammogram, Sci. Rep. 12 (1) (2022) 5913.
[30] F. Cotter, "Uses of Complex Wavelets in Deep Convolutional Neural Networks," University of Cambridge, 2019.
[31] A.P. Daga, A. Fasana, S. Marchesiello, L. Garibaldi, The Politecnico di Torino rolling bearing test rig: Description and analysis of open access data, Mech. Syst. Sig. Process. 120 (2019) 252–273.
[32] L. Wen, X. Li, L. Gao, Y. Zhang, A new convolutional neural network-based data-driven fault diagnosis method, IEEE Trans. Ind. Electron. 65 (7) (2018) 5990–5998.
[33] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. CVPR, 2016, pp. 770-778.
[34] Karen Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Proc. International Conference on Learning Representations, 2015.
[35] C. Lessmeier, J.K. Kimotho, D. Zimmer and W. Sextro, "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 Proc. European Conference of the Prognostics and Health Management Society, 2016, pp. 5-8.

Interpretable Convolutional Filters with SincNet

Mirco Ravanelli
Mila, Université de Montréal

Yoshua Bengio
Mila, Université de Montréal
CIFAR Fellow

Abstract

Deep learning is currently playing a crucial role toward higher levels of artificial intelligence. This paradigm allows neural networks to learn complex and abstract representations, that are progressively obtained by combining simpler ones. Nevertheless, the internal "black-box" representations automatically discovered by current neural architectures often suffer from a lack of interpretability, making of primary interest the study of explainable machine learning techniques.

This paper summarizes our recent efforts to develop a more interpretable neural model for directly processing speech from the raw waveform. In particular, we propose *SincNet*, a novel Convolutional Neural Network (CNN) that encourages the first layer to discover more meaningful filters by exploiting parametrized sinc functions. In contrast to standard CNNs, which learn all the elements of each filter, only low and high cutoff frequencies of band-pass filters are directly learned from data. This inductive bias offers a very compact way to derive a customized filter-bank front-end, that only depends on some parameters with a clear physical meaning. Our experiments, conducted on both speaker and speech recognition, show that the proposed architecture converges faster, performs better, and is more interpretable than standard CNNs.

1 Introduction

Deep learning has recently contributed to achieving unprecedented performance levels in numerous tasks, mainly thanks to the progressive maturation of supervised learning techniques [1]. The increased discrimination power of modern neural networks, however, is often obtained at the cost of a reduced interpretability of the model. Modern end-to-end systems, whose popularity is increasing in many fields such as speech recognition [2–4], often discover "black-box" internal representations that make sense for the machine but are arguably difficult to interpret by humans. The remarkable sensitivity of current neural networks toward adversarial examples [5], for instance, not only highlights how superficial the discovered representations could be but also raises crucial concerns about our capabilities to really interpret neural models. Such a lack of interpretability can be a major bottleneck for the development of future deep learning techniques. Having more meaningful insights on the logic behind network predictions and errors, in fact, can help us to better trust, understand, and diagnose our model, eventually guiding our efforts toward more robust deep learning. In recent years, a growing interest has been thus devoted to the development of interpretable machine learning [6, 7], as witnessed by the numerous works in the field, ranging from visualization [8, 9], diagnosis of DNNs [10], explanatory graphs [11], and explainable models [12], just to name a few.

Interpretability is a major concern for audio and speech applications as well [13]. CNNs and Recurrent Neural Networks (RNNs) are the most popular architectures nowadays used in speech and speaker recognition [2]. RNN can be employed to capture the temporal evolution of the speech signal [14–17], while CNNs, thanks to their weight sharing, local filters, and pooling networks are normally employed to extract robust and invariant representations [18]. Even though standard hand-crafted features such as FBANK and Mel-Frequency Cepstral Coefficients (MFCC) are still employed in many

32nd Conference on Neural Information Processing Systems (NIPS 2018) IRASL workshop, Montréal, Canada.

state-of-the-art systems [19–21], directly feeding a CNN with spectrogram bins [22–24] or even with raw audio samples [25–37] is an approach of increasing popularity. The engineered features, in fact, are originally designed from perceptual evidence and there are no guarantees that such representations are optimal for all speech-related tasks. Standard features, for instance, smooth the speech spectrum, possibly hindering the extraction of crucial narrow-band speaker characteristics such as pitch and formants. Conversely, directly processing the raw waveform allows the network to learn low-level representations that are possibly more customized on each specific task.

The downside of raw speech processing lies in the possible lack of interpretability of the filter bank learned in the first convolutional layer. According to us, the latter layer is arguably the most critical part of current waveform-based CNNs. This layer deals with high-dimensional inputs and is also more affected by vanishing gradient problems, especially when employing very deep architectures. As will be discussed in this paper, the filters learned by CNNs often take noisy and incongruous multi-band shapes, especially when few training samples are available. These filters certainly make some sense for the neural network, but they do not appeal to human intuition, nor appear to lead to an efficient representation of the speech signal.

To help the CNNs discover more meaningful filters, this work proposes to add some constraints on their shape. Compared to standard CNNs, where the filter-bank characteristics depend on several parameters (each element of the filter vector is directly learned), SincNet convolves the waveform with a set of parametrized sinc functions that implement band-pass filters [38]. The low and high cutoff frequencies are the only parameters of the filter learned from data. This solution still offers considerable flexibility but forces the network to focus on high-level tunable parameters that have a clear physical meaning. Our experimental validation has considered both speaker and speech recognition tasks. Speaker recognition is carried out on TIMIT [39] and Librispeech [40] datasets under challenging but realistic conditions, characterized by minimal training data (i.e., 12-15 seconds for each speaker) and short test sentences (lasting from 2 to 6 seconds). With the purpose of validating SincNet in both clean and noisy conditions, speech recognition experiments are conducted on both the TIMIT and DIRHA dataset [41, 42]. Results show that the proposed SincNet converges faster, achieves better performance, and is more interpretable than a more standard CNN.

The remainder of the paper is organized as follows. The SincNet architecture is described in Sec. 2. Sec. 3 discusses the relation to prior work. The experimental activity on both speaker and speech recognition is outlined in Sec. 4. Finally, Sec. 5 discusses our conclusions.

2 The SincNet Architecture

The first layer of a standard CNN performs a set of time-domain convolutions between the input waveform and some Finite Impulse Response (FIR) filters [43]. Each convolution is defined as follows[1]:

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l] \tag{1}$$

where $x[n]$ is a chunk of the speech signal, $h[n]$ is the filter of length L, and $y[n]$ is the filtered output. In standard CNNs, all the L elements (taps) of each filter are learned from data. Conversely, the proposed SincNet (depicted in Fig. 1) performs the convolution with a predefined function g that depends on few learnable parameters θ only, as highlighted in the following equation:

$$y[n] = x[n] * g[n, \theta] \tag{2}$$

A reasonable choice, inspired by standard filtering in digital signal processing, is to define g such that a filter-bank composed of rectangular bandpass filters is employed. In the frequency domain, the magnitude of a generic bandpass filter can be written as the difference between two low-pass filters:

$$G[f, f_1, f_2] = rect\Big(\frac{f}{2f_2}\Big) - rect\Big(\frac{f}{2f_1}\Big), \tag{3}$$

[1]Most deep learning toolkits actually compute *correlation* rather than *convolution*. The obtained flipped (mirrored) filters do not affect the results.

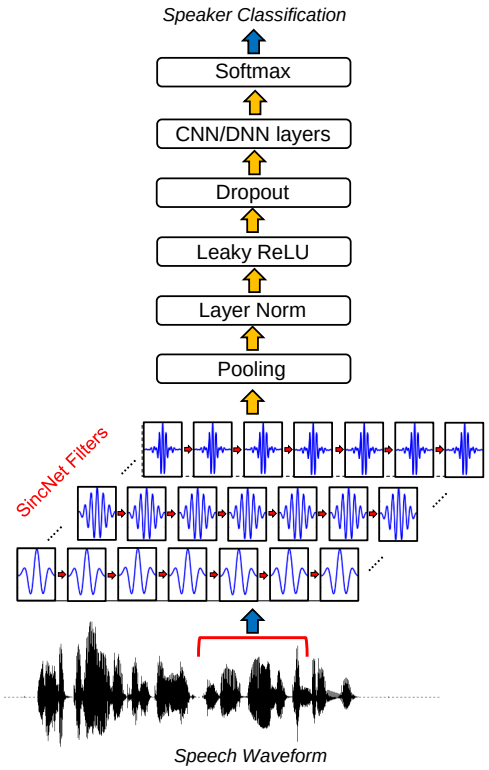


Figure 1: Architecture of SincNet.

where f_1 and f_2 are the learned low and high cutoff frequencies, and $rect(\cdot)$ is the rectangular function in the magnitude frequency domain[2]. After returning to the time domain (using the inverse Fourier transform [43]), the reference function g becomes:

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi f_2 n) - 2f_1 sinc(2\pi f_1 n), \tag{4}$$

where the sinc function is defined as $sinc(x) = sin(x)/x$.

The cut-off frequencies can be initialized randomly in the range $[0, f_s/2]$, where f_s represents the sampling frequency of the input signal. As an alternative, filters can be initialized with the cutoff frequencies of the mel-scale filter-bank, which has the advantage of directly allocating more filters in the lower part of the spectrum, where crucial speech information is located. To ensure $f_1 \geq 0$ and $f_2 \geq f_1$, the previous equation is actually fed by the following parameters:

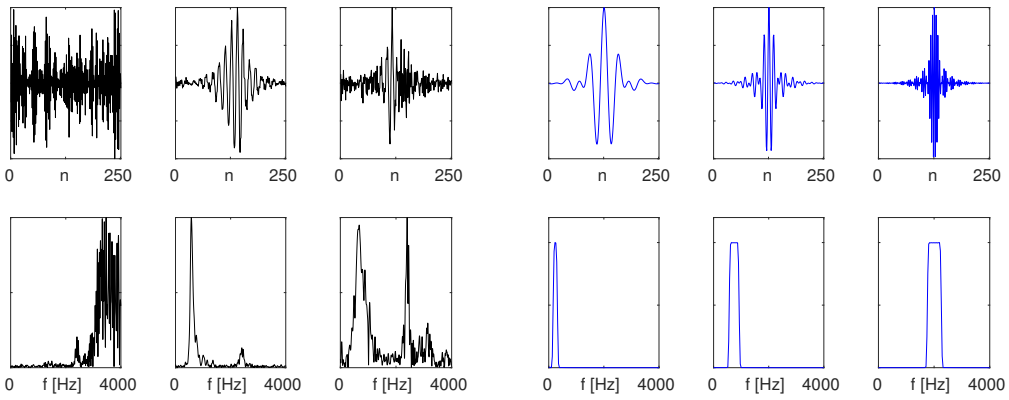
$$f_1^{abs} = |f_1| \tag{5}$$

$$f_2^{abs} = f_1 + |f_2 - f_1| \tag{6}$$

Note that no bounds have been imposed to force f_2 to be smaller than the Nyquist frequency, since we observed that this constraint is naturally fulfilled during training. Moreover, the gain of each filter is not learned at this level. This parameter is managed by the subsequent layers, which can easily attribute more or less importance to each filter output.

An ideal bandpass filter (i.e., a filter where the passband is perfectly flat and the attenuation in the stopband is infinite) requires an infinite number of elements L. Any truncation of g thus inevitably leads to an approximation of the ideal filter, characterized by ripples in the passband and limited attenuation in the stopband. A popular solution to mitigate this issue is windowing [43]. Windowing

[2]The phase of the $rect(\cdot)$ function is considered to be linear.



(a) CNN Filters

(b) SincNet Filters

Figure 2: Examples of filters learned by a standard CNN and by the proposed SincNet (using the Librispeech corpus on a speaker-id task). The first row reports the filters in the time domain, while the second one shows their magnitude frequency response.

is performed by multiplying the truncated function g with a window function w, which aims to smooth out the abrupt discontinuities at the ends of g:

$$g_w[n, f_1, f_2] = g[n, f_1, f_2] \cdot w[n]. \tag{7}$$

This paper uses the popular Hamming window [44], defined as follows:

$$w[n] = 0.54 - 0.46 \cdot cos\Big(\frac{2\pi n}{L}\Big). \tag{8}$$

The Hamming window is particularly suitable to achieve high frequency selectivity [44]. However, results not reported here reveal no significant performance difference when adopting other functions, such as Hann, Blackman, and Kaiser windows.

All operations involved in SincNet are fully differentiable and the cutoff frequencies of the filters can be jointly optimized with other CNN parameters using Stochastic Gradient Descent (SGD) or other gradient-based optimization routines. As shown in Fig. 1, a standard CNN pipeline (pooling, normalization, activations, dropout) can be employed after the first sinc-based convolution. Multiple standard convolutional, fully-connected or recurrent layers [15–17, 45] can then be stacked together to finally perform a classification with a softmax classifier.

Fig. 2 shows some examples of filters learned by a standard CNN and by the proposed SincNet for a speaker identification task trained on Librispeech (the frequency response is plotted between 0 and 4 kHz). As observed in the figures, the standard CNN does not always learn filters with a well-defined frequency response. In some cases, the frequency response looks noisy (see the first CNN filter), while in others assuming multi-band shapes (see the third CNN filter). SincNet, instead, is specifically designed to implement rectangular bandpass filters, leading to more a meaningful filter-bank.

2.1 Model properties

The proposed SincNet has some remarkable properties:

- **Fast Convergence:** SincNet forces the network to focus only on the filter parameters with a major impact on performance. The proposed approach actually implements a natural inductive bias, utilizing knowledge about the filter shape (similar to feature extraction methods generally deployed on this task) while retaining flexibility to adapt to data. This prior knowledge makes learning the filter characteristics much easier, helping SincNet to converge significantly faster to a better solution. Fig. 3 shows the learning curves of SincNet and CNN obtained in a speaker-id task. These results are achieved on the TIMIT dataset and highlight a faster decrease of the Frame Error Rate ($FER\%$) when SincNet is used.

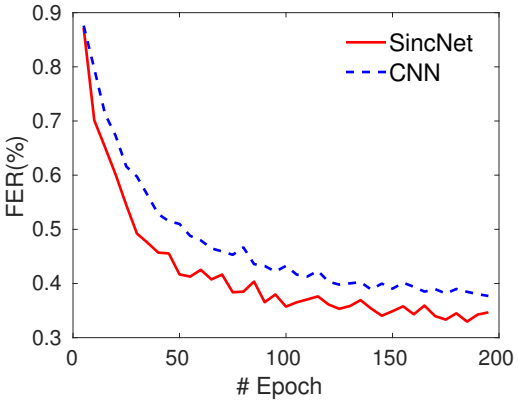


Figure 3: Frame Error Rate (%) obtained on speaker-id with the TIMIT corpus (using held-out data).

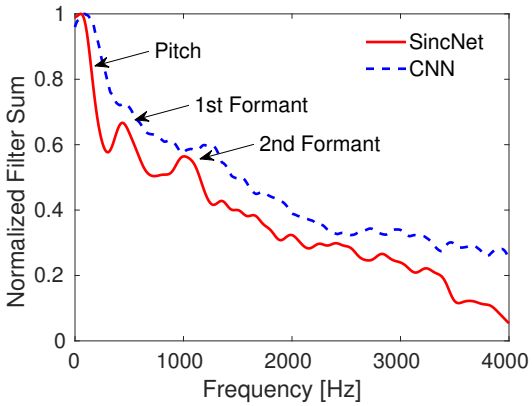


Figure 4: Cumulative frequency response of SincNet and CNN filters on speaker-id.

Moreover, SincNet converges to better performance leading to a FER of 33.0% against a FER of 37.7% achieved with the CNN baseline.

- **Few Parameters:** SincNet drastically reduces the number of parameters in the first convolutional layer. For instance, if we consider a layer composed of F filters of length L, a standard CNN employs $F \cdot L$ parameters, against the $2F$ considered by SincNet. If $F = 80$ and $L = 100$, we employ 8k parameters for the CNN and only 160 for SincNet. Moreover, if we double the filter length L, a standard CNN doubles its parameter count (e.g., we go from 8k to 16k), while SincNet has an unchanged parameter count (only two parameters are employed for each filter, regardless its length L). This offers the possibility to derive very selective filters with many taps, without actually adding parameters to the optimization problem. Moreover, the compactness of the SincNet architecture makes it suitable in the few sample regime.
- **Computational Efficiency**: The proposed function g is symmetric. This means we can perform convolution in a very efficient way by only considering one side of the filter and inheriting the results for the other half. This saves 50% of the first-layer computation over a standard CNN.
- **Interpretability**: The SincNet feature maps obtained in the first convolutional layer are definitely more interpretable and human-readable than other approaches. The filter bank, in fact, only depends on parameters with a clear physical meaning. Fig. 4, for instance, shows the cumulative frequency response of the filters learned by SincNet and CNN on a speaker-id task. The cumulative frequency response is obtained by summing up all the discovered filters and is useful to highlight which frequency bands are covered by the learned filters.

  Interestingly, there are three main peaks which clearly stand out from the SincNet plot (see the red line in the figure). The first one corresponds to the pitch region (the average pitch is 133 Hz for a male and 234 for a female). The second peak (approximately located at 500 Hz) mainly captures first formants, whose average value over the various English vowels is indeed 500 Hz. Finally, the third peak (ranging from 900 to 1400 Hz) captures some important second formants, such as the second formant of the vowel $/a/$, which is located on average at 1100 Hz. This filter-bank configuration indicates that SincNet has successfully adapted its characteristics to address speaker identification. Conversely, the standard CNN does not exhibit such a meaningful pattern: the CNN filters tend to correctly focus on the lower part of the spectrum, but peaks tuned on first and second formants do not clearly appear. As one can observe from Fig. 4, the CNN curve stands above the SincNet one. SincNet, in fact, learns filters that are, on average, more selective than CNN ones, possibly better capturing narrow-band speaker clues.

  Fig. 5 shows the cumulative frequency response of a CNN and SincNet obtained on a noisy speech recognition task. In this experiment, we have artificially corrupted TIMIT with a significant quantity of noise in the band between 2.0 and 2.5 kHz (see the spectrogram) and

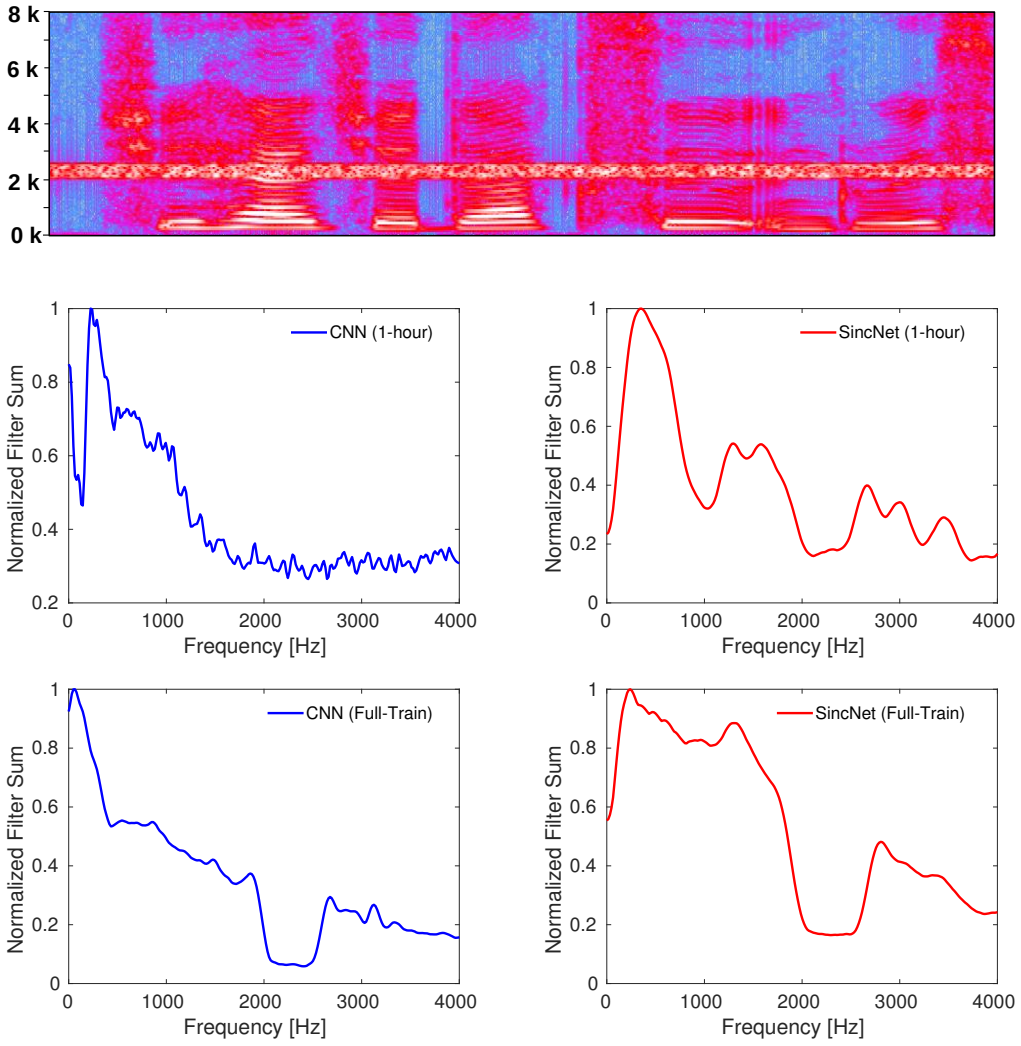


Figure 5: Cumulative frequency responses obtained on a speech recognition task trained with a noisy version of TIMIT. As shown in the spectrogram, noise has been artificially added into the band 2.0-2.5 kHz. Both the CNN and SincNet learn to avoid the noisy band, but SincNet learns it much faster, after processing only one hour of speech.

we have analyzed how fast the two architectures learn to avoid such a useless band. The second row of sub-figures compares the CNN and the SincNet at a very early training stage (i.e., after having processed only one hour of speech in the first epoch), while the last row shows the cumulative frequency responses after completing the training. From the figures emerges that both CNN and SincNet have correctly learned to avoid the corrupted band at end of training, as highlighted by the holes between 2.0 and 2.5 kHz in the cumulative frequency responses. SincNet, however, learns to avoid such a noisy band much earlier. In the second row of sub-figures, in fact, SincNet shows a visible valley in the cumulative spectrum even after processing only one hour of speech, while CNN has only learned to give more importance to the lower part of the spectrum.

3 Related Work

Several works have recently explored the use of low-level speech representations to process audio and speech with CNNs. Most prior attempts exploit magnitude spectrogram features [22–24, 46–48]. Although spectrograms retain more information than standard hand-crafted features, their design still

requires careful tuning of some crucial hyper-parameters, such as the duration, overlap, and typology of the frame window, as well as the number of frequency bins. For this reason, a more recent trend is to directly learn from raw waveforms, thus completely avoiding any feature extraction step. This approach has shown promise in speech [25–29], including emotion tasks [30], speaker recognition [35], spoofing detection [34], and speech synthesis [31, 32].

Similar to SincNet, some previous works have proposed to add constraints on the CNN filters, for instance forcing them to work on specific bands [46, 47]. Differently from the proposed approach, the latter works operate on spectrogram features and still learn all the L elements of the CNN filters. An idea related to the proposed method has been recently explored in [48], where a set of parameterized Gaussian filters are employed. This approach operates on the spectrogram domain, while SincNet directly considers the raw waveform in the time domain.

Some valuable works have recently proposed theoretical and experimental frameworks to analyze CNNs [49, 50]. In particular, [51, 35, 52] feed a standard CNN with raw audio samples and analyze the filters learned in the first layer on both speech recognition and speaker identification tasks. The authors highlight some interesting properties emerged from analyzing the cumulative frequency response and propose a spectral dictionary interpretation of the learned filters. Similarly to our findings, the latter works noticed that the filters tend to focus more on the lower part of the spectrum and they can sometimes highlight some peaks that likely corresponds to the fundamental frequency. In this work, we argue that all of these interesting properties can be observed more clearly and at an earlier training stage with SincNet.

This paper extends our previous studies on the SincNet [38]. To the best of our knowledge, this paper is the first that shows the effectiveness of the proposed SincNet in a speech recognition application. Moreover, this work not only considers standard close-talking speech recognition, but it also extends the validation of SincNet to distant-talking speech recognition [53–55].

4 Results

The proposed SincNet has been evaluated on both speech and speaker recognition using different corpora. This work considers a challenging but realistic speaker recognition scenario: for all the adopted corpora, we only employed 12-15 seconds of training material for each speaker, and we tested the system performance on short sentences lasting from 2 to 6 seconds. In the spirit of reproducible research, we release the code of SincNet for speaker identification[3] and speech recognition[4] (under the PyTorch-Kaldi project [56]). More details on the adopted datasets as well as on the SincNet and baseline setups can found in the **appendix**.

4.1 Speaker Recognition

Table 1 reports the Classification Error Rates (CER%) achieved on a speaker-id task. The table shows that SincNet outperforms other systems on both TIMIT (462 speakers) and Librispeech (2484 speakers) datasets. The gap with a standard CNN fed by raw waveform is larger on TIMIT, confirming the effectiveness of SincNet when few training data are available. Although this gap is reduced when LibriSpeech is used, we still observe a 4% relative improvement that is also obtained with faster convergence (1200 vs 1800 epochs). Standard FBANKs provide results comparable to SincNet only on TIMIT, but are significantly worse than our architecture when using Librispech. With few training data, the network cannot discover filters that are much better than that of FBANKs, but with more data a customized filter-bank is learned and exploited to improve the performance.

Table 2 extends our validation to speaker verification, reporting the Equal Error Rate (EER%) achieved with Librispeech. All DNN models show promising performance, leading to an EER lower than 1% in all cases. The table also highlights that SincNet outperforms the other models, showing a relative performance improvement of about 11% over the standard CNN model. Note that the speaker verification system is derived from the speaker-id neural network using the *d-vector* technique. The *d-vector* [19, 24] is extracted from the last hidden layer of the speaker-id network. A speaker-dependent d-vector is computed and stored for each enrollment speaker by performing an L2 normalization and averaging all the d-vectors of the different speech chunks. The cosine distance

[3] at `https://github.com/mravanelli/SincNet/`.

[4] at `https://github.com/mravanelli/pytorch-kaldi/`.

	TIMIT	LibriSpeech
DNN-MFCC	0.99	2.02
CNN-FBANK	0.86	1.55
CNN-Raw	1.65	1.00
SincNet	**0.85**	**0.96**

Table 1: Classification Error Rate (CER%) of speaker identification systems trained on TIMIT (462 spks) and Librispeech (2484 spks) datasets. SincNet outperforms the competing alternatives.

	EER(%)
DNN-MFCC	0.88
CNN-FBANK	0.60
CNN-Raw	0.58
SINCNET	**0.51**

Table 2: Speaker Verification Equal Error Rate (EER%) on Librispeech datasets over different systems. SincNet outperforms the competing alternatives.

between enrolment and test d-vectors is then calculated, and a threshold is then applied on it to reject or accept the speaker. Ten utterances from impostors were randomly selected for each sentence coming from a genuine speaker. To assess our approach on a standard open-set speaker verification task, all the enrolment and test utterances were taken from a speaker pool different from that used for training the speaker-id DNN.

For the sake of completeness, experiments have also been conducted with standard i-vectors. Although a detailed comparison with this technology is out of the scope of this paper, it is worth noting that our best i-vector system achieves an EER=1.1%, rather far from what is achieved with DNN systems. It is well-known in the literature that i-vectors provide competitive performance when more training material is used for each speaker and when longer test sentences are employed [57–59]. Under the challenging conditions faced in this work, neural networks achieve better generalization.

4.2 Speech Recognition

Tab. 3 reports the speech recognition performance obtained by CNN and SincNet using the TIMIT and the DIRHA dataset [41]. To ensure a more accurate comparison between the architectures, five experiments varying the initialization seeds were conducted for each model and corpus. Table 3 thus reports the average speech recognition performance. Standard deviations, not reported here, range between 0.15 and 0.2 for all the experiments.

	TIMIT	DIRHA
CNN-FBANK	18.3	40.1
CNN-Raw waveform	18.3	40.5
SincNet-Raw waveform	**18.0**	**38.2**

Table 3: Speech recognition performance obtained on the TIMIT and DIRHA datasets.

For all the datasets, SincNet outperforms CNNs trained on both standard FBANK and raw waveforms. The latter result confirms the effectiveness of SincNet not only in close-talking scenarios but also in challenging noisy conditions characterized by the presence of both noise and reverberation. As emerged in Sec.2, SincNet is able to effectively tune its filter-bank front-end to better address the characteristics of the noise.

5 Conclusions and Future Work

This paper proposed SincNet, a neural architecture for directly processing waveform audio. Our model, inspired by the way filtering is conducted in digital signal processing, imposes constraints on the filter shapes through efficient parameterization. SincNet has been extensively evaluated on challenging speaker and speech recognition tasks, consistently showing some performance benefits.

Beyond performance improvements, SincNet also significantly improves convergence speed over a standard CNN, is more computationally efficient due to the exploitation of filter symmetry, and it is more interpretable than standard black-box models. Analysis of the SincNet filters, in fact, revealed

that the learned filter-bank is tuned to the specific task addressed by the neural network. In future work, we would like to evaluate SincNet on other popular speaker recognition tasks, such as VoxCeleb. Inspired by the promising results obtained in this paper, in the future we will explore the use of SincNet for supervised and unsupervised speaker/environmental adaptation. Moreover, although this study targeted speaker and speech recognition only, we believe that the proposed approach defines a general paradigm to process time-series and can be applied in numerous other fields.

Acknowledgement

This research was enabled in part by support provided by Calcul Québec and Compute Canada.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[2] D. Yu and L. Deng. *Automatic Speech Recognition - A Deep Learning Approach*. Springer, 2015.

[3] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-end attention-based large vocabulary speech recognition. In *Proc. of ICASSP*, pages 4945–4949, 2016.

[4] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. of ICML*, pages 1764–1772, 2014.

[5] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proc.of ICLR*, 2015.

[6] C. Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Leanpub, 2018.

[7] S. Chakraborty et al. Interpretability of deep learning models: A survey of results. In *Proc. of SmartWorld*, 2017.

[8] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. of ECCV*, 2014.

[9] Q.-S. Zhang and S.-C. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, Jan 2018.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proc. of ACM SIGKDD*, pages 1135–1144, 2016.

[11] Q. Zhang, R. Cao, F. Shi, Y. N. Wu, and S.-C. Zhu. Interpreting CNN Knowledge via an Explanatory Graph. In *Proc. of AAAI*, 2018.

[12] S. Sabour, N. Frosst, and G. E Hinton. Dynamic routing between capsules. In *Proc. of NIPS*, pages 3856–3866. 2017.

[13] S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Müller, and W. Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[15] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proc. of NIPS*, 2014.

[16] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Improving speech recognition by revising gated recurrent units. In *Proc. of Interspeech*, 2017.

[17] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2):92–102, April 2018.

[18] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, London, UK, UK, 1999. Springer-Verlag.

[19] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *Proc. of ICASSP*, pages 4052–4056, 2014.

[20] F. Richardson, D. A. Reynolds, and N. Dehak. A unified deep neural network for speaker and language recognition. In *Proc. of Interspeech*, pages 1146–1150, 2015.

[21] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *Proc. of Interspeech*, pages 999–1003, 2017.

[22] C. Zhang, K. Koishida, and J. Hansen. Text-independent speaker verification based on triplet convolutional neural network embeddings. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 26(9):1633–1644, 2018.

[23] G. Bhattacharya, J. Alam, and P. Kenny. Deep speaker embeddings for short-duration speaker verification. In *Proc. of Interspeech*, pages 1517–1521, 2017.

[24] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *Proc. of Interspech*, 2017.

[25] D. Palaz, M. Magimai-Doss, and R. Collobert. Analysis of CNN-based speech recognition system using raw speech as input. In *Proc. of Interspeech*, 2015.

[26] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *Proc. of Interspeech*, 2015.

[27] Y. Hoshen, R. Weiss, and K. W. Wilson. Speech acoustic modeling from raw multichannel waveforms. In *Proc. of ICASSP*, 2015.

[28] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and A. Senior. Speaker localization and microphone spacing invariant acoustic modeling from raw multichannel waveforms. In *Proc. of ASRU*, 2015.

[29] Z. Tüske, P. Golik, R. Schlüter, and H. Ney. Acoustic modeling with deep neural networks using raw time signal for LVCSR. In *Proc. of Interspeech*, 2014.

[30] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Proc. of ICASSP*, pages 5200–5204, 2016.

[31] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016.

[32] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *CoRR*, abs/1612.07837, 2016.

[33] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur. Acoustic modelling from the signal domain using CNNs. In *Proc. of Interspeech*, 2016.

[34] H. Dinkel, N. Chen, Y. Qian, and K. Yu. End-to-end spoofing detection with raw waveform CLDNNS. *Proc. of ICASSP*, 2017.

[35] H. Muckenhirn, M. Magimai-Doss, and S. Marcel. Towards directly modeling raw speech signal for speaker verification using CNNs. In *Proc. of ICASSP*, 2018.

[36] J.-W. Jung, H.-S. Heo, I.-H. Yang, H.-J. Shim, , and H.-J. Yu. A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result. In *Proc. of ICASSP*, 2018.

[37] J.-W. Jung, H.-S. Heo, I.-H. Yang, H.-J. Shim, and H.-J. Yu. Avoiding Speaker Overfitting in End-to-End DNNs using Raw Waveform for Text-Independent Speaker Verification. In *Proc. of Interspeech*, 2018.

[38] M. Ravanelli and Y. Bengio. Speaker Recognition from raw waveform with SincNet. In *Proc. of SLT*, 2018.

[39] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM, 1993.

[40] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *Proc. of ICASSP*, pages 5206–5210, 2015.

[41] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo. The DIRHA-ENGLISH corpus and related tasks for distant-speech recognition in domestic environments. In *Proc. of ASRU 2015*, pages 275–282.

[42] M. Ravanelli, P. Svaizer, and M. Omologo. Realistic multi-microphone data simulation for distant speech recognition. In *Proc. of Interspeech*, 2016.

[43] L. R. Rabiner and R. W. Schafer. *Theory and Applications of Digital Speech Processing*. Prentice Hall, NJ, 2011.

[44] S. K. Mitra. *Digital Signal Processing*. McGraw-Hill, 2005.

[45] M. Ravanelli, D. Serdyuk, and Y. Bengio. Twin regularization for online speech recognition. In *Proc. of Interspeech*, 2018.

[46] T. N. Sainath, B. Kingsbury, A. R. Mohamed, and B. Ramabhadran. Learning filter banks within a deep neural network framework. In *Proc. of ASRU*, pages 297–302, 2013.

[47] H. Yu, Z. H. Tan, Y. Zhang, Z. Ma, and J. Guo. DNN Filter Bank Cepstral Coefficients for Spoofing Detection. *IEEE Access*, 5:4779–4787, 2017.

[48] H. Seki, K. Yamamoto, and S. Nakagawa. A deep neural network integrated with filterbank learning for speech recognition. In *Proc. of ICASSP*, pages 5480–5484, 2017.

[49] V. Papyan, Y. Romano, and M. Elad. Convolutional neural networks analyzed via convolutional sparse coding. *Journal of Machine Learning Research*, 18:83:1–83:52, 2017.

[50] S. Mallat. Understanding deep convolutional networks. *CoRR*, abs/1601.04920, 2016.

[51] D. Palaz and R. Magimai-Doss, M.and Collobert. End-to-end acoustic modeling using convolutional neural networks for automatic speech recognition. 2016.

[52] H. Muckenhirn, M. Magimai-Doss, and S. Marcel. On Learning Vocal Tract System Related Speaker Discriminative Information from Raw Signal Using CNNs. In *Proc. of Interspeech*, 2018.

[53] M. Ravanelli. *Deep learning for Distant Speech Recognition*. PhD Thesis, Unitn, 2017.

[54] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. A network of deep neural networks for distant speech recognition. In *Proc. of ICASSP*, pages 4880–4884, 2017.

[55] M. Ravanelli and M. Omologo. Contaminated speech training methods for robust DNN-HMM distant speech recognition. In *Proc. of Interspeech 2015*, pages 756–760.

[56] M. Ravanelli, T. Parcollet, and Y. Bengio. The PyTorch-Kaldi Speech Recognition Toolkit. In *arXiv:1811.07453*, 2018.

[57] A. K. Sarkar, D Matrouf, P.M. Bousquet, and J.F. Bonastre. Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification. In *Proc. of Interspeech*, pages 2662–2665, 2012.

[58] R. Travadi, M. Van Segbroeck, and S. Narayanan. Modified-prior i-Vector Estimation for Language Identification of Short Duration Utterances. In *Proc. of Interspeech*, pages 3037–3041, 2014.

[59] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason. i-vector based speaker recognition on short utterances. In *Proc. of Interspeech*, pages 2341–2344, 2011.

[60] M. Matassoni, R. Astudillo, A. Katsamanis, and M. Ravanelli. The DIRHA-GRID corpus: baseline and tools for multi-room distant speech recognition using distributed microphones. In *Proc. of Interspeech 2014*, pages 1616–1617.

[61] E. Zwyssig, M. Ravanelli, P. Svaizer, and M. Omologo. A multi-channel corpus for distant-speech interaction in presence of known interferences. In *Proc. of ICASSP 2015*, pages 4480–4484.

[62] L. Cristoforetti, M. Ravanelli, M. Omologo, A. Sosi, A. Abad, M. Hagmueller, and P. Maragos. The DIRHA simulated corpus. In *Proc. of LREC 2014*, pages 2629–2634.

[63] Douglas P. and J. M. Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the Workshop on Speech and Natural Language*, Proc. of HLT, pages 357–362, 1992.

[64] M. Ravanelli, A. Sosi, P. Svaizer, and M. Omologo. Impulse response estimation for robust speech recognition in a reverberant environment. In *Proc. of EUSIPCO 2012*.

[65] M. Ravanelli and M. Omologo. On the selection of the impulse responses for distant-speech recognition based on contaminated speech training. In *Proc. of Interspeech 2014*, pages 1028–1032.

[66] J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[67] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of ICML*, pages 448–456, 2015.

[68] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Batch-normalized joint training for dnn-based distant speech recognition. In *Proc. of SLT*, 2016.

[69] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. of ICML*, 2013.

[70] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of AISTATS*, pages 249–256, 2010.

[71] D. Povey et al. The Kaldi Speech Recognition Toolkit. In *Proc. of ASRU*, 2011.

[72] A. Larcher, K. A. Lee, and S. Meignier. An extensible speaker identification sidekit in python. In *Proc. of ICASSP*, pages 5095–5099, 2016.

Appendix

Corpora

To provide experimental evidence on datasets characterized by different numbers of speakers, this paper considers the TIMIT (462 spks, *train* chunk) [39] and Librispeech (2484 spks) [40] corpora. For speaker verification experiments, non-speech intervals at the beginning and end of each sentence were removed. Moreover, the Librispeech sentences with internal silences lasting more than 125 ms were split into multiple chunks. To address text-independent speaker recognition, the calibration sentences of TIMIT (i.e., the utterances with the same text for all speakers) have been removed. For the latter dataset, five sentences for each speaker were used for training, while the remaining three were used for test. For the Librispeech corpus, the training and test material have been randomly selected to exploit 12-15 seconds of training material for each speaker and test sentences lasting 2-6 seconds. To evaluate the performance in a challenging distant-talking scenario, speech recognition experiments have also considered the DIRHA dataset [41]. This corpus, similarly to the other DIRHA corpora [60, 61], has been developed in the context of the DIRHA project [62] and is based on WSJ sentences [63] recorded in a domestic environment. Training is based on contaminating WSJ-5k utterances with realistic impulse responses [64, 65], while the test phase test phase consists of 409 WSJ sentences recorded by native speakers in a domestic environment (the average SNR is 10 dB).

SincNet Setup

The waveform of each speech sentence was split into chunks of 200 ms (with 10 ms overlap), which were fed into the SincNet architecture. The first layer performs sinc-based convolutions as described in Sec. 2, using 80 filters of length $L = 251$ samples. The architecture then employs two standard convolutional layers, both using 60 filters of length 5. Layer normalization [66] was used for both the input samples and for all convolutional layers (including the SincNet input layer). Next, three fully-connected layers composed of 2048 neurons and normalized with batch normalization [67, 68] were applied. All hidden layers use leaky-ReLU [69] non-linearities. The parameters of the sinc-layer were initialized using mel-scale cutoff frequencies, while the rest of the network was initialized with the well-known "Glorot" initialization scheme [70]. Frame-level speaker and phoneme classifications were obtained by applying a softmax classifier, providing a set of posterior probabilities over the targets. For speaker-id, a sentence-level classification was simply derived by averaging the frame predictions and voting for the speaker which maximizes the average posterior. Training used the RMSprop optimizer, with a learning rate $lr = 0.001$, $\alpha = 0.95$, $\epsilon = 10^{-}7$, and minibatches of size 128. All the hyper-parameters of the architecture were tuned on TIMIT, then inherited for Librispeech as well. The speaker verification system was derived from the speaker-id neural network using the *d-vector* approach [19, 24], which relies on the output of the last hidden layer and computes the cosine distance between test and the claimed speaker d-vectors. Ten utterances from impostors were randomly selected for each sentence coming from a genuine speaker. Note that to assess our approach on a standard open-set speaker-id task, all the impostors were taken from a speaker pool different from that used for training the speaker-id DNN.

Baseline Setups

We compared SincNet with several alternative systems. First, we considered a standard CNN fed by the raw waveform. This network is based on the same architecture as SincNet, but replacing the sinc-based convolution with a standard one.

A comparison with popular hand-crafted features was also performed. To this end, we computed 39 MFCCs (13 static+Δ+$\Delta\Delta$) and 40 FBANKs using the Kaldi toolkit [71]. These features, computed every 25 ms with 10 ms overlap, were gathered to form a context window of approximately 200 ms (i.e., a context similar to that of the considered waveform-based neural network). A CNN was used for FBANK features, while a Multi-Layer Perceptron (MLP) was used for MFCCs. Note that CNNs exploit local correlation across features and cannot be effectively used with uncorrelated MFCC features. Layer normalization was used for the FBANK network, while batch normalization was employed for the MFCC one. The hyper-parameters of these networks were also tuned using the aforementioned approach.

For speaker verification experiments, we also considered an i-vector baseline. The i-vector system was implemented with the SIDEKIT toolkit [72]. The GMM-UBM model, the Total Variability (TV) matrix, and the Probabilistic Linear Discriminant Analysis (PLDA) were trained on the Librispeech data (avoiding test and enrollment sentences). GMM-UBM was composed of 2048 Gaussians, and the rank of the TV and PLDA eigenvoice matrix was 400. The enrollment and test phase is conducted on Librispeech using the same set of speech segments used for DNN experiments.

Toward end-to-end interpretable convolutional neural networks for waveform signals

Linh Vu
Monash University
linh.vu@monash.edu

Thu Tran
Singapore Management University
ndttran.2019@phdcs.smu.edu.sg

Lim Wern Han
Monash University
lim.wern.han@monash.edu

Raphaël C.-W. Phan
Monash University
raphael.phan@monash.edu

***Abstract*—This paper introduces a novel convolutional neural networks (CNN) framework tailored for end-to-end audio deep learning models, presenting advancements in efficiency and explainability. By benchmarking experiments on three standard speech emotion recognition datasets with five-fold cross-validation, our framework outperforms Mel spectrogram features by up to seven percent. It can potentially replace the Mel-Frequency Cepstral Coefficients (MFCC) while remaining lightweight. Furthermore, we demonstrate the efficiency and interpretability of the front-end layer using the PhysioNet Heart Sound Database, illustrating its ability to handle and capture intricate long waveform patterns. Our contributions offer a portable solution for building efficient and interpretable models for raw waveform data.**

***Index Terms*—speech recognition, audio classification, signal processing, interpretable neural networks, convolutional neural networks**

I. Introduction

The use of deep learning and representation learning has significantly impacted the signal processing field, similar to the success observed in the fields of Natural Language Processing (NLP) and Computer Vision (CV). Deep neural networks can automatically learn hierarchical representations from raw data. However, simply using deep learning as a one-tool-fit-all solution often leads to poor interpretability, i.e., monitoring how the model works is difficult.

Recent developments have led to the creation of efficient and easy-to-understand models for processing raw waveform signals. Cheuk et al. [1] have introduced nnAudio, a framework that uses 1D convolutional neural networks to extract spectrograms on-the-fly. Leiber et al. [2] have introduced a differentiable modification of the Short-Time Fourier Transform (STFT) that makes it possible to optimize the window length parameter through gradient descent. These works primarily revolve around rethinking the fusion of spectrograms and neural networks, placing emphasis on employing an adapted spectrogram as the initial layer for input signals. Seki et al. [3] showed that Gaussian filters with three trainable parameters of gain, bandwidth, and centre frequency can outperform Mel-filterbank in both clean and noise-corrupted environments. Inspired by standard filtering in digital signal processing, Ravanelli and Bengio [4] proposed a novel CNN called SincNet that uses rectangular band-pass filters as kernels for the first convolution layer with two learnable parameters: high and low cut-off frequencies. Right from the first layer, SincNet

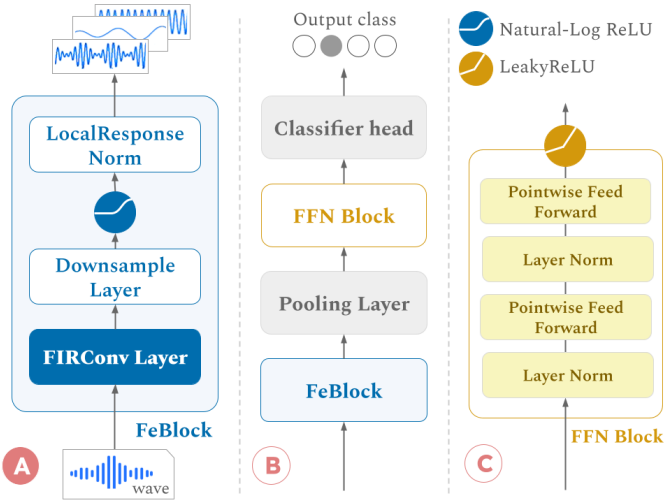


Fig. 1: The proposed IConNet architecture for end-to-end audio classification: *A*- the front-end block containing the FIR-conv layer; *B*- the proposed general architecture for end-to-end audio classification; *C*- the classifier used in the experiments.

can already learn meaningful filters and thus converge faster than the conventional CNN. In a comparison between different parametric modulated kernel-based filters including SincNet (rectangular filters), Sinc2Net (triangular filters), GammaNet (gammatone filters) and GaussNet (Gaussian filters); it is found that SincNet is the best for the speech recognition task [5].

According to Ravanelli and Bengio [4], the frequency gain of the SincNet filters is only defined in subsequent layers of the neural network, which are conventional CNN layers. To further contribute to this research direction, we propose a new interpretable CNN architecture called IConNet that utilizes a finite impulse response (FIR)-based kernel with learnable window functions. FIR filters play a crucial role in signal processing as they enable the extraction of critical information; and different filter designs can enhance or mitigate unwanted effects. With adaptive window functions, the filters can dynamically adjust to varying signal profiles depending on the data and the problem. The primary benefit of this approach is the transparency in the way the model learns – which frequency bands it focuses on and which will be cut off.

We illustrate the effectiveness of this approach in two health-related problems: speech emotion recognition and abnormal heart sound detection.

In the next section II, we will describe the proposed method. Section III is about the Speech Emotion Recognition (SER) experiment, followed by section IV about abnormal heart sound detection.

II. The IConNet architecture

The proposed Interpretable Convolutional Neural Network (IConNet) architecture in this research is designed to leverage insights from audio signal processing; aiming to improve end-to-end deep neural networks' ability to extract features and patterns from raw waveform signals. The foundation of our method draws inspiration from the standard signal processing process, in which the input audio signal undergoes a windowing operation to segment it into smaller frames and simultaneously mitigate spectral leakage to improve frequency resolution. A key novelty lies in using the Generalized Cosine Window function as parametrization for the convolution kernels to enable the neural networks to choose the most suitable shape for each frequency band.

The convolution layer of the proposed model has restricted-shaped kernels which is a band-pass filter defined by non-learnable low cut-off frequency f_0 and frequency bandwidth f_δ. The filter shape and frequency gain are determined by the window function W with p learnable parameters ϕ_p. Let $H = \{h_k : \quad k = 1, ..., K\}$ denotes the kernel (filter) of width K in the time domain. V_n denotes the output of the convolution layer for each n input time-domain value. $H(k, f_0, f_\delta, \phi)$ is parametrically modulated by $sinc$ as a non-learnable function and W with learnable parameters.

$$V_n = X \cdot H = \sum_{k=1}^{K} X_k \cdot h_{n-k} \tag{1}$$

$$H = T * W = T(k, f_0, f_\delta) * W(k, \phi), \qquad 0 \le k < K \tag{2}$$

$$\begin{aligned} T(k, f_0, f_\delta) &= 2(f_0 + f_\delta) sinc(2\pi k(f_0 + f_\delta)) \\ &\quad -2f_0 sinc(2\pi k f_0), \qquad sinc(a) = \sin(a)/a \end{aligned} \tag{3}$$

$$W(k, \phi) = \left\{ w_k \middle| w_k = \sum_{i=0}^{p} (-1)^i \phi_i \cos \frac{2\pi ik}{K-1} \right\} \tag{4}$$

Thus, for each kernel, there are only p parameters ϕ_p and two band parameters f_0 and f_δ to train via gradient descent. f_0 and f_δ can also be non-learnable, which is how we can incorporate prior domain knowledge into the deep neural network design to solve specific problems and control what information is fed into the model to prevent unwanted effects from the first layers.

After separating signals into different channels, we use a downsampling layer to reduce the data in the time domain to a lower sample rate. This compression step decreases the data's dimensions, making it easier to process in subsequent steps while retaining the essential features needed for precise analysis and classification. It's worth noting that the convolution filters from the previous step also serve as anti-aliasing filters like traditional signal downsampling. In the second-to-last step, we use an NLRelu activation function recommended in [6] to mimic the amplitude-to-decibel conversion since human hearing functions on a logarithmic scale. Finally, we apply the Local Respond Normalization function [7] to the output before forwarding it to the next layer. This entire process can be repeated by stacking these blocks on top of each other to achieve further pattern decorrelation from previous decomposed channels while maintaining a more compressed representation. The outputs of the front-end blocks can be combined together as long as they are resampled to the same sampling rate to preserve the temporal characteristic. Depending on the task, these front-end blocks can be incorporated into any deep neural network architecture.

Figure 1 illustrates the proposed architecture, with the part ***A*** on the left describing the front-end block. The middle part ***B*** of the figure 1 is the high-level deep neural network architecture that incorporates the front-end block to process raw waveform signal data. Part ***C*** depicts a simple classifier block consisting of a pooling layer, a two-layer feed-forward neural network with layer norm and a LeakyRelu activation function. The following experiments are designed to demonstrate the feature extraction ability of the front-end blocks and evaluate the effectiveness of this new architecture.

We have selected two classification problems and designed a classifier consisting of a pooling layer, a two-layer feed-forward neural network with layer norm and a LeakyRelu activation function. In both experiments, we use k-fold cross-validation settings and report three different metrics: unweighted accuracy (UA), unweighted F1 (UF1) and weighted-F1 (F1). For training the neural networks, we employed RAdam optimizer [8] with OneCycleLR learning rate scheduler [9], Cross Entropy loss and trained each model to up to 60 epochs on each fold.

III. Speech Emotion Recognition

A. Background

Emotions play a pivotal role in revealing essential cues about a speaker's intentions, attitudes, and mental state, particularly in the context of AI health. This significance is underscored by a recent study from Gheorghe et al. [10] where a system leveraging deep neural networks successfully identifies depression from speech samples with an unweighted accuracy of 91.25% using Mel-frequency cepstral coefficients (MFCC). MFCC is the most popular feature based on the short-time Fourier Transform spectrogram and the Mel filterbank, which was designed based on human perception of our hearing system. As humans perceive sound on a logarithmic scale, Mel filterbank uses narrower bandwidths at the lower frequencies to capture more information. Its filters have a triangular shape and are used in many tasks. However, according to [3], [11], in

the data-driven filterbanks, each filterbank's centre frequencies and shapes are different depending on the tasks and the presence of background noise.

Based on these insights, we've developed three IConNet variants detailed in Table I. These variants assess the influence of adjustable bands versus adaptive windows on CNN model classification outcomes. The front-end kernels in all IConNet models are initialized with the Mel-filterbank, with additional filters allocated to the lower frequency range, prioritizing crucial speech signals.

B. Experiment setup

1) Datasets: Our first experiment incorporates the proposed method, evaluating its performance on three standard Speech Emotion Recognition (SER) datasets with similar settings described in [12] [1]. Firstly, the **RAVDESS** dataset [13] features high-quality audio speech with eight emotional expressions from 24 North-American actors. Secondly, the **CREMA-D** dataset [14] comprises recordings from 91 speakers with diverse races and ethnicities, representing real-world audio recordings where the recordings are often noise-corrupted. Lastly, **IEMOCAP** [15] is a popular SER dataset with 12-hour conversational speech audio from 10 speakers. We use four classes of data, namely *happy, sad, angry, neutral*, which are shared between the aforementioned three datasets for a fair comparison.

2) Classifiers and evaluation: For the classifier, we use a dense neural network consisting of two layers with 512 nodes in each layer followed by layer norm, as described in the diagram 1. To evaluate the performance of the proposed models, we adopted 5-fold cross-validation with stratified train-test splitting to ensure the proportion of each class in both sets. We trained each model for up to 60 epochs with early stopping, then evaluated the model on the test set and reported the average metrics across folds. We benchmarked IConNet with different settings against Mel and MFCC features extracted using the *TorchAudio* library [16] with different resolutions as described in the table I.

Table I: Models used in the SER experiment

Model	Description
Mel-*K*	Mel spectrogram with *K* Mel frequency bins
MFCC-*K*	MFCC from Mel-*K*
IConNet-B-*K*	IConNet with *K* learnable-band filters
IConNet-W-*K*	IConNet with *K* learnable-window filters
IConNet-BW-*K*	IConNet with *K* learnable-band-window filters

C. Experiment results

Table II provides detailed experiment result on the three datasets mentioned above when using early stopping. On the RAVDESS dataset, the IConNet model with 456 adaptive-window FIR kernels achieved an unweighted accuracy of 66.83%, which is 4.83% higher than the adjustable-band-FIR model with the same number of kernels. When the number of kernels was reduced to 256, the UA of the IConNet-W dropped 1%, while the IConNet-B increased 1.73%. For the IConNet-W models and Mel-spectrogram models, higher resolution helped the models make better predictions. The size of Mel-256/IConNet-256 and Mel-456/IConNet-456 models is 1.6 MB and 2 MB, respectively. Despite having the same size, Mel models performed poorly compared to all IConNet models, even after being trained for more epochs.

The bar chart 2 compares the F1 scores of Mel-456, MFCC-456, and IConNet-456 models on the RAVDESS and CREMA-D datasets. After 60 epochs, the performance of MFCC on the RAVDESS dataset increased from 46.13% to 67.14%, which is a more than 21% improvement. However, the gain for MFCC-456 model on CREMA-D is less than 1%. The IConNet-W model still gave the highest F1 score on both the RAVDESS

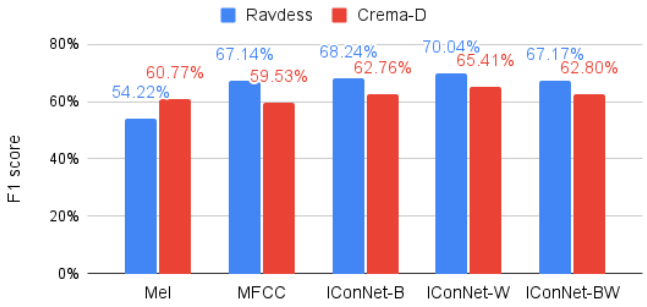


Fig. 2: Result on RAVDESS and CREMA-D datasets after 60 epochs

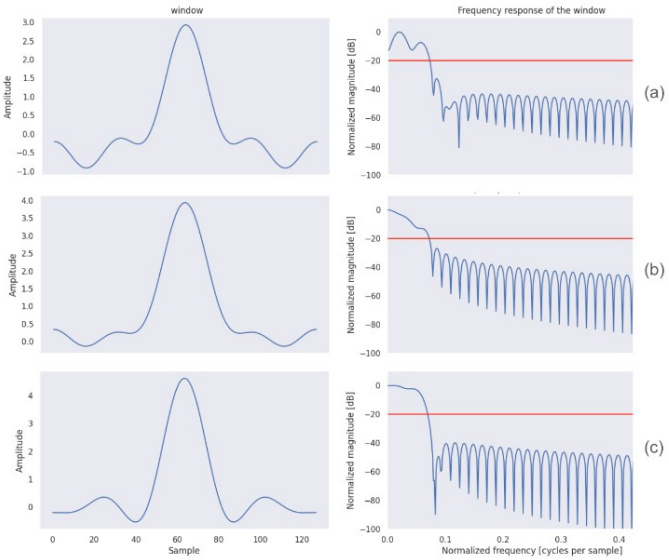


Fig. 3: Comparison of Window Shape and Frequency Response of Filters from Different Bands. The chart displays the frequency response of low-range (a), mid-range (b), and high-range (c) frequency bands. The red line at -20dB represents the threshold at which noise is perceived as not noticeable.

[1] It is worth mentioning that in [12], Vu et al. benchmarked the performance of handcrafted features after 500 epochs on the 20% test set after 10-fold cross validation on 80% of the train set.

Table II: Results of hand-crafted features and IConNet on three datasets in percentage with early stopping (%)

Dataset	RAVDESS			CREMA-D			IEMOCAP			Average		
Model	**UA**	**UF1**	**F1**	**UA**	**UF1**	**F1**	**UA**	**UF1**	**F1**	**UA**	**UF1**	**F1**
Mel-256	51.59	51.73	53.22	59.07	59.07	58.83	50.85	50.85	51.13	54 ±5	54 ±5	54 ±4
Mel-456	52.00	49.09	51.67	59.98	59.98	59.60	52.85	52.85	53.03	55 ±4	54 ±6	55 ±4
MFCC-256	49.29	46.01	47.58	56.47	56.27	56.70	**56.68**	**56.41**	**56.25**	54 ±4	53 ±6	54 ±5
MFCC-456	45.21	42.27	46.13	56.70	56.57	56.86	56.34	56.07	56.14	53 ±7	52 ±8	53 ±6
IConNet-B-256	63.73	63.40	65.01	61.67	61.31	61.53	53.72	53.37	53.37	60 ±5	59 ±5	60 ±6
IConNet-B-456	62.00	62.46	64.65	61.40	61.67	61.94	53.17	53.12	53.51	59 ±5	59 ±5	60 ±6
IConNet-W-256	**65.83**	**65.02**	**66.65**	**62.30**	**62.30**	**62.06**	56.44	55.94	56.17	**62 ±5**	**61 ±5**	**62 ±5**
IConNet-W-456	**66.83**	**66.15**	**67.37**	**62.34**	**62.34**	**62.56**	**56.67**	**56.32**	**56.56**	**62 ±5**	**62 ±5**	**62 ±5**
IConNet-BW-256	64.84	64.83	66.35	61.08	61.08	60.75	52.78	52.77	53.37	60 ±6	60 ±6	60 ±7
IConNet-BW-456	62.36	62.07	63.73	61.28	61.28	61.00	53.26	53.76	54.21	59 ±5	59 ±5	60 ±5

Notation: **bold-underline**: best results, **bold**: second-best results

and CREMA-D datasets at 70.04% and 65.41%, respectively.

The table in II clearly indicates that IConNet-W-456 and MFCC-256 attained the highest unweighted accuracy on IEMOCAP with only a 0.01% difference between them. On average, the IConNet-W models outperformed other IConNet variants by roughly 2%. Overall, IConNet models gave better results than Mel-spectrogram and MFCC models, especially on the CREMA-D dataset.

On the interpretable ability of the IConNet, Figure 3 demonstrates the alterations in the shape of windows that are used to extract important information from different frequency ranges. The window *(a)* is learned during the model training process for the low-frequency range and exhibits a complex shape, determining the frequency gain at each point. This complex shape enhances the robustness of the model because the low-frequency range contains crucial speech signals that are susceptible to noise. On the other hand, the window *(c)* is tailored for the higher frequency range, serving as a narrow-band filter that effectively mitigates spectral leakage with its ideal sidelobes.

In summary, the experiment results have proven the appropriate use of the proposed IConNet, especially the IConNet with adaptive-window FIR kernel for SER. Moreover, the IConNet end-to-end models are portable, with the model size being 30% smaller than the hand-crafted feature set models proposed by Vu et al. in [12].

IV. Abnormal heart sound detection

A. Background

Cardiovascular diseases are a leading global health concern, demanding improved diagnostic precision. A *heart murmur* is an unusual sound heard during the heartbeat cycle, indicating underlying cardiac conditions. Various ML approaches have been developed to identify heart murmurs from stethoscope recordings. As the recording duration is quite long (30 seconds on average), there are often many preprocessing steps, such as heartbeat segmentation to trim the waveform to a smaller size (3 to 5 seconds) and bandpass filtering as the most critical signals lay in the range from 10 Hz to 400 Hz. After that, MFCC and its derivatives are extracted to apply CNN or LSTM models or a combination of both [17], [18].

B. Proposed model

This experiment mainly examines the potential role of the IConNet in refining abnormal heart sound detection. Our proposed model is an end-to-end lightweight neural network architecture that eliminates the need for heart sound segmentation, low-pass filtering and MFCC-based feature extraction. We designed an IConNet-W model with two front-end blocks with 128 and 32 kernels respectively, a max-pooling layer and a 2-layer FFN classifier with 256 nodes on each layer. The total number of parameters is below 200K.

C. Experiment setup

We employ the widely-used **PhysioNet/CinC Challenge** dataset [19] for heart sound classification evaluation. This dataset comprises 2575 *normal* and 665 *abnormal* samples. To validate the effectiveness of the IConNet in identifying relevant features, we resample the waveform from 2000 Hz to 16000 Hz and conduct 4-fold cross-validation, reporting UA, UF1, and F1 metrics. Deng et al. [20] serve as our baseline, utilizing a preprocessing pipeline with a bandpass filter, MFCC features, and a CRNN model consisting of three 2D CNN layers and two LSTM layers. We also include the MFCC performance for comparison. Preprocessing for other models involves waveform trimming and downsampling, excluding the IConNet model.

D. Experiment results

Based on the results presented in Table III, it is clear that the baseline model [20] performed better than the MFCC + FFN model, thanks to its preprocessing steps that included bandpass filtering and the use of MFCC deltas. The baseline model achieved 90.06% F1 score. However, our proposed architecture surpassed both models with an F1 score of 92.05%, which is 2% higher than the baseline model. While this result still not yet outperforms the state-of-the-art Resnet result reported by Li et al. in [18], it has successfully demonstrated the effectiveness of our proposed method in classifying heart

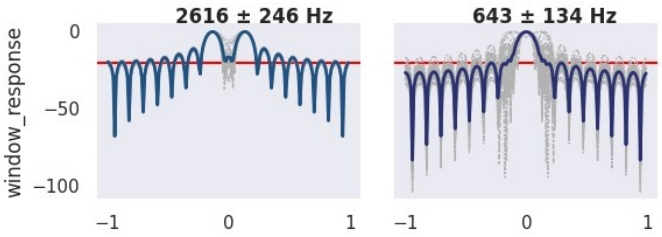


Fig. 4: Frequency response of filters from different bands

sound data. Furthermore, the visualization of the front-end filters confirms that it allocates band-pass filters that actively change the window shapes to extract essential information in the range of 643 ±134 Hz. The windows have learned to transform into band-stop filter shapes for the high-frequency range above 2000 Hz, which only contain meaningless artifacts from the resampling step. Understanding the features utilized by the backbone neural network model, which influences its decisions, is vital for ensuring reliable outcomes, particularly in health applications.

Table III: Abnormal heart sound detection result on the **PhysioNet** dataset

Model	UA	UF1	F1
MFCC + FFN	82.98	83.97	88.68
MFCC deltas + CRNN [20]	85.67	80.21	90.60
IConNet	87.48	81.12	92.05

V. Conclusions

Our proposed framework introduces a novel method for constructing end-to-end audio deep-learning models, showcasing its efficacy in healthcare applications such as speech emotion recognition and abnormal heart sound detection. Our findings reveal that the proposed CNN framework surpasses traditional methods utilizing the Mel spectrogram, and potentially MFCC (further experimentation required for confirmation), for both tasks. Moreover, visualization of CNN kernels underscores the value of transparent CNN architectures in healthcare settings, shedding light on the features extracted from input signals in mission-critical tasks.

Insightful observations arise regarding the front-end layer within our proposed framework. Our results indicate that front-end layers featuring learnable windows demonstrate superior performance compared to those employing learnable bands, diverging from the predominant focus in existing literature. We suggest exploring methods to incorporate prior knowledge into the front-end layer to improve performance, especially with the learnable window model. While the proposed framework offers promising results, it requires slightly more parameters and, thus, more training resources.

In conclusion, this study emphasizes the advantages of employing interpretable CNN for analyzing raw waveform data, showcasing their potential benefits in healthcare settings to foster transparency and trust in medical AI applications.

References

[1] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks," *IEEE Access*, vol. 8, pp. 161 981–162 003, 2020.

[2] M. Leiber, A. Barrau, Y. Marnissi, and D. Abboud, "A differentiable short-time fourier transform with respect to the window length," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 1392–1396.

[3] H. Seki, K. Yamamoto, and S. Nakagawa, "A deep neural network integrated with filterbank learning for speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5480–5484.

[4] M. Ravanelli and Y. Bengio, "Interpretable convolutional filters with sincnet," *Proc. of IRASL@NIPS*, 2019.

[5] E. Loweimi, P. Bell, and S. Renals, "On Learning Interpretable CNNs with Parametric Modulated Kernel-Based Filters," in *Proc. Interspeech 2019*, 2019, pp. 3480–3484.

[6] Y. Liu, J. Zhang, C. Gao, J. Qu, and L. Ji, "Natural-logarithm-rectified activation function in convolutional neural networks," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, 2019, pp. 2000–2008.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[8] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *Proc. International Conference on Learning Representations 2020*, 2020.

[9] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. SPIE, 2019, pp. 369–386.

[10] M. Gheorghe, S. Mihalache, and D. Burileanu, "Using deep neural networks for detecting depression from speech," in *2023 31st European Signal Processing Conference (EUSIPCO)*. IEEE, 2023, pp. 411–415.

[11] S. Sarangi, M. Sahidullah, and G. Saha, "Optimization of data-driven filterbank for automatic speaker verification," *Digital Signal Processing*, vol. 104, p. 102795, 2020.

[12] L. Vu, R. C.-W. Phan, L. W. Han, and D. Phung, "Improved speech emotion recognition based on music-related audio features," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 120–124.

[13] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PloS one*, vol. 13, no. 5, p. e0196391, 2018.

[14] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset," *IEEE transactions on affective computing*, vol. 5, no. 4, pp. 377–390, 2014.

[15] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, p. 335, 2008.

[16] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, "Torchaudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[17] T. Li, Y. Yin, K. Ma, S. Zhang, and M. Liu, "Lightweight end-to-end neural network model for automatic heart sound classification," *Information*, vol. 12, no. 2, p. 54, 2021.

[18] F. Li, Z. Zhang, L. Wang, and W. Liu, "Heart sound classification based on improved mel-frequency spectral coefficients and deep residual learning," *Frontiers in Physiology*, vol. 13, p. 1084420, 2022.

[19] C. Liu, D. Springer, Q. Li, B. Moody, R. A. Juan, F. J. Chorro, F. Castells, J. M. Roig, I. Silva, A. E. Johnson *et al.*, "An open access database for the evaluation of heart sound algorithms," *Physiological measurement*, vol. 37, no. 12, p. 2181, 2016.

[20] M. Deng, T. Meng, J. Cao, S. Wang, J. Zhang, and H. Fan, "Heart sound classification based on improved mfcc features and convolutional recurrent neural networks," *Neural Networks*, vol. 130, pp. 22–32, 2020.

Learnable filter-banks for CNN-based audio applications

Helena Peic Tukuljac[1], Benjamin Ricaud*[1,2], Nicolas Aspert[1], and Laurent Colbois[3]

[1]LTS2, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[2]Dept. of Physics and Technology, UiT The Arctic University of Norway, Tromsø, Norway
[3]Idiap Research Institute, Martigny, Switzerland

Abstract

We investigate the design of a convolutional layer where kernels are parameterized functions. This layer aims at being the input layer of convolutional neural networks for audio applications or applications involving time-series. The kernels are defined as one-dimensional functions having a band-pass filter shape, with a limited number of trainable parameters. Building on the literature on this topic, we confirm that networks having such an input layer can achieve state-of-the-art accuracy on several audio classification tasks. We explore the effect of different parameters on the network accuracy and learning ability. This approach reduces the number of weights to be trained and enables larger kernel sizes, an advantage for audio applications. Furthermore, the learned filters bring additional interpretability and a better understanding of the audio properties exploited by the network.

1 Introduction

In audio signal processing, time-frequency representations such as spectrograms are central tools. They have an intuitive interpretation and reveal insightful information to the human expert. It is not a surprise that many deep learning approaches to audio signals use such representations as well [5, 26]. It is also convenient as most of the deep network architectures have been developed for image processing and require 2D arrays of values as inputs. The network learns to detect time-frequency patterns, similarly to what is done on images. Depending on the task, it may then output a classification of a sound [25, 28], a denoised signal [15] or separated sources [4].

These representations are conventionally made using several types of transformations. In turn, each transformation may have several parameters that influence the representation. Until recently these transformations and their parameters were carefully chosen using expert knowledge.

The recent success of end-to-end learning where the raw audio file is the input of the network (e.g. Wavenet: [18, 22, 19, 30], Tasnet: [16, 17]), and more recently LEAF [34], demonstrates the efficiency of this approach for a variety of audio tasks. In this setting, one-dimensional convolutions are applied to raw audio signals and the network creates its own representation by learning the convolution kernels. However, kernel size needs to be much larger than the one used for image applications. Indeed, at a sampling rate of 44kHz, 44 samples represent 1 ms of audio signal. To capture audio patterns that have duration of 10, 100 ms or more, in particular low frequency patterns, either large kernels are needed or deeper convolutional architectures (to allow for combinations of kernels at many different positions in time). Both solutions lead to a large increase in the number of parameters to be learned and hence require more training time and more data. Dilated convolutions or "atrous" convolution employed in Wavenet have been introduced in order to increase the time length of the kernel without increasing the number of weights to learn. Finding alternative ways for unlocking the time-length limit is an important challenge for raw audio processing in deep learning.

Replacing free kernels by parameterized filters,

*Corresponding Author: benjamin.ricaud@uit.no

https://doi.org/10.7557/18.6279

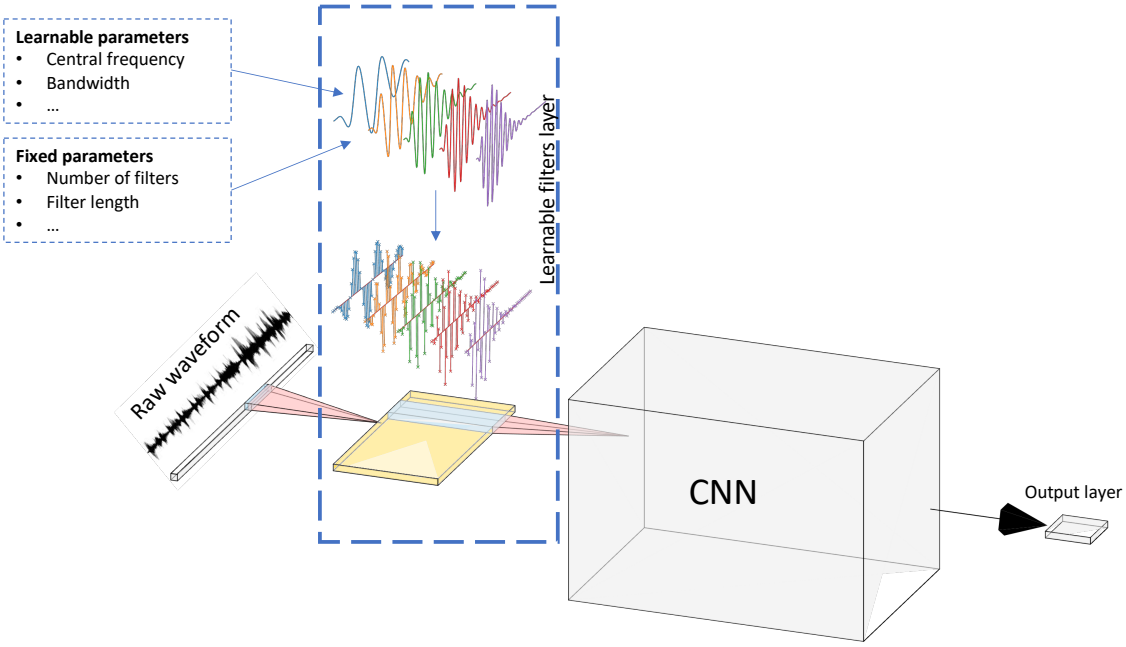


Figure 1: Network architecture using learnable filters. The first layer (in yellow) is a convolution layer where the kernels are defined as functions (colored waveforms) with learnable parameters, such as frequency and bandwidth. This functions are defined on a continuous domain and are then discretized to fit in the convolutional layer. The rest of the convolutional neural network (CNN) can have any standard architecture.

were the parameters are learnt, is an alternative way for reducing the computational burden. This is what we propose to investigate in the present work. The free kernels are replaced by filters with a few parameters in the first layer of the network, as shown in Fig. 1.

Learning parametric filters is halfway between 1) learning a standard convolutional layer, where all the weights of the kernels are learnable and unconstrained and 2) having a layer of kernels being fixed functions, where only the combination of these predefined functions may be learnt. The first approach is the most versatile but is computationally intensive and more prone to overfitting. The second approach used for example in the Scattering transform [3, 1, 6], or in [11] benefits from an inductive bias through the chosen kernel functions but is less flexible. The concept of learning filters aims at making an ideal compromise between flexibility and inductive bias. It has been first introduced in [29], [27], [33] and [12]. The first one introduces Gaussian filters in the input layer. Parameters are the amplitude, the Gaussian width and the modulation frequency. An increase of the classification accuracy is reported with the learned parameters. However, the filter learning is seen as a fine-tuning of the network after the first training pass with fixed Gaussian parameters. In the present work, the filter layer is fully integrated in the learning process, the parameters are learned from the beginning. In [27], the authors introduce a layer, called *SincNet*, made of sine modulated functions that approximate band-pass rectangular windows in the frequency domain. The learned parameters are the minimal and maximal cut-off frequencies of each band-pass filter. One of the main results is given by the cumulative frequency response of the SincNet filters. The network tends to focus more on particular regions of the frequency space, where formants are localized. This is interesting, as it shows

how the parameterized filters enable a precise interpretation of the learning and underline particular spectral properties of the data. The present work goes further in this direction. Eventually, [12] introduce Wavelet filter banks learned for speech recognition. Each kernel is a Wavelet defined by a single parameter, its scale. It shows evidences both of the efficiency of this approach and of the possibility to interpret the shape of the learned kernels. We compare the efficiency of the Wavelet filters with several other modulated windows and show that the former under-performs on audio signals. More recently and in line with our approach, [14] present complementary results, on a different dataset, with a focus on the sinc-square function, learning either the frequency or the bandwidth of the filters. Learnable Gabor functions combined with a modulus layer and a learnable PCEN layer showed state-of-the-art performance [34]. Comparing the effect of replacing a standard convolutional layer by a set of gammatone filters, [8] show an increase in the accuracy of a speech separation task. This suggests that an hybrid approach of learned gammatone filters would combine the best of both worlds.

We propose several parameterized functions and compare them to recent works on the same topics that use learnable filters. We confirm that this approach reaches state-of-the-art accuracy and even improves the accuracy on several audio classification tasks. We explore the influence of different parameters on the learning, such as the numbers of kernels and their length. Our classification experiments show that the number of filters required to obtain the best results remains small, around 20-30. We also demonstrate that the performances of different functions proposed in audio signal processing (modulated Gaussian, Gammatone) give close results and are better than Wavelets at classifying sounds. Last but not least, a relationship between the central frequency of the filter and its temporal width emerges with the learning. We provide evidences that the network converges to an auditory frequency spacing, close to the ERB (Equivalent Rectangular Bandwidth) and Bark scales found in psycho-acoustic studies [35, 9].

2 Learnable filter banks

We call the parameterized kernels in the convolutional layer *filters*, making a parallel with filters in signal processing. Indeed, these functions have the property of being band-pass filters and are well known in audio signal processing. One of the trainable parameters of each filter is the central frequency of the band-pass filter. The second parameter is the bandwidth of the filter (or a quantity closely related to it). Hence this set of filters forms a filter bank where the frequency and bandwidth of the filters may be adapted to the data and to the learning task. Note that the learned filterbank may not cover the entire spectrum but should focus on important spectral regions that are the most discriminative for classification.

We call the convolutional layer made of learnable filters, Learnable Filter (LF) layer. The input of the LF layer is a 1D audio signal and the output is a 2D representation. The output representation axes are time and filter number. Since each filter is associated to a particular frequency band, this 2D representation can be seen as a time-frequency one (or time-scale in the case of Wavelets). Initializing the filters by increasing frequencies (or scales), we can influence the frequency ordering to follow the filter number.

In all the definitions, N denotes the filter length and n is the variable (sample number). The time in seconds is expressed using the sampling frequency f_s with $t = n/f_s$. The frequency in Hertz is defined by $f \times f_s$, where $f \in [0, 0.5]$ is the normalized frequency in the formulas.

Mexican hat Wavelet. In order to compare to the state-of-the-art, we use the Mexican hat Wavelet introduced in the paper by [12]:

$$w(n) = \frac{2}{\pi^{1/4}\sqrt{3s}} \left(\frac{n^2}{s^2} - 1 \right) e^{-\frac{n^2}{s^2}}, \qquad (1)$$

with $n \in [-N/2, (N-1)/2]$ and $s > 0$ being the scale parameter.

Gaussian filter. Here, $n \in [-N/2, (N-1)/2]$. The Gaussian filter, also used in [29, 34], g is defined as follows:

$$g(n) = \sqrt{\frac{2}{\sqrt{\pi}\sigma}} e^{-\frac{n^2}{2\sigma^2}} \left(\cos(2\pi f n) + i \sin(2\pi f n) \right). \qquad (2)$$

The parameter $\sigma > 0$ is the variance of the Gaussian (temporal window width) and f is the oscillating frequency. It is a complex-valued function that we split into its real and imaginary parts. For each f and σ two kernels are created, one with the cosine modulation and one with the sine one.

Gammatone filter. The Gammatone filter [7, 24, 10] is another example of kernel. It is defined on the interval $n \in [0, N-1]$ as :

$$h(n) = A(\gamma, b) n^{\gamma-1} e^{-2\pi b n} \cos(2\pi f n) \qquad (3)$$

where A is the normalization[1], $A(\gamma, b) = \sqrt{2(4\pi b)^{(2\gamma+1)}/\Gamma(2\gamma+1)}$. The parameter γ is the order of the Gammatone. It can be learned or fixed to e.g. 2 or 4. These two latter values are the best suited ones for modeling the human hearing related filter bank [23]. The other learnable parameters are b, related to the width of the function, and f the frequency. The symbol Γ denotes the Gamma function. The bandwidth B of h depends linearly on b and is given by the following formula [7]:

$$B(\gamma, b) = 2(2^{1/\gamma} - 1)^{1/2} b. \qquad (4)$$

Remark 1: All the functions are defined and normalized in the continuous domain. In our application, the filters are discretized and truncated in order to be implemented in the convolution layer. Since they all vanish away from zero, it remains a good approximation, provided that the function's width does not exceed the fixed filter length N.

Remark 2: The modulated window functions are defined with a cosine (real part) and a sine (imaginary part) term, relating them to the Fourier transform, the spectral domain and the standard definition of filters in signal processing. For the sake of simplicity, in our experiments, we have chosen to use only the cosine term. The absence of the sine term did not affect the accuracy of our classification results. The network is able to adapt and detect discriminative patterns with a shifted cosine modulation.

Remark 3: It is important to distinguish the filter length N from the filter temporal width σ or b (or s for the scale). The filter length is fixed, can not be learned and is the size of the vector on which the filter is defined. The temporal width is learned and specifies the spread of the function over the vector of size N.

[1]This is an approximation of the normalization obtained by computing the integral of the continuous function $t^{\gamma-1}e^{-2\pi bt}$, using the following result: $\int_0^\infty t^n e^{-bt} dt = \frac{\Gamma(n+1)}{b^{n+1}}$.

3 Experiments

We apply our LF layer to several classification tasks described in the following sub-sections. We assess it on standard tasks found in the literature presented in the introduction. We have chosen 2 freely available speech datasets: *AudioMNIST* [2] and *Google Speech Commands v2* [32]. Both datasets contain words pronounced by different speakers. These datasets are dedicated to limited-vocabulary speech recognition tasks and the goal is to train the network to correctly recognize the word present in each audio sequence.

In order to compare the impact of the LF layer on the learning and classification results, we use existing network architectures and modify the first layer. For networks with raw audio input, the first convolutional layer (performing a standard 1D convolution) is replaced by our proposed parameterized convolution layer, as illustrated in Fig. 1. Our layer is then followed by a non-linear ReLU activation function. A stride parameter is available allowing to define the overlap in time of consecutive convolutions. The code needed to reproduce the experiments is publicly available on GitHub[2].

3.1 AudioMNIST Results

The original AudioMNIST paper [2] performs digit classification using raw audio as input to a network called AudioNet. The code[3] supplied with the paper has been re-used to perform 5-fold validation on the data. AudioNet is made of six convolutional layers, each convolution being followed by a max-pooling layer, and two dense layers, connected to an output layer. In all tests performed using this dataset, the models were trained using the Adam optimizer with default parameters during 50 epochs. Batch size used was set to 256 and loss function used was the categorical cross-entropy.

[2]`https://github.com/epfl-lts2/learnable-filterbanks`

[3]`https://github.com/soerenab/AudioMNIST`

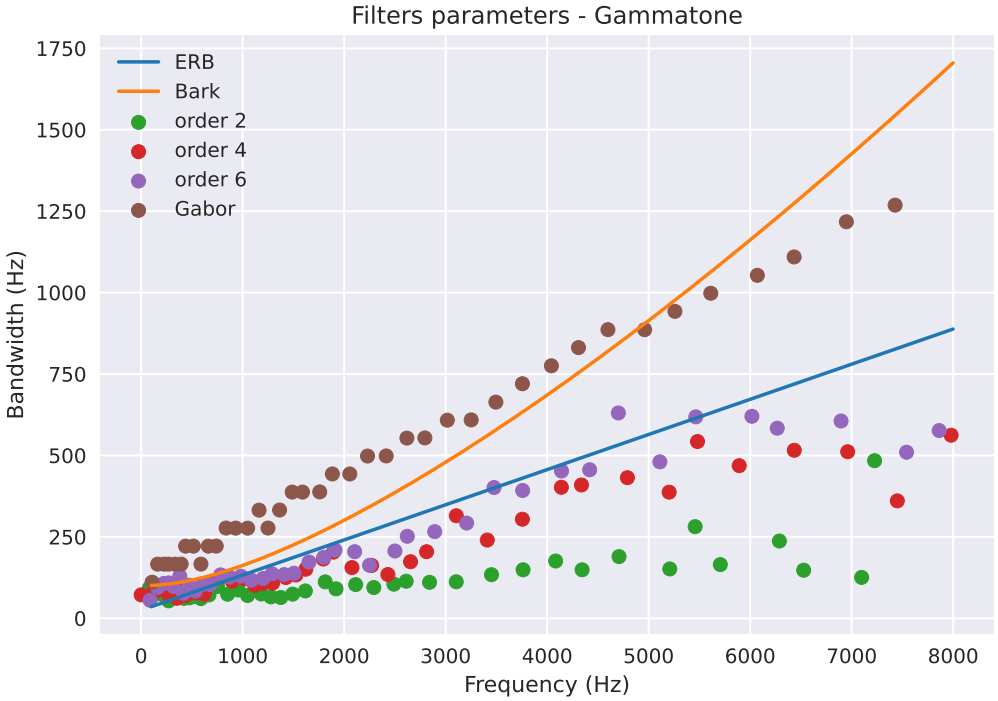


Figure 2: Bandwidth and frequency of learned filters.The curves are the psycho-acoustical relationships given by the ERB and Bark scales.

Test accuracy was then computed after this training phase and the same process was repeated for each fold.

On the AudioMNIST dataset sampled at 8 kHz, AudioNet has ca. 17 million trainable parameters. The original paper from [2] claims an accuracy of 92.53%±2.04%, whereas our implementation of AudioNet using Keras and Adam optimizer (instead of SGD in the original paper) yields an average accuracy of 94.9%±1.54%, which is already a significant improvement. We performed the same 5-fold validation using a modified version of AudioNet where the first convolutional layer is replaced by a LF layer. This layer consists in 32 4th-order Gammatone filters of length 80 (corresponding to 10 ms at 8 kHz). The stride has been set such that the overlap between two consecutive convolution steps is equal to 75%. In this modified network, the number of trainable parameters drops to ca. 3.5 million trainable parameters, i.e. a reduction in size by a factor 5. Using the LF-enabled AudioNet the average accuracy increases to $96.8\% \pm 1.22\%$.

Another LF-enabled network was used to perform the classification task on AudioMNIST. The architecture is derived from the raw waveform model *SampleCNN* introduced in [13]. Despite its much smaller number of trainable parameters (ca. 300'000), its average accuracy improves to $98.0\% \pm 0.41\%$. For the sake of completeness, we also trained this network, replacing the Gammatone filters by the learned wavelets as in [12], and the learned SincNet filters from [27]. A summary of all results achieved using AudioMNIST can be found in Table 1.

3.2 Google Speech Command

The Google Speech Command dataset [32] provides similar data to the AudioMNIST one, with a larger number of classes (35). As done in [34] and its ac-

Table 1: AudioMNIST mean test accuracy

Network	# Trainable parameters	Avg. accuracy
AudioNet	17 M	$94.9\% \pm 1.54\%$
LF-AudioNet	3.5 M	$96.8\% \pm 1.22\%$
LF-custom (Gammatone)	**300 k**	$\mathbf{98.0\% \pm 0.41\%}$
LF-custom (SincNet)	300 k	$97.2\% \pm 1.0\%$
LF-custom (Wavelet)	300 k	$89.9\% \pm 1.18\%$

companying code, we used the pre-defined dataset from Tensorflow which reduces the number of labels to 12, by merging a number of samples into an *unknown* class.

Given that Google Speech Commands does not possess pre-defined folds for n-fold validation, the experiments were repeated 3 times in order to compute the mean accuracy. In [34], the authors train a learnable parametric frontend similar to the one introduced in this paper. Their framework, called "LEAF", consists in a frontend, a convolutional network, and a final layer adapted to the number of classes in the dataset. The frontend is made of a learnable Gabor filter bank, a learnable pooling function, and a learnable smooth compression function. We reproduced the experimental setting from [34], using a frontend made of 40 order 4 Gammatone filters, overlapping by 80% and having a length representing 25 ms. In one experiment, we did not use the learnable pooling and compression methods present in LEAF and in the other we did use the complete LEAF pipeline. The convolutional network, based on EfficientNet-B0 [31], had been trained using the Adam optimizer during 30 epochs with batches of 128 and 256 samples, and using learning rate reduction on plateaus. The resulting network has ca. 3.5 million trainable parameters. The test accuracy from [34] using the complete LEAF model with Gabor filters is $\mathbf{93.4\% \pm 0.3}$. In our experiments, we observed that using Gammatones over the complete LEAF pipeline lead to results very close to the ones achieved with Gabor filters, i.e. ca. 93% of test accuracy. Using the simpler version without learnable pooling and compression, test accuracy improves to $\mathbf{94.31\% \pm 0.1}$, when using batches of 128 samples.

3.3 Properties of learned filters

The learned parameters of the LF filters can reveal insights about the data and the learning process. As stated in the introduction, several studies have shown a tendency governing the spacing in frequency of their learned kernels. The spacing becomes exponentially large as the frequency increases, following what is called a Mel scale [21]. This is in agreement with psycho-acoustics tests on the human cochlear system. In order to go further in this direction, we investigate 1) the frequency spacing and 2) we test the relationship between the temporal width of the filters and their central frequency. Indeed, psycho-acoustic models (the equivalent rectangular bandwidth (ERB) model [9] and the Bark model [35]) provide such a relationship. This is made possible by our approach where the temporal width as well as the filter central frequency are well defined for each filter.

Bandwidth and frequency. The learned filter banks can be compared to filter banks modeling the human auditory system. Two main models can be found in the literature, the Equivalent Rectangular Bandwidth (ERB) model [9], and the Bark model [35]. The ERB and Bark curves are plotted on Fig. 2, together with the learned parameters of the Gammatone filters initialized with different orders, and the Gabor filters. All the filters have been trained using the LEAF network and the Google Speech dataset, used in section 3.2. We observe a good agreement between the ERB curve and the learned Gammatone filters of order 4 and 6. The agreement is even stronger below 2 kHz. Gammatone filters of order 2 and Gabor filters do not exhibit this behavior and do not follow neither the ERB, nor the Bark curves, while keeping a similar test accuracy on the Google speech commands

dataset. In [27], the authors show that for a neural network applied to a speech dataset, the focus of the learning is situated around the pitch frequency located at 130Hz (male) and 230Hz (female), and the first and second formants (i.e. resonances of the vocal tract [20]), which are around 500Hz and 1kHz respectively. This is exactly the frequency region where our learned filters match the ERB scale.

Frequency spacing. To show the importance of the frequency spacing, we initialized the LF layer with a linear frequency spacing from 0 to the Nyquist frequency. After the learning phase, the filter frequencies evolved and moved away from their initial value as can be seen on Fig. 3. The frequency distribution is not exponential (as in the case of the Mel scale) but we can point out several interesting facts. Firstly, the final curve is flatter than the initialization in the range 0-2kHz (more filters in this range). It shows that the network tends to favour filters with a band-pass in this range for its discriminative process. Secondly, beyond 4kHz, the filters stay close to their original value. This suggests that there is not enough meaningful information in this frequency range for a correct learning. This is indeed the case for speech dataset where we found that the main information resides below 4kHz.

Figure 3: Frequency distribution of the filters before (straight line) and after training (green curve)

4 Conclusion

Decades of research in audio signal processing have brought us an important knowledge about sounds, speech and audio information. This knowledge may be inserted within neural networks as a priori information and turned into efficient inductive biases. This is what we show with the example of the LF layer, a layer of parameterized filters adapted to the extraction of audio information. Moreover, the trained network possesses properties than can, in turn, bring new insights about audio data back to the audio signal processing community. For example, the optimal relationship between frequency and bandwidth seems to be influenced by the envelop shape in a non-trivial manner.

Future work in this direction and further developments of convolutions with parameterized functions may lead to important progress both in deep learning and audio signal processing. The reduction of the number of trainable parameters decreases the network complexity, along with the training time. It also enables a better interpretation of the network adaptation to the data.

References

[1] J. Andén and S. Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014. doi:10.1109/TSP.2014.2326991.

[2] S. Becker, M. Ackermann, S. Lapuschkin, K. Müller, and W. Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018. URL `http://arxiv.org/abs/1807.03418`.

[3] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013. doi:10.1109/TPAMI.2012.230.

[4] P. Chandna, M. Miron, J. Janer, and E. Gómez. Monoaural audio source separation using deep convolutional neural networks. In *International conference on latent variable analysis and signal separation*, pages 258–266. Springer, 2017. doi:10.1007/978-3-319-53547-0_25.

[5] K. Choi, G. Fazekas, K. Cho, and M. Sandler. A tutorial on deep learning for music information retrieval. *arXiv preprint arXiv:1709.04396*, 2017.

[6] F. Cotter and N. G. Kingsbury. A learnable Scatternet: Locally invariant convolutional layers. *2019 IEEE International Conference on Image Processing (ICIP)*, pages 350–354, 2019. doi:10.1109/ICIP.2019.8802977.

[7] A. Darling. Properties and implementation of the gammatone filter: a tutorial. *Speech Hearing and Language, Work in Progress, University College London, Department of Phonetics and Linguistics*, pages 43–61, 1991.

[8] D. Ditter and T. Gerkmann. A multi-phase gammatone filterbank for speech separation via TasNet. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 36–40. IEEE, 2020. doi:10.1109/ICASSP40776.2020.9053602.

[9] B. R. Glasberg and B. C. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing research*, 47(1-2):103–138, 1990. doi:10.1016/0378-5955(90)90170-T.

[10] V. Hohmann. Frequency analysis and synthesis using a gammatone filterbank. *Acta Acustica united with Acustica*, 88(3):433–442, 2002.

[11] J.-H. Jacobsen, J. van Gemert, Z. Lou, and A. W. Smeulders. Structured receptive fields in CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2610–2619, 2016. doi:10.1109/CVPR.2016.286.

[12] H. Khan and B. Yener. Learning filter widths of spectral decompositions with wavelets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4601–4612. Curran Associates, Inc., 2018. doi:10.5555/3327345.3327371.

[13] T. Kim, J. Lee, and J. Nam. Comparison and analysis of SampleCNN architectures for audio classification. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):285–297, May 2019. doi:10.1109/JSTSP.2019.2909479.

[14] E. Loweimi, P. Bell, and S. Renals. On learning interpretable CNNs with parametric modulated kernel-based filters. In *Proc. Interspeech*, pages 3480–3484, 2019. doi:10.21437/Interspeech.2019-1257.

[15] X. Lu, Y. Tsao, S. Matsuda, and C. Hori. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, pages 436–440, 2013. doi:10.21437/Interspeech.2013-130.

[16] Y. Luo and N. Mesgarani. TasNet: time-domain audio separation network for real-time, single-channel speech separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 696–700. IEEE, 2018. doi:10.1109/ICASSP.2018.8462116.

[17] Y. Luo and N. Mesgarani. Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266, 2019. doi:10.1109/TASLP.2019.2915167.

[18] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[19] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*, 2017.

[20] D. O'Shaughnessy. *Speech Communications: Human and Machine*, chapter 3, page 45. Wiley-IEEE Press, 2000. doi:10.1109/9780470546475.

[21] D. O'Shaughnessy. *Speech Communications: Human and Machine*, chapter 4, pages 127–128. Wiley-IEEE Press, 2000. doi:10.1109/9780470546475.

[22] T. L. Paine, P. Khorrami, S. Chang, Y. Zhang, P. Ramachandran, M. A. Hasegawa-Johnson, and T. S. Huang. Fast wavenet generation

algorithm. *arXiv preprint arXiv:1611.09482*, 2016.

[23] R. D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice. An efficient auditory filterbank based on the gammatone function. In *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, volume 2, 1987.

[24] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand. Complex sounds and auditory images. In *Auditory physiology and perception*, pages 429–446. Elsevier, 1992. doi:10.1016/B978-0-08-041847-6.50054-X.

[25] K. J. Piczak. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2015. doi:10.1109/MLSP.2015.7324337.

[26] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019. doi:10.1109/JSTSP.2019.2908700.

[27] M. Ravanelli and Y. Bengio. Speaker recognition from raw waveform with SincNet. *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, 2018. doi:10.1109/SLT.2018.8639585.

[28] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017. doi:10.1109/LSP.2017.2657381.

[29] H. Seki, K. Yamamoto, and S. Nakagawa. A deep neural network integrated with filterbank learning for speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5480–5484, March 2017. doi:10.1109/ICASSP.2017.7953204.

[30] D. Stoller, S. Ewert, and S. Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185*, 2018. doi:10.5281/ZENODO.1492417.

[31] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/tan19a.html`.

[32] P. Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *CoRR*, abs/1804.03209, 2018. URL `http://arxiv.org/abs/1804.03209`.

[33] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schaiz, G. Synnaeve, and E. Dupoux. Learning filterbanks from raw speech for phone recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5509–5513. IEEE, 2018.

[34] N. Zeghidour, O. Teboul, F. de Chaumont Quitry, and M. Tagliasacchi. LEAF: A learnable frontend for audio classification. *ICLR*, 2021.

[35] E. Zwicker and E. Terhardt. Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *The Journal of the Acoustical Society of America*, 68 (5):1523–1525, 1980. doi:10.1121/1.385079.

BIOCYBERNETICS AND BIOMEDICAL ENGINEERING 41 (2021) 1629–1645

Available at www.sciencedirect.com

ScienceDirect

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

Original Research Article

Compact convolutional neural network (CNN) based on SincNet for end-to-end motor imagery decoding and analysis

Tarmizi Ahmad Izzuddin [a,b,*], *Norlaili Mat Safri* [b], *Mohd Afzan Othman* [b]

[a] *Department of Control, Instrumentation and Automation, Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia*

[b] *Department of Electronic and Computer Engineering, School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor, Malaysia*

ARTICLE INFO

Article history:
Received 9 July 2021
Received in revised form
13 September 2021
Accepted 1 October 2021
Available online 13 October 2021

Keywords:
Brain-Computer Interface (BCI)
Convolutional Neural Network (CNN)
Electroencephalogram (EEG)
Motor imagery

ABSTRACT

In the field of human-computer interaction, the detection, extraction and classification of the electroencephalogram (EEG) spectral and spatial features are crucial towards developing a practical and robust non-invasive EEG-based brain-computer interface. Recently, due to the popularity of end-to-end deep learning, the applicability of algorithms such as convolutional neural networks (CNN) has been explored to achieve the mentioned tasks. This paper presents an improved and compact CNN algorithm for motor imagery decoding based on the adaptation of SincNet, which was initially developed for speaker recognition task from the raw audio input. Such adaptation allows for a compact end-to-end neural network with state-of-the-art (SOTA) performances and enables network interpretability for neurophysiological validation in cortical rhythms and spatial analysis. In order to validate the performance of proposed algorithms, two datasets were used; the first is the publicly available BCI Competition IV dataset 2a, which was often used as a benchmark in validating motor imagery classification algorithms, and the second is a dataset consists of primary data initially collected to study the difference between motor imagery and mental-task associated motor imagery BCI and was used to test the plausibility of the proposed algorithm in highlighting the differences in terms of cortical rhythms. Competitive decoding performance was achieved in both datasets in comparisons with SOTA CNN models, albeit with the lowest number of trainable parameters. In addition, it was shown that the proposed architecture performs a cleaner band-pass, highlighting the necessary frequency bands that were crucial and neurophysiologically plausible in solving the classification tasks.

* Corresponding author at: Department of Control, Instrumentation and Automation, Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.
E-mail address: tarmizi@utem.edu.my (T.A. Izzuddin).

https://doi.org/10.1016/j.bbe.2021.10.001

1. Introduction

A Brain-Computer Interface (BCI) can be defined as a system that translates brain activity patterns into messages or commands that represent the user's intention or condition by using a direct brain-to-computer mode of communication [1–3]. There are multiple methods of measuring brain activity ranging from the non-invasive method of measuring Electroencephalography (EEG) signals to a more invasive method called Electrocorticography (ECoG) that places a thin layer of electrodes directly on top of the exposed layer of the brain [4]. In both methods, one particular means of enabling BCI is through the detection of the brain's Motor-imagery (MI) signals, signals that arise due to the synchronization/desynchronization of specific frequency bands during the imagined movement of different body parts [5]. Such BCI systems are often called MI-based BCI or MI-BCI for short.

1.1. *Related works*

Over the past years, numerous decoding techniques and approaches have been developed to achieve good decoding performance from EEG signals. Generally, decoding MI signals follow a particular pipeline, consisting of filtering the raw EEG signals, extracting features from the filtered signals, and classifying them using suitable classifiers. Thus, the MI decoding can be categorized depending on the types of extracted features used and the classification's approach.

One competition-winning approach of decoding MI signal is by obtaining the Common Spatial Pattern (CSP) features to be classified with a classifier such as linear discriminant analysis (LDA), support vector machine (SVM) or similar linear classifier. In this form, a combination of band-pass with spatial filtering is used to emphasizes the MI differences of different body parts. Spatial filtering is commonly achieved using CSP algorithms to maximize an MI class variance while minimising the other [6–8]. One popular and competition-winning variant of CSP is the Filter Bank Common Spatial Pattern (FBCSP) algorithm in which CSP is used in conjunction with a bank of band-pass filters [9–10] and L1-Norm based features selection method [11] in order to obtain the optimal sets of spatial filters in multiple frequency bands. A more recent method uses the Spiking Neural Network (SNN) [12] to classify features extracted from FBCSP, enhancing multiple classes MI classification.

Apart from CSP, a more state-of-the-art approach is through the use of Riemannian Geometry based classifier. The basic concept of a Riemannian Geometry Classifier (RGC) is to map the data directly to a geometric space fitted with an appropriate metric. In this geometric space, manipulating interpolation and classification data can be performed much easily compared to the non-geometric space. The earliest study that proposed the use of this approach is detailed in [13] in which a Riemmanian Distance To Mean (RDTM) classifier was used to classify band-pass covariance features for MI decoding. Apart from RDTM, a more accurate Riemannian geometric approach is highlighted in [14] in which an SVM Riemannian kernel is proposed for classification.

However, decoding the MI signal using previously mentioned techniques requires separately tuning and optimizing the decoding pipeline's modulation, filtering, and classification stage. This process often requires a priori or subject-matter expert about the expected signal outcome. To alleviate this matter, many research studies have explored the use of *Deep Learning* (also known as deep neural networks) towards end-to-end decoding of MI signals. The use of deep Convolutional Neural Networks (CNN), for example, has become highly successful in many applications such as in computer vision and speech recognition, often outperforming conventional engineered methods. Using CNN with many layers, researchers managed to reduce the error rates on the ImageNet image recognition challenge, where 1.2 million images must be classified into 1000 different classes, from 26% to just below 4% in 4 years [15]. Deep CNN also contributes to the success of reducing the error rate on speech recognition and enables the development of current mobile speech recognition technology [16].

Owing to the great success in these fields, the applicability of using deep CNN for EEG-based MI signal classification has been explored by researchers [17–20]. CNN enables end-to-end learning, which is learning directly from raw EEG signals without a priori collection of features, scales to large datasets, and takes the hierarchy of natural signals (learning from simple concepts in its early layer into complex ones in its last layer), unlike the traditional processing and classification system.

One such example is a work reported in [17], in which the authors exhaustively explored the feasibility of using both deep and shallow CNN for EEG motor imagery signal decoding and reported excellent performance even when compared with the conventional FBCSP method. In [21], the authors introduced a compact CNN for BCI purposes called EEGNet, which can be applied to multiple BCI paradigms. Unlike standard methods, which are often tailored to specific BCI paradigms, EEGNet can be applied towards multiple classification tasks (here the author listed four: P300 classification, ERD/ERS, movement-related cortical potential - MRCP, and sensory-motor rhythms -SMR) without changing the network architecture.

A more MI-specific CNN architecture based on EEGNeT was later proposed by authors in [22], which uses the temporal convolutional layer from EEGNeT and spatial feature extraction convolutional layer and activation function from CSP-

Table 1 – Total number of CNN parameters architectures used in this study.

	Shallow ConvNet	EEGNet	TA-CSPNN	Sinc-EEGNet	Sinc-CSPNN
No. of parameters	40,566	1876	978	1380	644

NN. The proposed architecture (called Temporally Adaptive Common Spatial Pattern, TA-CSPNN) uses half the parameters from EEGNet but retains similar accuracy. Another study that explores the use of CNN for MI-based EEG classification is given in [23], in which the 5-layer CNN model is built to classify MI tasks (left- and right-hand movement). Here it was reported that using CNN improved classification accuracy over conventional methods such as SVM and CSP.

1.2. *Proposed method*

This paper introduces a much compact and improved EEG-based MI decoding architecture based on EEGNeT and TA-CSPNN but proposes using a parametrized Sinc-based convolution network using SincNet [24] on the first CNN layer. Initially proposed for the speaker recognition task from raw audio waveform, SincNet allows the first convolutional layer to act as a differentiable band-pass filter and co-joint with standard NN architecture. This is achieved via the convolution of a parameterized Sinc function with the input signals.

Although it was shown that a standard CNN could be trained to act as a Finite-Impulse-Response (FIR) filter on EEG signals [17,21–22,25], the application of SincNet on the first convolutional layer allows the implementation of band-pass filters with a fewer number of high-level tuneable parameters. SincNet thus emphasizes the network on generating band-pass filters that have a better impact on the shape and bandwidth and allowing for better frequency band interpretability. Such architecture was, in part, motivated by the original FBCSP algorithm in which a bank of band-pass filters was used in the early stage of the decoding pipeline.

However, unlike FBCSP, the proposed architecture allows auto-optimization of these band-pass filters during network training since the SincNet layer parameters are differentiable and can be jointed with other CNN layers. This can also be viewed as a form of Differentiable Digital Signal Processing (DDSP) in which the deep learning method is integrated with classical signal processing elements [26]. Such an approach benefits from the inductive bias of using proven signal processing methods while retaining NN's expressive power and end-to-end learning.

2. Methodology

2.1. *SincNet adaptation*

In a standard CNN architecture, time-domain convolution is performed between the input waveform and a set of learned kernels. In the case of using 1-D time-series data such as EEG signal as an input, this kernel act as a Finite Impulse Response (FIR) filter. The original convolution process is defined as follows:

$$y[n] = (x * h)[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l] \tag{1}$$

here $y[n]$ is the filtered output, $x[n]$ is the input signal and $h[n]$ is the kernel of length L. Since an EEG data with C number of channels with trial length T can be viewed as a 2D input, we proposed the following 2D convolution by modifying Eq. (1):

$$y_{c,j}[n] = (x_c * h_j)[n] = \sum_{l=0}^{L-1} x_c[l] \cdot h_j[n-l] \tag{2}$$

Where $c \in C$ and $j \in F_K$. Here hyperparameter F_K denotes the total number of band-pass filters to be used with CNN. Generally, decoding EEG signals using CNN architecture end-to-end enforces optimization of first layer kernel (which generally act as a band-pass filter) during CNN training [17,21,23], with all L elements of the filter are learned during this process. On the other hand, SincNet [24] proposed the use of a predefined Sinc function for a kernel that depends on few learnable parameters. This is motivated by standard filtering in digital signal processing, in which the idea is to design a function that acts as a rectangular band-pass filter. Since in the frequency domain, a band-pass filter can be viewed as a difference between two low-pass filters, the proposed kernel of equation (2) can therefore be viewed in frequency domain as:

$$H_j[f, f_1, f_2] = rect\left(\frac{f}{2f_2}\right) - rect\left(\frac{f}{2f_1}\right) \tag{3}$$

where f_1 and f_2 are the low and high cut-off frequencies respectively, and $rect(.)$ is the rectangular function. The Inverse Fourier Transform (IFT) of this equation then becomes:

$$\begin{aligned} h_j[n, f_1, f_2] &= 2f_2 \frac{\sin(2\pi f_2 n)}{2\pi f_2 n} - 2f_1 \frac{\sin(2\pi f_1 n)}{2\pi f_1 n} \\ &= 2f_2 \mathrm{sinc}(2\pi f_2 n) - 2f_1 \mathrm{sinc}(2\pi f_1 n) \end{aligned} \tag{4}$$

Hence, instead of learning a L sized kernel during CNN training of, the proposed method allows learning of only two parameters, f_1 and f_2, which defines the kernel structure for a convolution process in the time domain. Since equation (4) shows that the band-pass filter can be constructed using a set of differentiable Sinc functions, then parameters f_1 and f_2 can be jointly optimized with other CNN parameters using Stochastic Gradient Descent (SGD) or any gradient-based optimization method. Formally, given $\mathscr{L}$ as the loss-function of the CNN, then using chain-rule, gradient calculation to parameters f_1 and f_2 is possible:

$$\frac{\partial \mathscr{L}}{\partial [f_1, f_2]} = \frac{\partial \mathscr{L}}{\partial h} \cdot \frac{\partial h}{\partial [f_1, f_2]} \tag{5}$$

where $\frac{\partial \mathscr{L}}{\partial [f_1 f_2]}$ is the partial derivative of $\mathscr{L}$ with respect to f_1 and f_2. Moreover, an update to the parameters is obtained by:

$$[f_1, f_2]' = [f_1, f_2] + \eta \frac{\partial \mathscr{L}}{\partial [f_1, f_2]} \tag{6}$$

here η is the learning rate used during CNN training. To ensure that update towards f_1 and f_2 follow $f_1 \geq 0$ and $f_2 \geq f_1$ during training, as suggested by the original SincNet authors, these parameters are fed by the following parameters:

$$f_1^{abs} = |f_1| \tag{7}$$

$$f_2^{abs} = f_1 + f_{band} \tag{8}$$

where $f_{band} = |f_2 - f_1|$ denotes the filter's band size. Consequently, only f_1^{abs} and f_{band} are the only parameters updated

during network training. Again, per the original SincNet authors' suggestion, this convolutional filter is windowed using the popular Hamming windows to mitigate the issue of filter truncation. Given $w[n]$ as the window function, proposed windowed filter $h_{j-windowed}[n,f_1,f_2]$ then becomes:

$$h_{j-windowed}[n,f_1,f_2] = h_j[n,f_1,f_2] \cdot w[n] \quad (9)$$

Where:

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{L}\right) \quad (8)$$

Reasoning from this fact, the construction of a filter-bank is possible using proposed Sinc-based band-pass filters. Although the use of filter-bank is inspired by a previous competitive algorithm such as the FBCSP, the proposed filter-bank can be viewed as a form of adaptive filter-bank that automatically adapts the low cut-off frequency and filter's bandwidth in order to minimize classification error, hence, improving model accuracy. Therefore, from the perspective of using CNN for EEG decoding, replacing the first layer with a Sinc-based convolutional filter allows adaptive filtering of particular EEG wave pertaining to the decoding task with fewer parameters.

Apart from this, since the proposed filter is fully differentiable, all standard CNN pipelines (such as pooling, dropout, and normalisation) can be employed together with the proposed filter. Standard convolutional or fully connected CNN layers can be used stacked together to perform EEG classification. In this study, two compact CNN motor image decoding architectures, the EEGNet (code adapted from https://github.com/vlawhern/arl-eegmodels) and the TA-CSPNN (code adapted https://github.com/mahtamsv/TA-CSPNN), were adapted to the proposed SincNet layer, i.e., the original first layer was removed and replaced with a SincNet layer (Hence, we named it as Sinc-EEGNet and Sinc-CSPNN, respectively) such as shown in Fig. 1.

In both architectures, a depthwise convolutional layer was employed after the SincNet layer to generate the frequency-specific spatial-filters F_s, for the network [17,21,22]. The weights and neurons in this layer are arranged in such a way as to mimics the spatial filters in motor imagery decoding using the CSP algorithm [6,9], which helps in discriminating between EEG signals belonging to a particular task. It was then constrained using the norm $\|w\|_2 \leq 1$ since spatial filters in the CSP algorithm are eigenvectors with a norm of 1.

Similar to the original EEGNet, in the proposed Sinc-EEGNet, a separable convolution layer (which consists of a depthwise convolution followed by pointwise convolutions) was employed to decouple the relationship within and across feature maps by summarising each feature map then merging it at the output. However, such a convolution layer was not employed in TA-CSPNN and thus was not used in our Sinc-CSPNN.

Another difference between both architectures is that the CSPNN uses a squared activation function instead of an exponential linear unit (ELU) since it was claimed that ERS/ERD features are variations in the power of EEG signal. However, this study found that the squared activation function's use

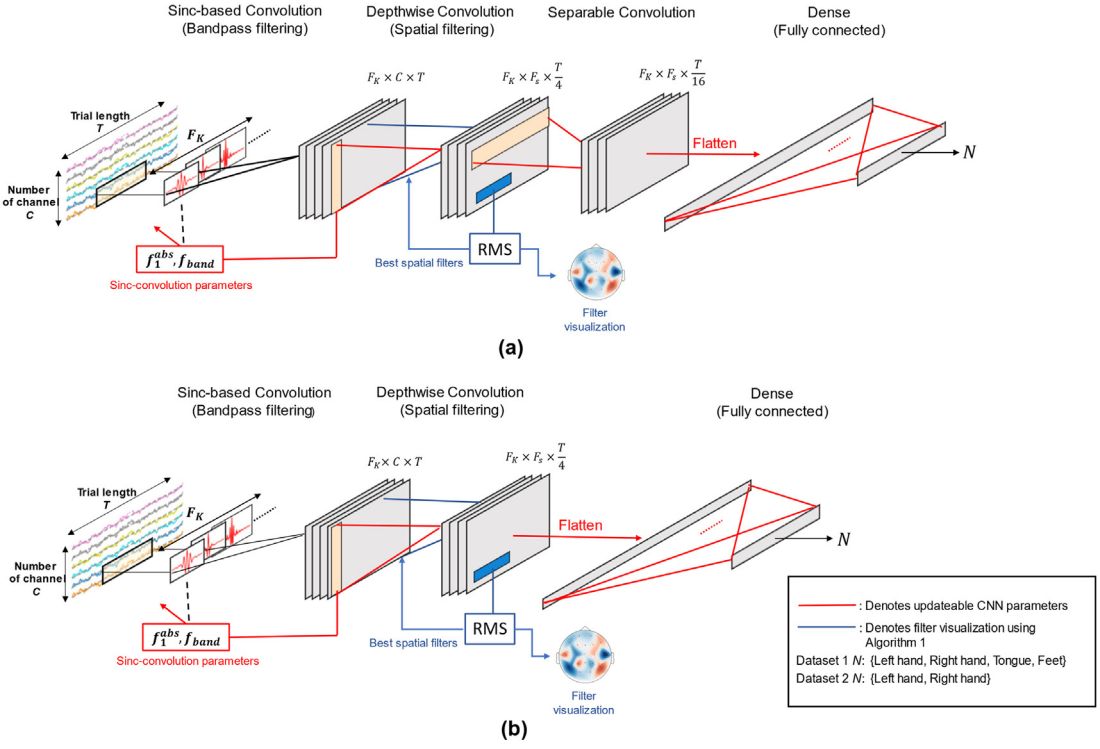


Fig. 1 – Proposed SincNet adapted CNN architecture: (a) Sinc-EEGNet; (b) Sinc-CSPNN.

causes the network not to learn during training. A simple experiment was conducted to replace the activation function by comparing the performance of the widely used ELU and rectified linear unit (RELU) for replacement, and both activation functions were found to alleviate the problem mentioned above. We then chose RELU for the Sinc-CSPNN's activation function.

The final layer of both Sinc-EEGNet and Sinc-CSPNN consists of dense NN connected to a softmax classifier with an output N which aggregates the features generated from the previous layer. Softmax is generally chosen to reduce the free parameters, such as shown in [27]. A detailed structure of the proposed architecture, along with its hyperparameters, are listed in Table A1 in the appendix.

2.2. *Feature visualization and interpretation*

Since CNN-based EEG decoding performs end-to-end learning on raw EEG data, visualization of the learned CNN filters is necessary to validate and understand how proposed Sinc-based architectures distinguish between tasks. The development of CNN visualization has been an important part of the research at this time. As an essential component of the validation process, it was proposed to ensure that the relevant features drive classification performance rather than random noise or artifacts in EEG data [17,21].

Thus, due to the structure of the proposed Sinc-based CNN, two forms of interpretation and visualization of the network are proposed. The first is made possible by the network's spatial filtering (depthwise convolution) layer, which allows for visualization of the topographic spatial patterns by extracting the trained weights in this layer. The second is the visualization of the frequency band that the network focuses on through the Sinc-convolutions layer. The subsequent section explains each form of proposed visualization and interpretation techniques.

2.2.1. *Task-related spatial pattern extraction*

In the original CSP algorithm, spatial patterns relating to one task can be visualized by obtaining the columns of the inverse projection matrix $\mathbf{W}^{-1}$, where the first and last columns are the most important spatial patterns which explain the greatest variance in one task and the smallest variance in the other [6]. In this study, through depth-wise convolution, such patterns can be obtained by visualizing the trained network's spatial filters.

Here, we introduce an algorithm to obtained task-related spatial patterns by identifying the spatial feature map with the highest activation value and backtracking the corresponding filters connected to this feature map. The highest activation was obtained by calculating the root-mean-square (RMS) on the i^{th} feature map (the i^{th} output of the spatial filters) while inputting the network with EEG trials data pertaining to task to be analyzed using proposed algorithm 1 listed below. Again, here F_K refers to the total number of band-pass while F_s refers to frequency-specific spatial filter. Due to the use of the Sinc convolutional layer and the structure of the proposed network, backtracking and extracting a set of spatial filter weights W_c, which gave the maximum activation of the task-related feature map is possible. The corresponding spatial patterns during task n can be obtained by mapping such weights onto an EEG topographic head plot for cortical activity analysis.

Algorithm 1: Spatial pattern extration

Input	:	i^{th} output of the spatial filters x_i pertaining to input data during task n, where $n \in N$
Output	:	A set of spatial filter weights W_c where $c \in C$
Step 1	:	**For** i in the total number of F_K : **For** j in the total number of F_S :
Step 2	:	Calculate the RMS for x_i feature map: $RMS_{i,j} = \sqrt{\frac{1}{T}\sum (x_{i,j})^2}$ Where T is the trial's length.
Step 3	:	**If** $RMS_{i,j} > RMS_{i-1,j-1}$: Save the index $[i_{max}, j_{max}] \leftarrow [i, j]$
Step 4	:	**For** c in the total number of EEG channels C:
	:	**If** W_c is connected to $[i_{max}, j_{max}]$:
		Return $W_{c-max\_i} \leftarrow W_c$

2.2.2. *Frequency band visualization over sinc-kernels*

Since the first layer of the network performs sinc-based convolution, which acts as a band-pass filter, visualization of its frequency response gives an intuition on the frequency band that the network focuses on to solve the classification task. Such information is crucial to establishing the validity of the proposed CNN architecture with known cortical rhythms such as the sensorimotor rhythm (SMR), a common control signal for oscillatory-based BCI [28]. Hence, for this purpose, we proposed two ways of visualizing the frequency response:

1) Visualization of individual kernel's frequency response, which gave an intuition on the frequency band that the i-th sinc-filter (which in turns connected to [i-th, j-th] spatial filter) focuses on. This was performed by performing discrete Fourier transform (DFT) over the individual Sinc kernel $h[n]$ using the standard DFT equation:

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi j \frac{kn}{N}} \cdot h[n] \tag{9}$$

2) The cumulative frequency response of all learned kernels provided insights on the frequency that the whole network focuses on. This can be obtained by performing by summing the frequency response of all kernels using the equation below:

$$Filter\ Sum = \frac{1}{F_K} \sum_{i=1}^{F_k} \sum_{k=1}^{T'} y_i[k]) \tag{10}$$

here T' denotes the window size, and F_k is the number of filters being used.

2.3. *Datasets*

We perform the validation of the proposed method using two datasets. The first Dataset is the publicly available BCI Com-

petition VI (2a) [29] dataset, often used as a benchmark to gauge classifier performance in decoding EEG-based motor imagery signals. Another dataset consists of our primary data that we initially used to study the performance of motor imagery associated with mental imagery tasks.

2.3.1. *Dataset I*

Initially used for BCI competition, this publicly available Dataset (http://www.bbci.de/competition/iv/#datasets) has been used in numerous studies concerning motor imagery BCI [17,21–22]. Dataset consists of EEG recordings of 9 participants performing left hand, right hand, tongue, and both feet motor imagery in two sessions; training and evaluation. Each session recording consists of 228 trials. Data was recorded using 22 Ag/AgCl electrodes (see Fig. 2 for electrode position) with a sampling frequency of 250 Hz. However, in this study, all data were resampled to 125 Hz following the pre-processing procedure that was described in the original data description. A band-pass filter in the range 4–40 Hz was employed, and data was epoch from 0.5 to 2.5 s (in the MI region) following Dataset's description. All data recorded from training sessions was used to train all classifiers, while data from evaluation sessions were used to evaluate and test (50% each from the total number of data) (Fig. 3).

2.3.2. *Dataset II*

This Dataset consists of EEG recordings from 11 participants (10 males, one female) who all voluntarily agreed to participate and signed a given consent form approved by the Ministry of Health Malaysia. Data were recorded using the medical-grade NVX-52 EEG amplifier from MKS utilizing 19 channels with electrodes (AgCl) placed according to the international 10–20 system. Recorded initially to study differences between the Motor Imagery (MI) and Motor Imagery associated with the mental rotation task (we denote this as MI + MR from now on), this Dataset consists of 8 trials per subject (4 trials MI, 4 trials MI + MR) with each trial lasted 7 s. The use of this Dataset is partly motivated in assessing proposed sinc based CNN as a tool for interpreting and visualizing differences between MI and MI + MR task in terms of features learned by CNN.

During MI tasks, participants were required to perform right and left-hand imagery movement. A short training session was then conducted in which participants were required to perform a virtual 3D object manipulation task by associating hand movements with the rotation of a 3D star-shaped object on a computer screen through the use of an accelerometer attached to the participant's hand. After training, participants were again required to perform a motor imagery task, but this time around while simultaneously mind visualizing the 3D object rotation seen and manipulated during the training session (hence such task was denoted as MI + MR). All EEG recordings were sampled at 500 Hz downsampled to 125 Hz and notch filtered at 60 Hz.

Since the number of trials per subject is fewer, albeit with a longer trial duration when compared to Dataset I, a sliding window strategy similar to [17] was employed to augment the number of trials and provide more training data for the network, in this strategy, "multiple crops" of EEG data are used to increase the EEG decoding accuracy. Formally, given original trial $X^j \in \mathbb{R}^T$ with T as timesteps, a set of crops with window size T' as time slices with hop size h of the trial is given as follows:

$$C^j = \left\{ X^j_{t+h\cdots t+h+T'} | t \in 1..T - T' \right\} \tag{11}$$

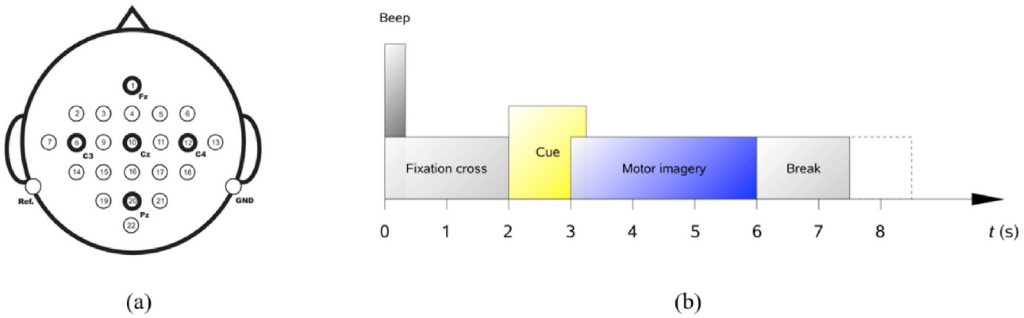
All of the T- T' crops are used as the new training data for our CNN classifier. A window size of 250 (around 1 s) with a hop size of 3 was chosen, yielding a total of 393 crops per trial. This hop size was chosen in such a way to ensure that total crops n fit within the trial length T (hence n must be a natural number) according to the following equation:

$$T' + nh = Tn \in \mathbb{N} \tag{12}$$

3. Results

3.1. *Parameters and architecture setup*

In order to gauge the performance of proposed NN architectures, proposed architectures were compared with their non-sinc counterpart (EEGNeT and TA-CSPNN) as well as the state-of-the-art (SOTA) ShallowConvNet from [17]. In addition, since Dataset I was used in the BCI competition IV 2a, a comparison with the competition-winning FBCSP approach was also made in this study. Same parameters were used for both dataset 1 and 2. Except for the first convolution layer in EEGNeT and TA-CSPNN, all parameters in the layers that follow remain the same with our Sinc-based architectures. Since both recordings were downsampled to 125 Hz, a temporal kernel sized $K = 63$ was chosen. This was chosen to allow the proposed NN to collect information in the 2 Hz and above



(a) (b)

Fig. 2 – (a) Electrode montage used in the dataset I; (b) its timing scheme [27].

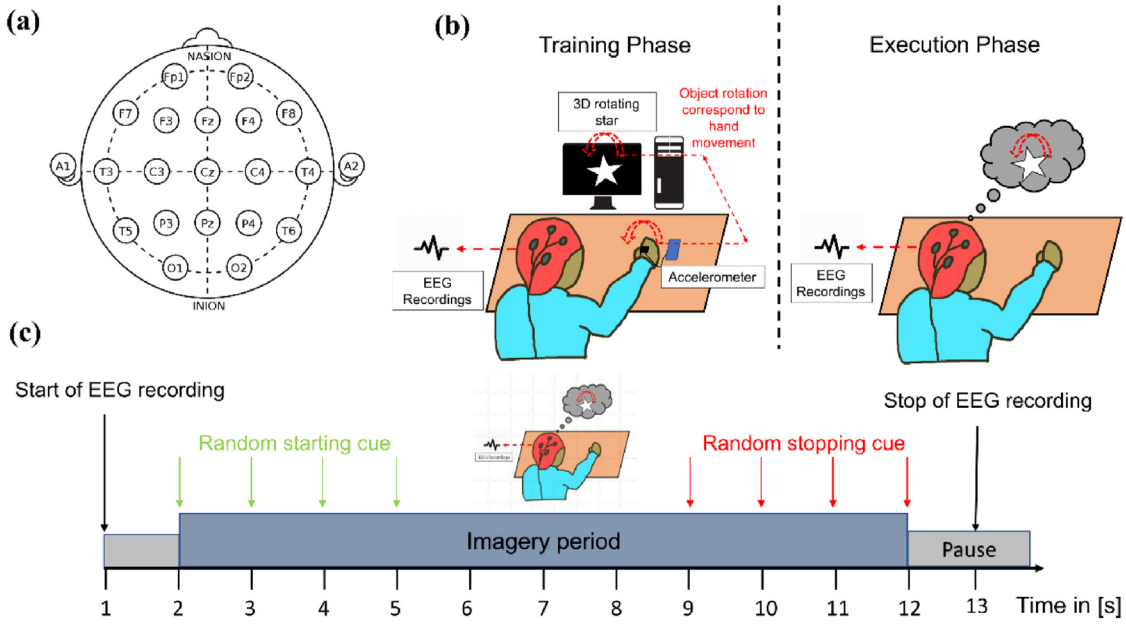


Fig. 3 – Depiction of BCI training protocol and execution phase of Dataset II. (a).

frequency region, as this number is about half the downsampled frequency.

The number of Sinc-based convolution filters F_k were set to 10 with each filter has a band-pass width f_{band} to be at 4 Hz. All band-pass filters are initialised in such a way that they are equally distributed in the range 4 Hz and a maximum of 40 Hz (i.e. starting with [4,8,9,12]… [36,40]. Such is partly inspired by the construction of filter banks constructed in the original FBCSP algorithm. However, it should be noted that since parameters f_{band} and f_1 are differentiable, this act as an adaptive band-pass filter during network training, unlike the static filters in FBCSP. The number of spatial filters F_s was set 2 following the original parameter in EEGNet and TA-CSPNN, while the number of pointwise convolution filters F_p in Sinc-EEGNet is set to 16 (Table A1 in the appendix section shows the overall architecture of the proposed. Sinc-EEGNeT and Sinc-CSPNN).

In both Sinc-convolution and depthwise layers, we employed dropout [30] with a dropout rate of $p = 0.5$, to minimise the effects of overfitting. Table 2 summarizes the total number of trainable parameters in comparisons with its non-sinc counterpart and SOTA shallowConvNet. Training of all CNN models were performed on a standard PC with Intel i5 CPU and 8 GB of RAM, equipped with an Nvidia GTX 1050 GPU with 4 GB of NVRAM. The use of a GPU was necessary in order to increase the training speed. During training, a 10-fold cross-validation method was performed to ensure stability and ensure the model's effectiveness.

3.2. *Classification performance on Dataset I*

All models were trained on the train set for Dataset I with 10% of data used as validation. The models were then tested on the evaluation dataset according to the original BCI Competition rules. Ten different weight's initializations were used for all models. For the FBCSP algorithm, filter-banks and feature extraction techniques were adopted from Kan Kai Ang et al. approach [9–10], and standard Linear Discriminant Analysis was used as the feature's classifier. As Dataset I was initially used for competition, results are reported as Kappa-Cohen coefficient κ calculated using the following equation:

$$\kappa = \frac{p_o - p_e}{1 - p_o} \tag{13}$$

where p_o is the classification accuracy p_e is the proportion of times the MI classes are expected to agree by chance. Such a metric was chosen following the metric used by the competition-winning approach, FBCSP. Table 2 shows the classification accuracy while Table 3 shows the obtained kappa coefficient values and its mean for all classifiers. In addition, Table 4 shows the average performance in terms of precision, recall and ROC's AUC scores for each architecture.

A slight increase in κ was observed on models adapted with Sinc-based convolutions (+0.048 for EEGNeT and +0.034 for CSPNN). A mean kappa of 0.598 was obtained for FBCSP, almost identical to the result reported in [9] (0.599). Our implementation of Shallow ConvNet obtained a slightly better mean than FBCSP, EEGNeT and TA-CSPNN; however, not with the Sinc-convolution adapted version of the latter two (difference of +0.036 with Sinc-EEGNeT and +0.025 with Sinc-CSPNN). From the obtained mean κ values, all architectures are found not to be significantly different from each other ($p > 0.3$).

3.2.1. *Network feature visualization and characterization on dataset I*

For cortical activity analysis, algorithm 1 described in section 2.2 was used to extract the spatial patterns learned by the network for a particular task n ($n \in \{Left\ hand, Right$

Table 2 – Classification performance (accuracy) on Dataset I.

Subject	FBCSP	Shallow ConvNet	EEGNet	TA-CSPNN	Sinc-EEGNet	Sinc-CSPNN
A1	0.74	0.68	0.75	0.76	0.86	0.82
A2	0.47	0.56	0.49	0.47	0.54	0.51
A3	0.76	0.79	0.76	0.78	0.85	0.81
A4	0.55	0.53	0.58	0.49	0.52	0.52
A5	0.61	0.65	0.64	0.63	0.67	0.65
A6	0.51	0.55	0.54	0.56	0.52	0.58
A7	0.69	0.71	0.69	0.61	0.78	0.66
A8	0.69	0.68	0.76	0.78	0.83	0.78
A9	0.69	0.7	0.6	0.66	0.72	0.66
Ave.	0.63	0.65	0.65	0.64	0.70	0.67

* Denotes significant improvement over its non-sinc counterpart (paired T-test, $p < 0.05$)).

Table 3 – Classification performance (Kappa coefficient values κ) on Dataset I.

Subject	FBCSP	Shallow ConvNet	EEGNet	TA-CSPNN	Sinc-EEGNet	Sinc-CSPNN
A1	0.739	0.683	0.716	0.749	**0.793**	0.752
A2	**0.475**	0.430	0.455	0.418	0.408	0.395
A3	0.752	0.759	0.736	0.766	0.761	**0.794**
A4	0.484	**0.512**	0.481	0.485	0.500	0.491
A5	0.601	0.581	0.613	0.592	**0.644**	0.627
A6	0.347	0.438	0.336	0.401	0.316	**0.393**
A7	0.64	0.695	0.680	0.590	**0.705**	0.644
A8	0.682	0.688	0.752	0.744	**0.775**	0.749
A9	0.663	**0.669**	0.578	0.631	0.659	0.641
Mean κ	0.598	0.606	0.594	0.597	**0.642**	**0.631**

Table 4 – Average classification performance of all models II on dataset I.

Model	Precision	Recall	AUC Score
FBCSP	0.640	0.621	0.610
ShallowConvNet	0.678	0.638	0.672
EEGNet	0.642	0.629	0.640
TA-CSPNN	0.646	0.645	0.644
Sinc-EEGNet	0.733	0.729	0.732
Sinc-CSPNN	0.680	0.716	0.691

hand, *Tongue*, *Feet*}). Here, trials from each task were fed to the network, and the spatial filter index [*i_max*,*j_max*], which corresponds to weights that causes the highest neuronal RMS activation on the feature map is extracted. Fig. 4a and 5a show an example of the obtained highest index for each task *n* for subject 3, and its corresponding spatial filters plotted as a topographic head map for Sinc-EEGNet and Sinc-CSPNN, respectively. Subject 3 was chosen as an example as it gave the best accuracy (up to 83% decoding accuracy) among all subjects.

The bar graph located at the upper figure represents the top 4 indexes with a high percentage of activation during the trials of a particular task *n*. Since index *i_max* in [*i_max*, *j_max*] also corresponds to the *i-th* Sinc-based filter which performs band-pass on the EEG data, the frequency response of the *i-th* filter is plotted (as shown in Fig. 4b and 5b to identify the frequency bands that the individual spatial kernel focuses using methods that were previously mentioned in section 2.2.2. In addition, as mentioned in the same section, to identify the frequency band that the whole network focuses on, the cumulative frequency response of all Sinc filters and comparisons with its non-Sinc counterpart is shown in Fig. 6.

3.3. *Classification performance on Dataset II*

In this Dataset, out of 4 trials per task, 3 of the trials were used as training with 10% left for validation. This results in a total of 1179 (393×3) of training data since sliding window strategy was used on each trial. Testing on the proposed model was done on the last trial. Again, similar to the Dataset I, all models were trained using ten sets of weight initialization. Table 3 shows the average test accuracy obtained on all MI and MI + MR tasks for all subjects, while Table 4 shows its performance in terms of its Kappa performance. Similar to the previous Dataset, the average performance in precision, recall, and ROC's AUC scores is shown in Table 5. Fig. 7 summarizes each model's average accuracy to highlight its performance between MI and MI + MR tasks (Table 6).

From Fig. 7, the average accuracy across all models for MI tasks does is almost similar. However, during the MI + MR task, an increase in accuracy can be observed in all models with the most apparent increase observed for sinc adapted CNN architectures. Compared with the MI counterpart, an average of 20% increase is observed with Sinc-EEGNeT and a slightly lower 15% increase for Sinc-CSPNN (Table 7).

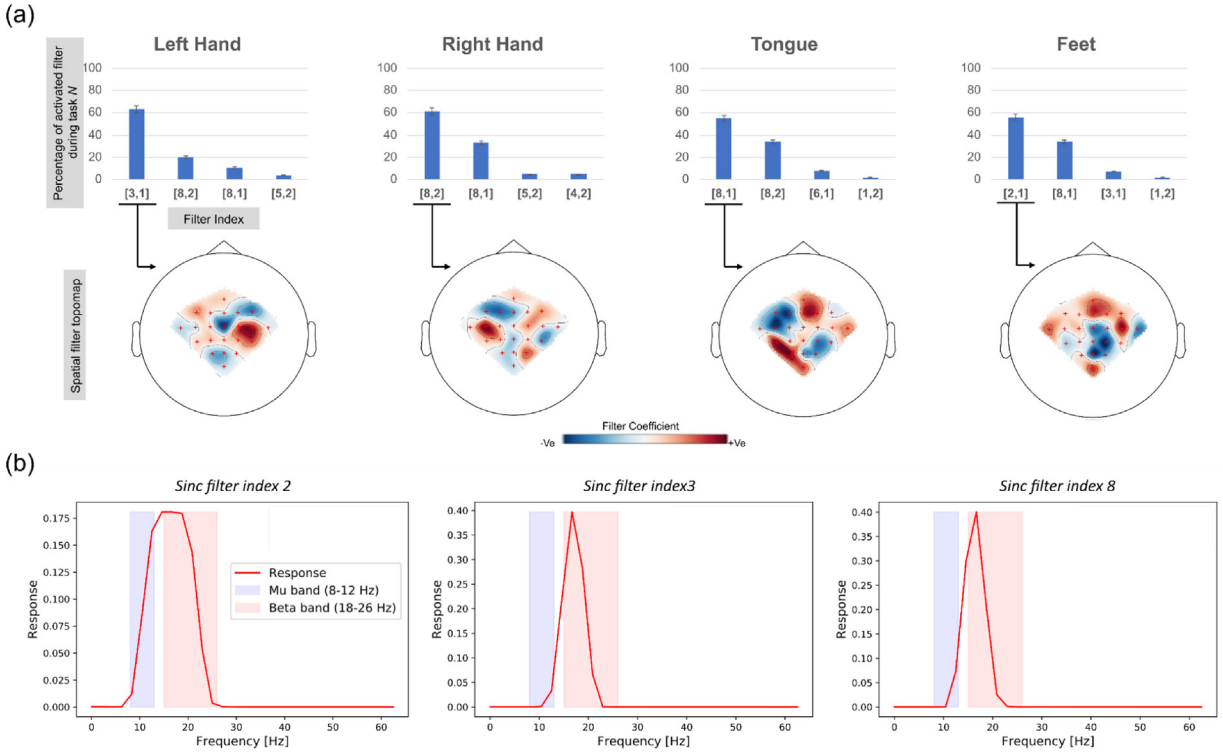


Fig. 4 – Example of subject 3: (a) Percentage of highest obtained index [i_*max*,j_*max*] for particular task N and its corresponding spatial filter obtained using proposed algorithm 1 for Sinc-EEGNet; (b) Index i_*max* individual sincNet filter response.

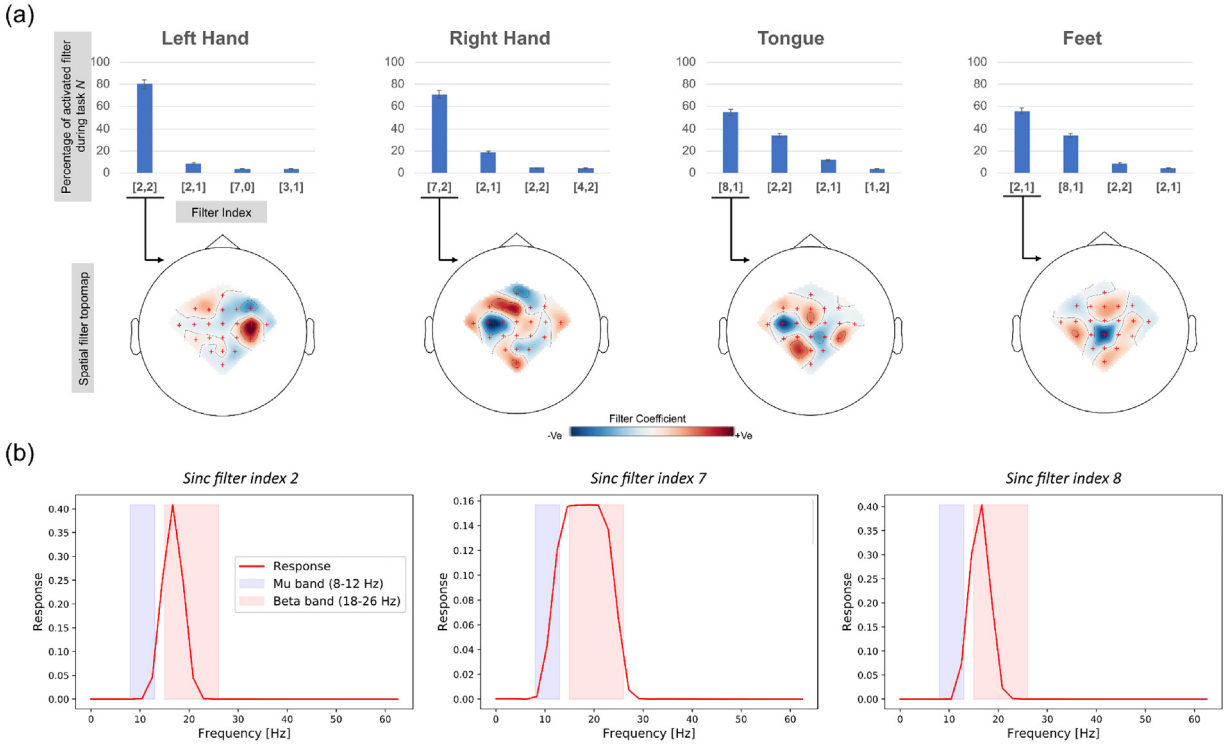


Fig. 5 – Example of subject 3: (a) Percentage of highest obtained index [i_*max*,j_*max*] for particular task N and its corresponding spatial filter obtained using proposed algorithm 1 for Sinc-CSPNN; (b) Index i_*max* individual sincNet filter response.

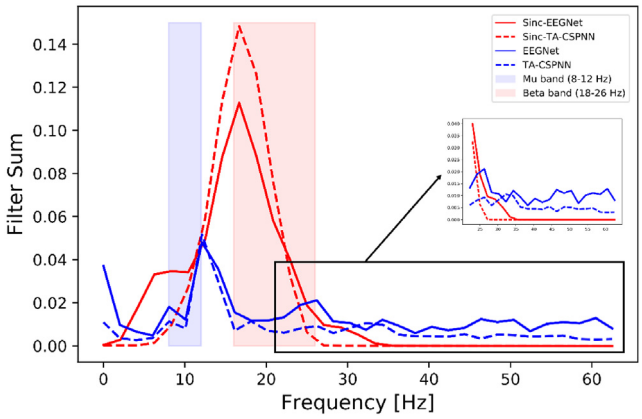


Fig. 6 – Cumulative frequency response of sinc based CNN models versus its non-sinc counterpart.

3.3.1. *Network feature visualization and characterization on Dataset II*

Similar method described in section 3.3 was again used for cortical activity analysis on Dataset II. Unlike Dataset, I, feature visualization and characterization on this Dataset were used to evaluate the proposed network performance in highlighting differences in cortical activity between MI and MI + MR tasks, such as shown in Fig. 8 for subject 12. Since the number on task n is now 2 ($n \in \{Left\ hand, Right\ hand\}$), the highest obtained index for MI and MI + MR task corresponds to the index that gives the highest activation during *Left hand* or *Right-hand* movement. Similar to the method mentioned in section 3.3 again, spatial filters corresponding to the obtained index is plotted as the topographic head map, such as shown in the lower part of Fig. 6, highlighting areas that the proposed network emphasized during the particular task.

Another important characteristic that should be identified in order to differentiate between MI and MI + MR task is the difference in cortical rhythms that appear during the executions of each task. For this purpose, it is necessary to perform frequency analysis on the learned network's sincNet filters to emphasize the cortical frequency band that the network focuses. Such is depicted in Fig. 9, in which the average frequency response of all Sinc filters from all subjects is plotted for both MI and MI + MR tasks.

3.4. *Training performance and characteristics*

For all CNN models, the use of early stopping was adopted. This strategy allows stopping the NN training once the loss on the validation set worsens (i.e., the network's performance stops improving on the validation data) and to prevent overfitting. Fig. 10 shows the average validation-loss curves obtained during training. It can be observed that the validation accuracy improves after each epoch, while the validation loss decreases as the number of epoch increases. This indicates the initialize networks (including the proposed) converge as training progresses. It also can be observed that, due to the early stopping strategy employed, the number of epochs needed for training differs on each architecture.

Since it is assumed that the reduction of parameters led to a reduction in the network's total training time, to verify this, the total number of epochs requires for each architecture (to achieve plateau on validation loss and accuracy) is averaged across all subjects and all folds, such as shown in Table 8 below. From the table, it can be observed that the earlier hypothesis is invalidated as the reduction in the number of parameters does not lead to a reduction in the number of epochs required for the network to achieve steady accuracy. Such is obvious for the case of ShallowConvNet, in which this benchmarking network contains the largest number of trainable parameters but having the least number of epochs required for training. In the case for EEGNeT and CSPNN, however, adopting the proposed sinc convolution layer does in fact cause a slight reduction in the number of epochs required for training, especially in the case of EEGNeT.

3.5. *Ablation: classification of FBCSP's extracted feature using SincNet*

To validate the effectiveness of the adaptive band-pass filtering of the earlier sincNet layer, the sinc convolutional and depthwise convolutional layers of the proposed Sinc-EEGNeT were removed, and the remaining layers were used to classify features extracted from FBCSP algorithm. For this ablation experiment, Sinc-EEGNeT was chosen as it gave the best performance among the proposed sinc based CNN architectures. The sinc convolutional and depthwise convolutional layers were removed as both of these layers mimic FBCSP, albeit being an adaptive one. The resulting architecture consists of only a CNN with a separable convolution layer and a fully connected layer. Unlike the originally proposed Sinc-EEGNeT architecture, output features of the FBCSP algorithm is a matrix with a size of $C \times T$ (because no downsampling was performed) hence, a kernel of size $K = 63$ was used on the separable convolution layer with the number of kernel F_c is maintained at 16. Due to changes in kernel size, the network was re-trained with data from Dataset I and II.

Table 9 below summarizes the obtained average performance metrics when classifying features extracted using FBCSP with the ablated Sinc-EEGNeT. Based on the obtained results, it can be concluded that such classification scheme resulted only in a performance almost similar to the reported classification of using FBCSP with LDA. This outcome is indicative that adapting the sinc convolutional layer with CNN architectures such as EEGNeT or CSPNN enables a better adaptation towards the relevant frequency bands, providing better classification performance.

4. Discussion

4.1. *SOTA performance with a fewer number of parameters*

Based on the obtained results presented in section 3.2 and section 3.3, a slight improvement in decoding accuracy can be observed with the proposed model compared to state-of-the-art (SOTA) models such as ShallowNet and EEGNet. In this case, replacing the first CNN layers with proposed SincNet results in slight improvement with fewer trainable parameters, such as shown in Table 1 27% reduction in the number

Table 5 – Classification performance(percentage) of all CNN models on Dataset II.

	Mental imagery task (MI)					Mental rotation associated motor imagery task (MI+MR)					
Subject	FBCSP	Shallow ConvNet	EEGNet	TA-CSPNN	**Sinc-EEGNet**	**Sinc-CSPNN**	FBCSP	ShallowConvNet	EEGNet	TA-CSPNN	**Sinc-EEGNet**	**Sinc-CSPNN**
1	0.551	0.554	0.542	0.499	**0.574**	0.535	0.681	0.723	0.751	0.718	**0.789**	0.752
2	0.722	0.787	0.721	0.827	**0.849**	0.846	0.541	0.504	0.577	0.541	0.575	**0.629**
3	0.523	0.452	0.464	0.478	0.459	**0.675**	0.585	0.508	0.643	0.462	**0.815**	0.703
4	0.487	0.472	0.476	0.454	**0.631**	0.564	0.479	0.512	0.678	0.581	0.819	**0.828**
5	**0.710**	0.666	0.585	0.494	0.668	0.496	0.841	0.869	0.907	0.609	**0.909**	0.639
6	0.790	0.828	0.783	0.805	0.826	**0.845**	0.442	0.507	0.469	**0.573**	0.548	0.434
7	**0.482**	0.412	0.293	0.415	0.392	0.367	0.477	0.337	0.532	0.648	**0.663**	0.556
8	0.255	0.156	0.324	0.218	0.334	**0.495**	0.514	0.568	0.522	**0.605**	0.555	0.554
9	0.271	0.222	0.256	0.653	**0.424**	0.401	0.471	0.542	0.388	0.482	0.615	0.534
10	0.491	**0.492**	0.481	0.359	0.327	0.442	0.334	0.276	0.344	**0.352**	0.338	0.272
11	0.214	0.202	0.273	0.278	**0.346**	0.169	0.209	0.237	0.274	0.252	0.507	**0.603**
12	0.470	0.395	0.422	**0.483**	0.357	0.457	0.572	0.656	0.754	0.741	0.816	**0.865**
13	0.491	0.493	0.613	0.492	0.529	**0.549**	0.327	0.346	**0.759**	0.614	0.602	0.687
Mean.	0.477	0.472	0.484	0.497	0.517	0.526	0.498	0.507	0.584	0.552	0.658	0.620

Bold indicates highest recorded kappa for a particular subject.

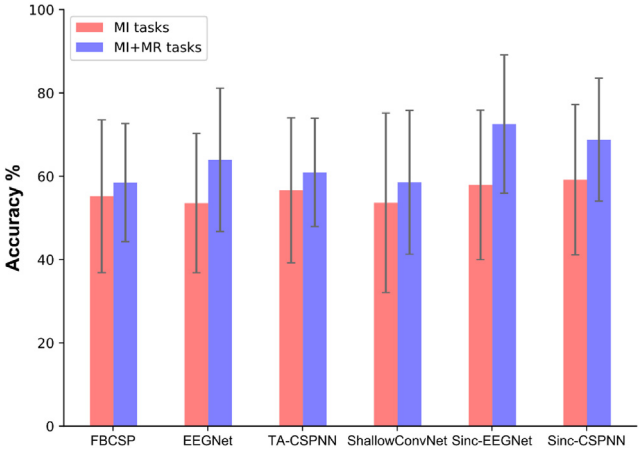


Fig. 7 – Average accuracy on all subjects for both MI and MI + MR tasks.

of trainable parameters compared to SOTA EEGNet, and 35% for TA-CSPNN).

Comparison between the proposed sinc based CNN architecture with the non-NN FBCSP approach on Dataset I yielded a slight improvement in mean kappa value, although such improvement is not statistically significant. Such a slight improvement is also notable in Dataset II classification results. It should be noted that although the proposed approach does not differ much in terms of classification performance, the NN based approach allows for end-to-end classification, that is, direct classification from the raw EEG data bypassing the needs for filtration, feature extraction and classification stage typically described in conventional MI decoding pipeline. This, in turn, eliminates the needs for manually tuning each element in the decoding pipeline as a NN based approach relies on automatically optimizing each network's layer during network training.

Although almost similar MI decoding performance can be observed in Dataset I and II, introducing mental rotation tasks during MI task (MI + MR) results in a general improvement in decoding accuracy in all models. This improvement is expected since it is hypothesized that such task allows the subject to engage in kinaesthetic motor imagery movement needed to ensure good performance while using a BCI system [31,32]. However, the best improvement in decoding accuracy during the MI + MR task can be observed in proposed Sinc based architectures (up to almost 90% accuracy in some subjects) with an average 20% improvement in Sinc-EEGNet and 15% for Sinc-CSPNN. Such performance may be attributed to SincNet's ability to focus and "notch" at a frequency band that only contains relevant information pertaining to a given task.

Besides, such an ability to highlight the necessary frequency band in the first layer of the network allows for the applicability of algorithm 1, which is used to extract learned spatial filters for neurophysiological interpretation and visualization. Another factor that may contribute to better classification in Dataset II is that unlike trial-wise decoding used in Dataset I, a sliding window over trial strategy may allow the CNN models to better adapt to only necessary frequency bands that contain relevant information. Such performance

Table 6 – Classification performance (Kappa coefficient).

	Mental imagery task (MI)						Mental rotation associated motor imagery task (MI+MR)					
Subject	FBCSP	Shallow ConvNet	EEGNet	TA-CSPNN	Sinc-EEGNet	Sinc-CSPNN	FBCSP	Shallow ConvNet	EEGNet	TA-CSPNN	Sinc-EEGNet	Sinc-CSPNN
1	0.551	0.554	0.542	0.499	**0.574**	0.535	0.681	0.723	0.751	0.718	**0.789**	0.752
2	0.722	0.787	0.721	0.827	**0.849**	0.846	0.541	0.504	0.577	0.541	0.575	**0.629**
3	0.523	0.452	0.464	0.478	0.459	**0.675**	0.585	0.508	0.643	0.462	**0.815**	0.703
4	0.487	0.472	0.476	0.454	**0.631**	0.564	0.479	0.512	0.678	0.581	0.819	**0.828**
5	**0.710**	0.666	0.585	0.494	0.668	0.496	0.841	0.869	0.907	0.609	**0.909**	0.639
6	0.790	0.828	0.783	0.805	0.826	**0.845**	0.442	0.507	0.469	**0.573**	0.548	0.434
7	**0.482**	0.412	0.293	0.415	0.392	0.367	0.477	0.337	0.532	0.648	**0.663**	0.556
8	0.255	0.156	0.324	0.218	0.334	**0.495**	0.514	0.568	0.522	**0.605**	0.555	0.554
9	0.271	0.222	0.256	0.653	**0.424**	0.401	0.471	0.542	0.388	0.482	0.615	0.534
10	0.491	**0.492**	0.481	0.359	0.327	0.442	0.334	0.276	0.344	**0.352**	0.338	0.272
11	0.214	0.202	0.273	0.278	**0.346**	0.169	0.209	0.237	0.274	0.252	0.507	**0.603**
12	0.470	0.395	0.422	**0.483**	0.357	0.457	0.572	0.656	0.754	0.741	0.816	**0.865**
13	0.491	0.493	0.613	0.492	0.529	**0.549**	0.327	0.346	**0.759**	0.614	0.602	0.687
Mean.	0.477	0.472	0.484	0.497	0.517	0.526	0.498	0.507	0.584	0.552	0.658	0.620

Bold indicates highest recorded kappa for a particular subject.

was also reported in [33] in which the performance of deep CNN models for motor imagery decoding is generally better for a slice-wise windowed approach compared to a trial-wise decoding approach.

While the performance of both Sinc-EEGNet and Sinc-CSPNN are comparable, it should be noted that the former generally performs better that the latter. Such performance maybe attributed to relatively large size and the addition of extra layer in Sinc-EEGnet thus allowing for more information to be learned by the network. In comparison with its non-sinc counterpart, the use of sinc based convolutions reduces the amount of overfitting, which are prone to occur in a network with larger number of parameters.

4.2. *Neurophysiological interpretability of adapted SincNet and spatial filters*

As mentioned earlier, spectral analysis on SincNet filters allows for neurophysiological interpretation of cortical rhythms that the learned network focuses on distinguishing between tasks. Here, spectral analysis on the Sinc layers was performed using the method described in section 2.2.1. Based on results shown in the lower part of Figs. 4 and 5, spectral analysis on the individual Sinc filters reveals that the adapted filters focus on bandpassing frequency located in the *mu* band (8–13 Hz) and the *beta* band (15–26 Hz). Such bands are known to neurophysiologically related to motor imagery movement, imagination, and visualization [34,35].

Apart from analysing sinc kernel individually, obtaining the cumulative frequency response of proposed architecture and comparing it with its non-sinc counterpart shows that the sincNet layer performed a "cleaner" band-pass with a notch-shaped filter starting at the *mu* band and peaked at the earlier *beta* band region, such as shown in Fig. 6. Such band-highlighting characteristics were also reported in the original SincNet literature [24] in which SincNet successfully adapted its characteristics to address the speaker identification task from the raw voice signal.

This characteristic was again validated in Dataset II in which sincNet was used to analysed differences in cortical rhythm between MI and MI + MR tasks. As shown in Fig. 9 section 3.3.1, Both models show a decrease in the SincNet filter's peak value, which lies in the *mu* band region for the MI + MR task. Another observation is that apart from decreasing, this peak is now shifted towards a higher *mu* region near the *beta* band region. The decreased in the filter's peak value during the MI + MR task is expected since subjects are hypothesized to suppress the mu rhythm better while performing these tasks.

It is interesting to note that apart from clearly highlighting the decrease in peak value in the *mu* band, visualization of SincNet's filter allows for the observation of a slight shift in the band-pass peak's position towards the *beta* region, which often associated with focused mental state, high arousal, and high alertness [36–38]. Such a state is hypothesized to be apparent during mental rotation tasks. In addition, beta waves have been shown to increase during concentration and immersion, particularly in frontal or occipital lobes [39], which is expected during MI + MR task.

Apart from spectral visualization of SincNet filters, this study also proposes algorithm 1 to extract and visualize the

Table 7 – Average classification performance of all models on Dataset II.

Model	MI Task			MI+MR Task		
	Precision	Recall	AUC Score	Precision	Recall	AUC Score
FBCSP	0.542	0.571	0.551	0.593	0.572	0.584
ShallowConvNet	0.533	0.525	0.528	0.677	0.611	0.610
EEGNet	0.601	0.556	0.556	0.606	0.448	0.651
TA-CSPNN	0.545	0.498	0.542	0.651	0.578	0.624
Sinc-EEGNet	0.661	0.694	0.625	0.780	0.766	0.751
Sinc-CSPNN	0.580	0.766	0.601	0.698	0.680	0.693

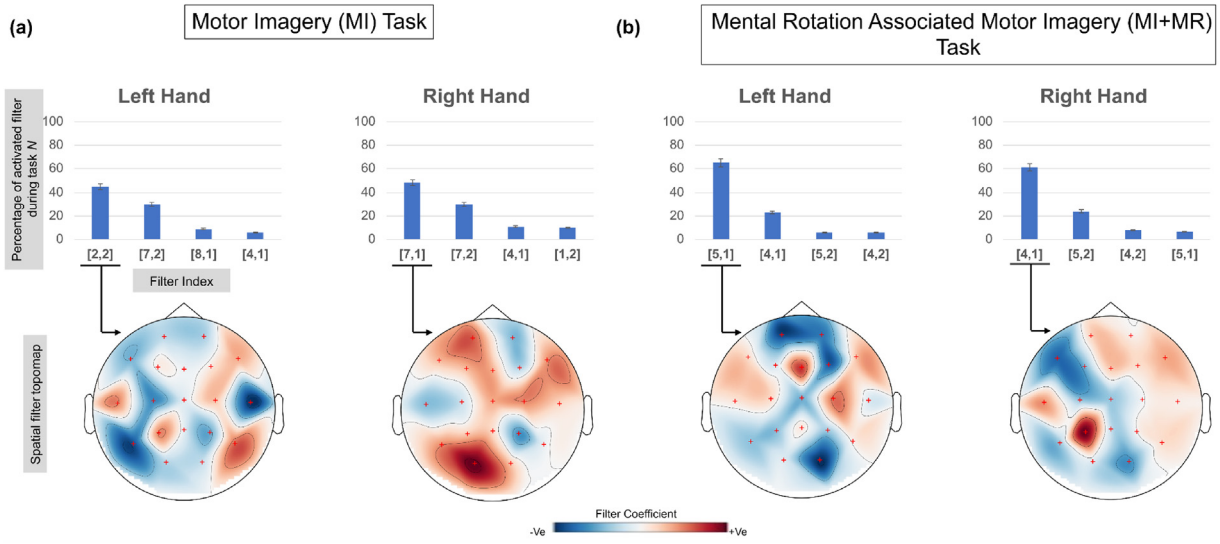


Fig. 8 – Subject 12 as an example: (a) highest obtained index [*i_max,j_max*] and its corresponding spatial filter visualization for MI task; (b) MI + MR task.

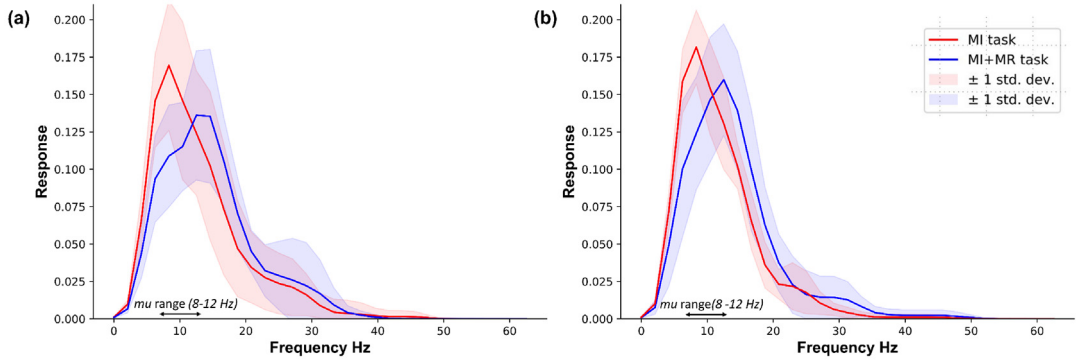


Fig. 9 – Average cumulative Sinc filters frequency response of all subjects: (a) Sinc-EEGNet; (b) Sinc-CSPNN.

proposed CNN model's spatial filters layer. By visualizing learned spatial filter weights as a topographic head map, analysis can be performed since these weights highlight cortical areas that the network focuses on to solve the classification task. In both Dataset I and Dataset II, areas corresponding to weights emphasized by the network (weights with relatively large coefficient value) were neurophysiologically related to the task being performed. For example, in Dataset I (Figs. 4 and 5, both Sinc-EEGNet and Sinc-CSPNN emphasize cortical motor areas C_3 and C_4 during *right* and *left* hand imagery movement, respectively. Similar results can also be observed in Dataset II, albeit with a different topographic EEG layout. An interesting observation in this dataset is that, compared to MI task, the proposed Sinc-EEGNet was able to emphasize weights corresponding to the prefrontal areas in the MI + MR task. Such emphasis was previously mentioned to occur during this task since the beta wave presence is ordinarily apparent in frontal and occipital lobes.

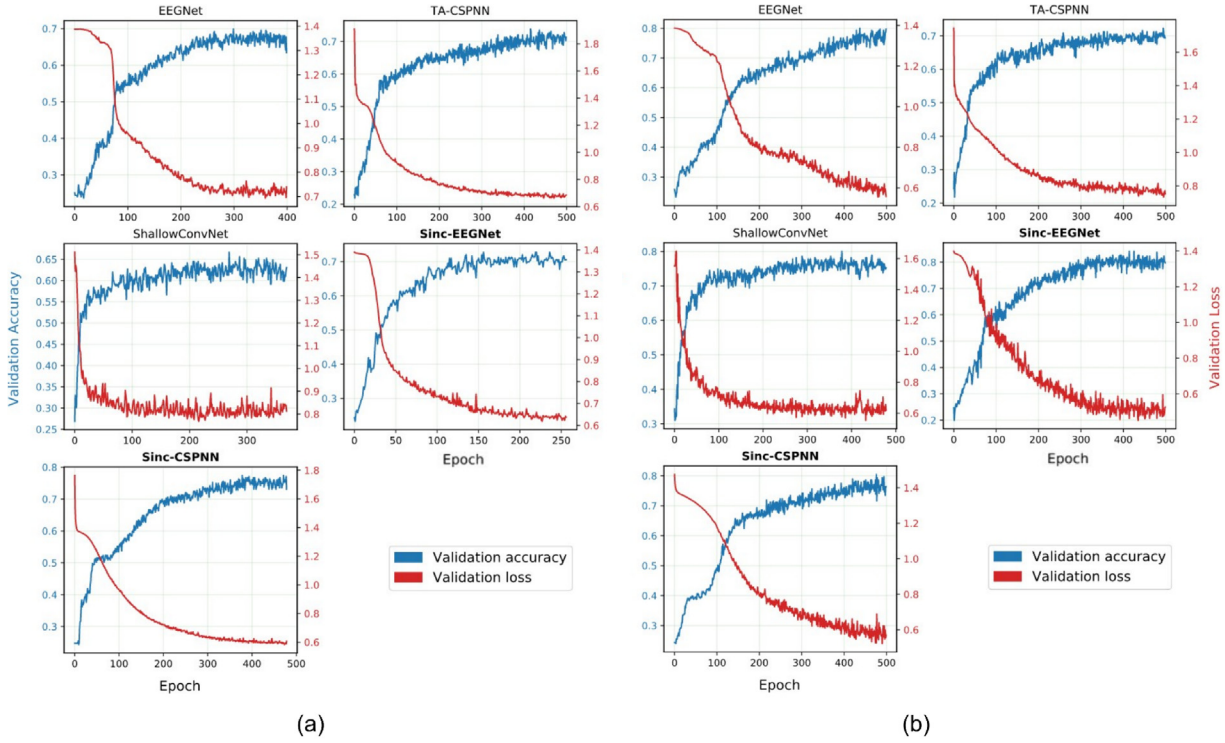


Fig. 10 – Average validation accuracy-loss curves for: (a) **Dataset I (secondary data)** (b) **Dataset II (primary data).**

Table 8 – Training time characteristics of tested CNN architectures.

	Shallow-ConvNet	EEGNeT	TA-CSPNN	**Sinc-EEGNeT**	**Sinc-CSPNN**
Average number of epochs	336 (± 98)	341 (± 88)	350 (± 99)	328 (± 88)	347 (± 89)
Time per epoch [ms]	46.42	30.55	20.86	22.42	21.65

Table 9 – Average performance metrics across all subjects for FBCSP-CNN classification.

Dataset	Accuracy	Precision	Recall	Kappa	AUC
Dataset I	0.61	0.631	0.625	0.517	0.611
Dataset II (MI task)	0.52	0.545	0.565	0.467	0.549
Dataset II (MI+MR task)	0.55	0.587	0.575	0.512	0.552

5. Conclusion

This paper presented two compact CNN architectures for decoding motor imagery signals based on SincNet. Compared to other SOTA models, the proposed models perform at par or, in some cases better, with the least number of trainable parameters. The proposed architecture was also shown to better adapt to the necessary cortical rhythms related to the given task and perform a cleaner band-pass filtering over these rhythms. Besides spectral visualization and analysis, SincNet also enables the development algorithm 1, in which spatial filters in the CNN's depthwise convolution layers are extracted for tasks related to cortical analysis.

In the future, the applicability of proposed Sinc based models will be expanded to other biomedical domains. One example is similar to the work in [40] in which SincNet model

is used for emotion classification from EEG signals. Another interesting application that could be explored is the use of proposed model towards epileptic seizure detection [41,42] in which the interpretability of SincNet could benefits clinicians and healthcare practitioners towards providing accurate epilepsy diagnostic. Classification of other BCI paradigms such as the recent concurrent SSVEP and P300 EEG features [43] can also be explored using the proposed model. Apart from EEG, the proposed algorithm can also be applied towards other biosignal classification, such as detecting heart failures from electrocardiogram (ECG) signals [44–46], detecting amytrophic lateral sclerosis (ALS) disease from EMG [47] and distinguishing audiovisual inputs for cocktail party problem [48], just to name a few.

Declaration of Competing Interest

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

Acknowledgement

The author would like to thank Universiti Teknikal Malaysia Melaka (UTeM) Center for Robotics and Industrial Automation (CERIA), and Universiti Teknologi Malaysia (UTM), Biomedical Instrumentation and Electronic (bMIE) research group for the financial and facilities support.

Compliance with ethical standards

All procedures performed in this study were in accordance with the ethical standards of the institutional and national research committee and in a compliance with the 1964 Helsinki Declaration or its later amendments. For Dataset II (primary data) ethical approval was obtained from the Ministry of Health Malaysia (**NMRR-19-1671-47228)** and informed consent was given and retrieved from all individuals participating in the study.

Appendix A

Table A1 – Summary of parameters of the proposed Sinc based architecture.

Sinc-EEGNet			Sinc-CSPNN		
Layer	Filter size	Output	Layer	Filter size	Output
Input		$(1, C, T)$	Input		$(1, C, T)$
Sinc Convolution 2D	$F_K \times 2$	(F_K, C, T)	Sinc Convolution 2D	$F_K \times 2$	(F_K, C, T)
Batch Normalisation		(F_K, C, T)	Batch Normalisation		(F_K, C, T)
Depthwise Convolution	$C \times F_s$	$(F_K \times F_s, 1, T)$	Depthwise Convolution	$C \times F_s$	$(F_K \times F_s, 1, T)$
Activation: ELU		$(F_K \times F_s, 1, T)$	Activation: RELU		$(F_K \times F_s, 1, T)$
Average pooling		$(F_K \times F_s, 1, T/4)$	Average pooling		$(F_K \times F_s, 1, T/4)$
Dropout		$(F_K \times F_s, 1, T/4)$	Dropout		$(F_K \times F_s, 1, T/4)$
Separable Convolution	$(16 \times F_K \times F_s) + (F_C \times F_K \times F_s)$	$(F_C, 1, T/4)$	Flatten		$(F_K \times F_s)$
BatchNorm		$(F_C, 1, T/4)$	Dense		N
Activation: ELU		$(F_C, 1, T/4)$	Softmax		N
Average pooling		$(F_C, 1, T/32)$			
Dropout		$(F_C, 1, T/32)$			
Flatten		$(F_C \times (T/32))$			
Dense		N			
Softmax		N			

REFERENCES

[1] Rao RPN. Brain-computer interfacing: an introduction. Cambridge University Press; 2011. 10.1017/CBO9781139032803.

[2] Lebedev MA, Nicolelis MAL. Brain-machine interfaces: from basic science to neuroprostheses and neurorehabilitation. Physiol Rev 2017;97:767–837. https://doi.org/10.1152/physrev.00027.2016.

[3] Abdulkader SN, Atia A, Mostafa M-S. Brain computer interfacing: applications and challenges. Egypt Informatics J 2015;16(2):213–30. https://doi.org/10.1016/j.eij.2015.06.002.

[4] Amiri S, Fazel-Rezai R, Asadpour V. A review of hybrid brain-computer interface systems. Adv Human-Computer Interact 2013;2013:1–8. https://doi.org/10.1155/2013/187024.

[5] Brodu N, Lotte F, Lecuyer A. Comparative study of band-power extraction techniques for Motor Imagery classification. In: IEEE Symp. Comput. Intell Cogn. Algorithms, Mind, Brain. IEEE; 2011. p. 1–6. 10.1109/CCMB.2011.5952105.

[6] Wang Y, Gao S, Gao X. Common spatial pattern method for channel selelction in motor imagery based brain-computer. Interface 2006:5392–5. https://doi.org/10.1109/iembs.2005.1615701.

[7] Jin J, Miao Y, Daly I, Zuo C, Hu D, Cichocki A. Correlation-based channel selection and regularized feature optimization for MI-based BCI. Neural Networks 2019;118:262–70. https://doi.org/10.1016/j.neunet.2019.07.008.

[8] Jin J, Fang H, Daly I, Xiao R, Miao Y, Wang X, et al. Optimization of model training based on iterative minimum covariance determinant in motor-imagery BCI. Int J Neural Syst 2021;31:2150030. https://doi.org/10.1142/S0129065721500301.

[9] Ang KK, Chin ZY, Wang C, Guan C, Zhang H. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. Front Neurosci 2012;6:1–9. https://doi.org/10.3389/fnins.2012.00039.

[10] Kai Keng Ang, Zheng Yang Chin, Haihong Zhang, Cuntai Guan, Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface, in: 2008 IEEE Int. Jt. Conf. Neural Networks (IEEE World Congr. Comput. Intell., 2008: pp. 2390–2397. 10.1109/IJCNN.2008.4634130.

[11] Jin J, Xiao R, Daly I, Miao Y, Wang X, Cichocki A. Internal feature selection method of CSP based on L1-norm and dempster-shafer theory. IEEE Trans Neural Networks Learn Syst PP 2020:1–12. https://doi.org/10.1109/TNNLS.2020.3015505.

[12] Wang H, Tang C, Xu T, Li T, Xu L, Yue H, et al. An approach of one-vs-rest filter bank common spatial pattern and spiking neural networks for multiple motor imagery decoding. IEEE Access 2020;8:86850–61. https://doi.org/10.1109/ACCESS.2020.2992631.

[13] Barachant A, Bonnet S, Congedo M, Jutten C. Multiclass brain-computer interface classification by riemannian geometry. IEEE Trans Biomed Eng 2012;59(4):920–8. https://doi.org/10.1109/TBME.2011.2172210.

[14] Barachant A, Bonnet S, Congedo M, Jutten C. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. Neurocomputing 2013;112:172–8. https://doi.org/10.1016/j.neucom.2012.12.039.

[15] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recognit. IEEE; 2016. p. 770–8. 10.1109/CVPR.2016.90.

[16] Sainath TN, Kingsbury B, Saon G, Soltau H, Mohamed AR, Dahl G, et al. Deep convolutional neural networks for large-scale speech tasks. Neural Networks 2015;64:39–48. https://doi.org/10.1016/j.neunet.2014.08.005.

[17] Schirrmeister RT, Springenberg JT, Fiederer LDJ, Glasstetter M, Eggensperger K, Tangermann M, et al. Deep learning with convolutional neural networks for EEG decoding and visualization. Hum Brain Mapp 2017;38(11):5391–420. https://doi.org/10.1002/hbm.23730.

[18] Tayeb Z, Fedjaev J, Ghaboosi N, Richter C, Everding L, Qu X, et al. Validating deep neural networks for online decoding of motor imagery movements from eeg signals. Sensors (Switzerland) 2019;19. https://doi.org/10.3390/s19010210.

[19] Amin SU, Alsulaiman M, Muhammad G, Bencherif MA, Hossain MS. Multilevel weighted feature fusion using convolutional neural networks for EEG motor imagery classification. IEEE Access 2019;7:18940–50. https://doi.org/10.1109/ACCESS.2019.2895688.

[20] Tang X, Wang T, Du Y, Dai Y. Motor imagery EEG recognition with KNN-based smooth auto-encoder. Artif Intell Med 2019;101. https://doi.org/10.1016/j.artmed.2019.101747 101747.

[21] Lawhern VJ, Solon AJ, Waytowich NR, Gordon SM, Hung CP, Lance BJ. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. J Neural Eng 2018;15. https://doi.org/10.1088/1741-2552/aace8c 056013.

[22] Mousavi M, de Sa VR. Temporally adaptive common spatial patterns with deep convolutional neural networks. In: 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE; 2019. p. 4533–6. 10.1109/EMBC.2019.8857423.

[23] Tang Z, Li C, Sun S. Single-trial EEG classification of motor imagery using deep convolutional neural networks. Optik (Stuttg) 2017;130:11–8. https://doi.org/10.1016/j.ijleo.2016.10.117.

[24] Ravanelli M, Bengio Y, Recognition S. Speaker recognition from raw waveform with SincNet. In: IEEE Spok. Lang. Technol. Work. IEEE; 2018. p. 1021–8. 10.1109/SLT.2018.8639585.

[25] Olivas-Padilla BE, Chacon-Murguia MI. Classification of multiple motor imagery using deep convolutional neural networks and spatial filters. Appl Soft Comput J 2019;75:461–72. https://doi.org/10.1016/j.asoc.2018.11.031.

[26] JEL (Hanoi) H, CG, Roberts A. DDSP:Differentiable Digital Signal Processing, in: Int. Conf. Learn. Represent., 2020: pp. 1–19.

[27] Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. Striving for simplicity: The all convolutional net, in: 3rd Int. Conf. Learn. Represent. ICLR 2015 – Work. Track Proc., 2015.

[28] Wolpaw JR, McFarland DJ, Vaughan TM. Brain-computer interface research at the Wadsworth Center. IEEE Trans Rehabil Eng 2000. https://doi.org/10.1109/86.847823.

[29] Tangermann M, Müller K-R, Aertsen A, Birbaumer N, Braun C, Brunner C, et al. Review of the BCI competition IV. Front Neurosci 2012;6. https://doi.org/10.3389/fnins.2012.00055.

[30] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15:1929–58.

[31] Jeunet C, N'Kaoua B, Subramanian S, Hachet M, Lotte F, Friedman D. Predicting mental imagery-based BCI performance from personality, cognitive profile and neurophysiological patterns. PLoS ONE 2015;10(12):e0143962. https://doi.org/10.1371/journal.pone.0143962.

[32] Kubler A, Holz E, Kaufmann T, Zickler C. A User Centred Approach for Bringing BCI Controlled Applications to End-Users, in: Brain-Computer Interface Syst. - Recent Prog. Futur. Prospect., InTech, 2013. https://doi.org/10.5772/55802.

[33] Ma X, Qiu S, Wei W, Wang S, He H. Deep channel-correlation network for motor imagery decoding from the same limb. IEEE Trans Neural Syst Rehabil Eng 2020;28(1):297–306. https://doi.org/10.1109/TNSRE.2019.2953121.

[34] Garcia-Rill E. The 10Hz Fulcrum, in: Waking Reticular Act. Syst. Heal. Dis., Elsevier, 2015: pp. 157–170. 10.1016/B978-0-12-801385-4.00008-2.
[35] Kübler A, Mattia D. Brain-computer interface based solutions for end-users with severe communication disorders. In: Neurol Conciousness. Elsevier; 2016. p. 217–40. 10.1016/B978-0-12-800948-2.00014-5.
[36] Abhang PA, Gawali BW, Mehrotra SC. Technical Aspects of Brain Rhythms and Speech Parameters, in: Introd. to EEG-Speech-Based Emot. Recognit., Elsevier, 2016: pp. 51–79. https://doi.org/10.1016/B978-0-12-804490-2.00003-8.
[37] Satapathy SK, Dehuri S, Jagadev AK, Mishra S. Introduction. In: EEG Brain Signal Classif Epileptic Seizure Disord Detect. Elsevier; 2019. p. 1–25. 10.1016/B978-0-12-817426-5.00001-6.
[38] Izzuddin TA, Safri NM, Othman MA. Mental imagery classification using one-dimensional convolutional neural network for target selection in single-channel BCI-controlled mobile robot. Neural Comput Appl 2020. https://doi.org/10.1007/s00521-020-05393-6.
[39] Lim S, Yeo M, Yoon G. Comparison between concentration and immersion based on EEG analysis. Sensors 2019;19:1669. https://doi.org/10.3390/s19071669.
[40] Zeng H, Wu Z, Zhang J, Yang C, Zhang H, Dai G, et al. EEG emotion classification using an improved sincnet-based deep learning model. Brain Sci 2019;9. https://doi.org/10.3390/brainsci9110326.
[41] Emami A, Kunii N, Matsuo T, Shinozaki T, Kawai K, Takahashi H. Seizure detection by convolutional neural network-based analysis of scalp electroencephalography plot images. NeuroImage Clin 2019;22. https://doi.org/10.1016/j.nicl.2019.101684 101684.
[42] Zhou M, Tian C, Cao R, Wang B, Niu Y, Hu T, et al. Epileptic seizure detection based on EEG signals and CNN. Front Neuroinform 2018;12. https://doi.org/10.3389/fninf.2018.00095.
[43] Xu M, Han J, Wang Y, Jung T-P, Ming D. Implementing Over 100 command codes for a high-speed hybrid brain-computer interface using concurrent P300 and SSVEP features. IEEE Trans Biomed Eng 2020;67(11):3073–82. https://doi.org/10.1109/TBME.2020.2975614.
[44] Acharya UR, Oh SL, Hagiwara Y, Tan JH, Adam M, Gertych A, et al. A deep convolutional neural network model to classify heartbeats. Comput Biol Med 2017;89:389–96. https://doi.org/10.1016/j.compbiomed.2017.08.022.
[45] Porumb M, Iadanza E, Massaro S, Pecchia L. Biomedical signal processing and control a convolutional neural network approach to detect congestive heart failure. Biomed Signal Process Control 2020;55. https://doi.org/10.1016/j.bspc.2019.101597 101597.
[46] Abdul-Kadir NA, Mat Safri N, Othman MA. Atrial fibrillation classification and association between the natural frequency and the autonomic nervous system. Int J Cardiol 2016;222:504–8. https://doi.org/10.1016/j.ijcard.2016.07.196.
[47] Sengur A, Akbulut Y, Guo Y, Bajaj V. Classification of amyotrophic lateral sclerosis disease based on convolutional neural network and reinforcement sample learning algorithm. Heal Inf Sci Syst 2017;5:9. https://doi.org/10.1007/s13755-017-0029-6.
[48] Li Y, Wang F, Chen Y, Cichocki A, Sejnowski T. The effects of audiovisual inputs on solving the cocktail party problem in the human brain: an fMRI study. Cereb Cortex 2018;28:3623–37. https://doi.org/10.1093/cercor/bhx235.

Fully learnable deep wavelet transform for unsupervised monitoring of high-frequency time series

Gabriel Michau[a], Gaetan Frusque[a], and Olga Fink[a,1]

[a]Chair of Intelligent Maintenance Systems, ETH Zürich, 8049 Zürich, Switzerland

Edited by David Donoho, Department of Statistics, Stanford University, Stanford, CA; received April 7, 2021; accepted January 10, 2022

High-frequency (HF) signals are ubiquitous in the industrial world and are of great use for monitoring of industrial assets. Most deep-learning tools are designed for inputs of fixed and/or very limited size and many successful applications of deep learning to the industrial context use as inputs extracted features, which are a manually and often arduously obtained compact representation of the original signal. In this paper, we propose a fully unsupervised deep-learning framework that is able to extract a meaningful and sparse representation of raw HF signals. We embed in our architecture important properties of the fast discrete wavelet transform (FDWT) such as 1) the cascade algorithm; 2) the conjugate quadrature filter property that links together the wavelet, the scaling, and transposed filter functions; and 3) the coefficient denoising. Using deep learning, we make this architecture fully learnable: Both the wavelet bases and the wavelet coefficient denoising become learnable. To achieve this objective, we propose an activation function that performs a learnable hard thresholding of the wavelet coefficients. With our framework, the denoising FDWT becomes a fully learnable unsupervised tool that does not require any type of pre- or postprocessing or any prior knowledge on wavelet transform. We demonstrate the benefits of embedding all these properties on three machine-learning tasks performed on open-source sound datasets. We perform an ablation study of the impact of each property on the performance of the architecture, achieve results well above baseline, and outperform other state-of-the-art methods.

fast discrete wavelet decomposition | deep learning | high-frequency signals | unsupervised anomaly detection | sparse decomposition

Monitoring of industrial assets often relies on high-frequency (HF) signal measurements, such as electric current, vibrations, or sound. One difficulty of dealing with such signals in the industrial context is the conciliation between the high-frequency sampling and low-dimensional decision states (e.g., healthy/unhealthy), in a context where, very often, labels are not available. Therefore, many industrial applications require unsupervised approaches able to extract meaningful and sparse information from HF signals, to ease the process analysis, the diagnostics, and more generally the optimization of the assets' life cycles.

Before the recent developments of large storage capacity and high computational powers, raw HF signals could not be recorded, forcing companies to spend time and budget on devising relevant features for later analysis, achieving in that way a sparse representation of the input data. These features could be of various natures, such as spectral features, based on the Fourier transform, the fast Fourier transform, or wavelets (1); on statistical features (moments, energy, entropy, etc.); or on descriptive features (envelopes, amplitude, etc.) (2).

In recent days, storing HF data has become less of a technical problem, and handling large datasets efficiently has been made possible with the rise of deep learning. However, most deep-learning tools are designed for inputs of fixed and/or very limited size. Many successful applications of deep learning to industrial context use as inputs extracted features, that is, a manually obtained compact representation of the original signals. Very often these features are a spectrogram (3), wavelet coefficient statistics (4), or others (5–7). Although such frameworks are extremely successful, they still require careful feature extraction with the right hyperparameters, which can be a time-consuming task. In addition, the extracted features might be sensitive to unexpected noise or to changing conditions, and the design of domain invariant unsupervised features, whether with postprocessing or with deep learning, is still an open research question (8).

With the development of convolutional neural networks (CNN) (9), early works realized that a temporal CNN is equivalent to a digital filter and that it could learn convolution kernels similar to a Fourier transform or to wavelets (10), or also be used to learn sparse representations (11). It was soon proposed to constrain the network to perform operations similar to the Fourier transform (12, 13) or to wavelet transform either with continuous wavelet transform (14, 15) or with discrete wavelet transform (16–19). All these works demonstrate that by using architectures or kernels inspired by spectral analysis, superior results could be obtained on supervised deep-learning tasks. Yet, these approaches are rarely adapted to unsupervised machine-learning tasks and the link with the spectral transformation is often restrained either to the network architecture only or to the initialization of the convolution kernels. In addition, Fourier transform-based deep-learning architectures become rapidly too heavy to handle when the size of the input time series increases.

To mitigate the abovementioned limitations, we propose in this work a deep-learning framework based on the fast discrete wavelet transform (FDWT) that allows an automatic and easy extraction of meaningful and sparse representation of the input

Significance

Monitoring of industrial assets often relies on high-frequency (HF) signal measurements. One difficulty of dealing with such measurements in the industrial context is the conciliation between the high-frequency sampling and low-dimensional decision states (e.g., healthy/unhealthy), in a context where, very often, labels are not available. Here, we propose a fully unsupervised deep-learning framework for high-frequency time series that is able to extract meaningful and sparse representation of raw signals and is able to handle different lengths of time series flexibly, overcoming thereby several of the limitations of existing deep-learning approaches. The decomposition framework will be very useful for handling in an automatic manner high-frequency signals and is an important basis for future applications with HF data.

Author contributions: G.M. and O.F. designed research; G.M. performed research; G.M., G.F., and O.F. analyzed data; and G.M., G.F., and O.F. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: ofink@ethz.ch.

Published February 18, 2022.

signals. First, we propose here to mimic the FDWT cascade architecture utilizing the deep-learning framework. We, thus, propose to learn at each decomposition level on the one hand the right high- and low-pass filters and on the other hand the right hard-thresholding coefficients for denoising. Second, for the learning of the filters, we take advantage of the long-known advantageous properties of orthogonal wavelet filter banks with the conjugate quadrature filter property to structure the learning process while making sure the final network remains similar to a FDWT operation. It also results in a very light architecture with only few parameters to learn (a few hundred). This is opposite to the general trend in deep learning to learn millions or billions of parameters. Third, to learn the right hard-thresholding operation, we propose a learnable activation function that is continuous and differentiable, and approximates the hard-thresholding operation. It can, thus, be used as an integral part of the deep-learning architecture and removes the need for human analysis and decisions on these difficult tasks (20).

After presenting in *1. Background on the Cascade and FDWT* important properties and characteristics of the FDWT, we show how to translate these properties into a deep neural network in *2. Learnable Denoising Sparse Wavelet Network*. In *3. Comparison between Traditional FDWT and DeSpaWN*, we test our approach on three tasks, one classification task and two unsupervised anomaly detection tasks. Starting from the FDWT, we demonstrate how each of our different contributions contributes toward better results, well above the baseline. Finally, in *4. Comparison to Other Frameworks*, we compare our approach to several other architectures, such as the scattering transform (21), U-Net (22), and a convolutional autoencoder. We show that without the need of any preprocessing steps, with our particularly light architecture, we achieve very competitive results.

1. Background on the Cascade and FDWT

A. Cascade Algorithm. The FDWT uses wavelets designed such that the family of wavelets made of their scaling and translation by any power of 2 makes an orthonormal family (23). Using such wavelets, and their corresponding scaling function, FDWT decomposes successively each approximation of a signal f into a coarser approximation a (low-pass–filtered version of the signal f) and its detail coefficients d (also denoted as wavelet coefficients, high-pass–filtered version of f). With an orthonormal basis of $\mathbf{L}^2(\mathbb{R})$, the decomposition of any function with such a transformation is invertible. Additionally, one can demonstrate that, due to the factor 2 in scale between the levels and in translation between the coefficients, the decomposition at level l can be expressed as a function of the previous approximation, subsampled by a factor 2 and the original nondilated wavelet (interested readers are referred to ref. 23, chap. 7.3.1 for the proof). Similarly, for the reconstruction at each level, it can be expressed as the convolution of the conjugate of the original wavelet with the previous level reconstruction, where zeros have been interpolated between every sample. FDWT uses this property to apply, in cascade, the exact same operation at each level, using a single wavelet, but down-scaling and interpolating with zeros the signals at each level of the decomposition and the reconstruction, respectively. It is denoted as the cascade algorithm. Fig. 1 illustrates the FDWT cascade algorithm, where g is the wavelet, h is the scaling function, and $\bar{g}$ and $\bar{h}$ are their respective conjugate filters.

B. CQF Properties. An interesting property of the fast discrete wavelet transform was discovered in 1976 by Croisier, Esteban, and Galand (23, 24) and was extended in 1984 (23, 25, 26). It establishes the ground for finding filters, allowing a perfect reconstruction of the input signal using the conjugate quadrature filter (CQF) bank property. The quadrature property ensures a symmetric response of the decomposition filters with respect to the cutoff frequency and ensures thus an antialiasing property. To do so, the filters can be designed such that the wavelet function g is the alternative flip of the scaling function h. The conjugate property ensures that the reconstruction filters have an anticancellation property. Both properties combined are usually denoted as a CQF, which is formalized as follows:

$$\begin{cases} g[n] &= (-1)^n \cdot h[-n], \\ \bar{h}[n] &= h[-n], \\ \bar{g}[n] &= (-1)^{(n+1)} \cdot h[n], \end{cases} \quad [1]$$

where $h[-n]$ denotes the nth coefficient of h in reversed order.

To achieve a perfect reconstruction, the frequency content conservation after applying the filters imposes that the sum of the responses of both filters should be 2, which imposes further

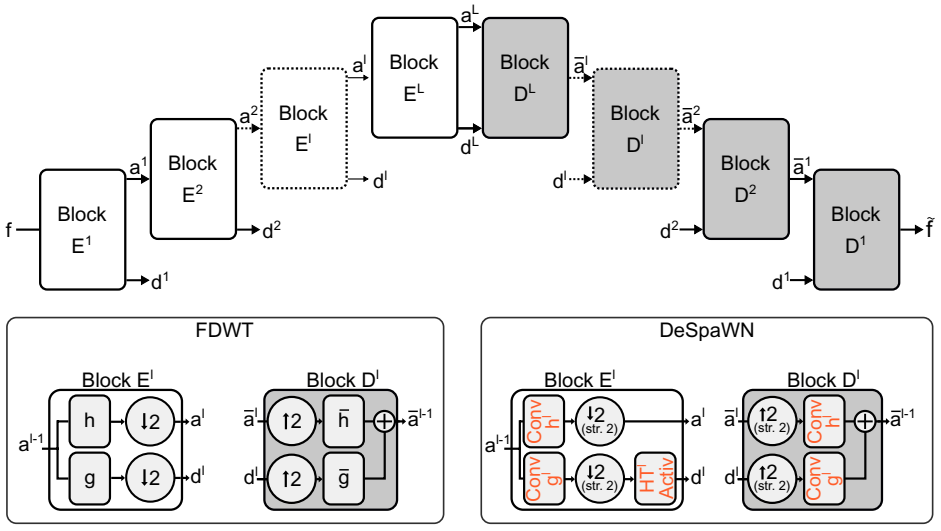


Fig. 1. Cascade: FDWT versus DeSpaWN. The FDWT can be modeled as a convolution neural network, an autoencoder of 2L layers, comprising L encoding blocks and L decoding blocks. Each encoding block has two outputs, one of which is connected to the corresponding decoding block with skip connections. In this work, we propose to make the network learnable, that is, to learn the right filters. In addition, we propose a learnable hard-thresholding activation function that allows one to learn the wavelet coefficient denoising operation at the same time. Red elements shown are learnable. The proposed architecture mimics the denoising FDWT and is denoted as DeSpaWN.

constraints on the filters' orthonormality (23), usually considered as part of the CQF properties.

C. Signal Denoising with FDWT. One of the major applications of the wavelet analysis is signal denoising (27, 28). The usual assumption is that regular and structured signals, once decomposed under the right wavelet basis, will naturally lead to a sparse decomposition (23). They will activate only certain wavelet coefficients at specific time and decomposition levels. As a consequence, noise, by nature unstructured, activates the wavelet filters at any level, but usually with a small amplitude. Thus, denoising usually consists of applying a hard thresholding function to the resulting wavelet coefficients before applying the reconstruction algorithm (29). However, finding the right thresholding parameters is a difficult task that has been the topic of extensive research (20).

2. Learnable Denoising Sparse Wavelet Network

A. Architecture Overview. In this research, we propose the denoising sparse wavelet network (DeSpaWN). DeSpaWN utilizes a fully learnable cascade network, mimicking an L-level wavelet cascade, such as illustrated in Fig. 1. It, thus, consists of L encoding blocks and L decoding blocks. Each encoding block l is composed of two learnable convolutional layers g^l and h^l with a stride of two, analogous to the wavelet and scaling filters with down-sampling in the FDWT. The resulting coefficients are fed to a specifically designed learnable hard-thresholding layer HT^l, which is similar to a wavelet denoising operation and to the next block for further decomposition. Similar to the FDWT, each decoding block takes two inputs, the coefficients from the previous block and the detail coefficients from its corresponding encoding block l. It applies to each input a learnable convolution transpose layer $\bar{g}^l$ and $\bar{h}^l$ with a stride of two and sums the results of both layers together. We designed the deep neural network with two main distinctive properties: First, all convolution kernels and second, all positive and negative hard thresholds are learnable. This makes it possible to learn fully and in a completely unsupervised way the most adapted denoising FDWT for the input signals.

In addition, according to the work in ref. 30, a sparsely connected deep neural network, such as the one proposed here, is able to approximate representation systems that encompass and are more general than the representation system provided by the wavelet.

Overall, with the proposed architecture, the network has $(k_n + 2) \cdot L$ learnable parameters, where k_n is the number of coefficients of the wavelets. For example, mimicking Daubechies-4 wavelets, k_n is set to 8 and the network has, thus, $10 \cdot L$ learnable parameters. Since L cannot be set larger than the nearest second logarithm of the training input size, the number of parameters is unlikely to exceed a few hundred.

B. Objective Function. In this work, we propose to learn the best wavelet and scaling functions for achieving a sparse decomposition. This means that we have two learning objectives: first, a good signal reconstruction and second, a sparse decomposition. Sparsity is usually measured by achieving the smallest ℓ_0 norm of the resulting coefficients. Yet, this is a nonconvex metric. A typical convex surrogate of the ℓ_0 norm is the ℓ_1 norm. Part of our objective function should be designed to minimize the ℓ_1 norm of the obtained wavelet coefficients. As a consequence, we propose that for the second part of our objective function, we also measure the ℓ_1 norm of the reconstruction error to make the two terms comparable.

We, thus, train our network with the following objective function:

$$\mathcal{L} = \frac{1}{\text{Card}(f)} |f - \tilde{f}|_1 + \gamma \cdot \mathbf{L}\left(\{\mathbf{d^l}\}_{l\in[1..L]}, \mathbf{a^L}\right), \qquad [2]$$

where the first part of the loss is the averaged ℓ_1 norm of the residuals on the reconstruction and the second part is the sparsity term, here proposed as the average of all wavelet coefficient moduli and of the last layer approximation coefficient modulus,

$$\mathbf{L}\left(\{\mathbf{d^l}\}_{l\in[1..L]}, \mathbf{a^L}\right) = \frac{1}{\text{Card}(\{\mathbf{d^l}\}_{l\in[1..L]}, \mathbf{a^L})} \sum_{l\in[1..L]} |\mathbf{d^l}|_1 + |\mathbf{a^L}|_1. \qquad [3]$$

C. CQF-Constrained Architecture. As seen in the previous section, the CQF property for the wavelet filters ensures a perfect and antialiasing reconstruction. We, thus, propose to utilize this property, at the same time to simplify the learning process and to harness its advantages. As a matter of fact, using the CQF property as defined in Eq. **1**, learning a single kernel would define the other three. The second advantage is that, based on the above idea of penalizing the ℓ_1 norm of the wavelet coefficient as a surrogate for the nonconvex ℓ_0 norm, the constraint-free learning of the kernels might lead toward a state where the coefficients of g and h are minimized to result in a small ℓ_1 value in the latent space, while the reconstruction is still possible due to large coefficients in $\bar{g}$ and $\bar{h}$, which goes against the original goal of achieving sparsity. Constraining the values of $\bar{g}$ and $\bar{h}$ based on those of g and/or h mitigates this issue.

One option would be to learn a single kernel h^0; use the CQF constraints to derive g^0, $\bar{g}^0$, and $\bar{h}^0$; and then impose that all layers of the network use the same kernels, mimicking in that way the traditional wavelet decomposition with a single wavelet basis. However, we state that this approach would not benefit from the full potential of deep learning.

Alternatively, we propose to learn one kernel per layer $\{h^l\}_{l\in[1..L]}$ and to use the CQF property to constrain the other three kernels of the layer (g^l, $\bar{h}^l$, $\bar{g}^l$). A similar alternative would be to learn independently the low-pass h^l and high-pass g^l filters but impose the partial CQF constraints such that $\bar{h}^l[n] = h^l[-n]$ and $\bar{g}^l[n] = g^l[-n]$. These two types of architectures learn at each level the best wavelets to describe the signal at this scale in a sparse way.

We implement all these different variations of the CQF property and compare them in *3. Comparison between Traditional FDWT and DeSpaWN*.

Finally, with the proposed architecture we already cover our three training objectives: a good and a sparse reconstruction due to the objective function and a stable learning due to the constraints imposed between decomposition and reconstruction filters. Therefore, we do not try to impose further constraints on the filters, in particular not the orthonormality property.

D. Learnable Denoising. One assumption behind the wavelet decomposition is that a structured signal should lead to sparse coefficients under the right wavelet basis. This is the property we encourage by minimizing the ℓ_1 norm of the wavelet coefficients in our objective function. However, the addition of noise to the input signal, which is by nature nonstructured, would lead to a random activation of the filters, independent of the chosen filters. This would, therefore, necessarily lead to a nonsparse decomposition. The sparsity of the decomposition is, thus, sought only once the noise has been canceled. This problem is usually tackled under the assumption that noise would lead to small activation of the filters. As a consequence, the impact of noise on the decomposition can be removed by hard thresholding the obtained coefficients. In ref. 31, guarantees are provided for recovering and denoising signals observed in Gaussian noise by applying the right hard-thresholding operation. Yet, finding the right thresholding parameters is a difficult task and usually depends on the use case and the specific dataset.

In this paper, we propose to make the thresholding step part of our architecture to learn the best thresholding parameters and to

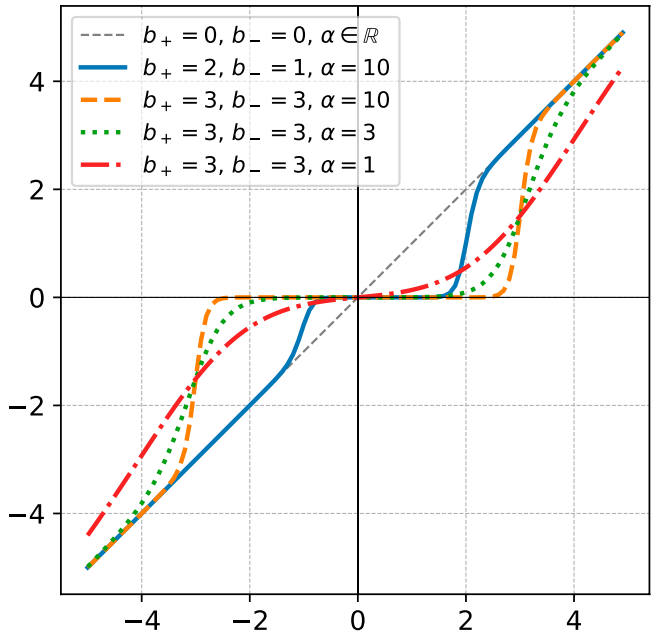


Fig. 2. Hard-thresholding activation function. Proposed activation function performs an operation close to an asymmetric hard thresholding. Both thresholds, in the negative and positive half-space, are learnable parameters. In our paper, α is set to 10 to simulate a "sharp" thresholding.

remove the need of handling it as a separate step. We introduce a learnable hard-thresholding activation function as a combination of two opposite sigmoid functions:

$$\forall x \in \mathbb{R}, \quad HT(x) = x \left[\frac{1}{1 + \exp\left(\alpha \cdot (x + b_-)\right)} + \frac{1}{1 + \exp\left(-\alpha \cdot (x - b_+)\right)} \right], \quad [4]$$

where α is a "sharpness" factor arbitrarily fixed to 10 in this paper, and b_+ and b_- are the positive and negative learnable biases acting as the thresholds on both sides of the origin, as illustrated in Fig. 2.

Denoting the sigmoid function $1/(1 + \exp(-x))$ by σ, Eq. **4** becomes

$$\forall x \in \mathbb{R}, \quad HT(x) = x \cdot \left[\sigma\left(-\alpha \cdot (x + b_-)\right) + \sigma\left(\alpha \cdot (x - b_+)\right)\right]. \quad [5]$$

To replicate the original FDWT without denoising, one can fix in this layer b_+ and b_- to zero, enforcing, thereby, a linear activation.

3. Comparison between Traditional FDWT and DeSpaWN

A. Machine Learning for Sound Data. We focus our experiments specifically on sound data as they are a fast-growing field in industrial applications (4, 32). Sound-based analysis has raised the interest of increasing numbers of companies for several reasons: Experts and machine operators already listen to the machines and are able to tell when the operation is not nominal. It should, therefore, be possible to train a machine-learning approach for continuous monitoring. Moreover, industrial processes are inherently noisy. Thus, it is expected that their sound contains information on their process state. Since the sound is by nature multiscale, it might also allow for the monitoring of several systems at once. Monitoring industrial machines with sound is also relatively simple and cheap. Hence, it is an attractive and scalable solution. Microphones are easy to install or retrofit. They are not intrusive. Hardware and software are readily available. Yet, sound data come with their own challenges such as high noise levels due to the machines usually operating in factory environments and, hence, in noisy environments. Besides, finding how to relate specific industrial processes to the recorded sound is a difficult task. Automatic and noise-robust handling of sound data is, therefore, of high interest for many industrial applications.

B. Classification and Anomaly Detection. The objective of DeSpaWN is learning of a robust autoencoder. This robustness is achieved through denoising with the hard thresholding of the learned coefficients and through sparsity, forcing the network to learn the most meaningful wavelet to describe the training data. Thus, we state that once trained, the autoencoder should produce for similar signals a similarly sparse latent space and achieve similarly low reconstruction residuals. We use these properties to design our classification and our anomaly detection strategies.

For classification, we propose an approach similar to dictionary learning, which consists of training one autoencoder per class and assigning to a sample the class corresponding to the autoencoder that led to the minimal loss of the objective function in Eq. **2**. This classification approach is common in signal processing when signal models are learned or trained to capture very specific properties of a class (17, 33).

For anomaly detection, we state that anomalies can be broadly separated into two types: local intermittent anomalies in the signal (an abnormal pulse due to an impact, e.g., a broken tooth in a gear box) and trend anomalies, when the signal behavior changes more globally (e.g., a change of frequency due to increased friction). When dealing with long signals (several seconds to minutes), capturing trend anomalies is usually achieved by looking at global indicators, such as the network objective function. However, such approaches tend to hide local anomalies, which are averaged out when the signal length increases. One possible solution can be shortening of the input time series to mitigate the impact of the average. However, this solution is not always practical. Here, we propose instead to jointly use averaged and local statistics. More precisely, per time series we propose to use the average residual (*Res*), the maximum residual (*MaxRes*), the average sparsity loss (average ℓ_1 norm of the coefficients per level, $\{\hat{\ell}_1^l\}_{l\in[1..L]}$), and the maximum sparsity loss ($\{\bar{\ell}_1^l\}_{l\in[1..L]}$). We then apply a one-class classifier on this new latent space, $Res \circ MaxRes \circ \{\hat{\ell}_1^l\} \circ \{\bar{\ell}_1^l\}_{l\in[1..L]}$, of size $(2 + 2 \cdot L)$.

In this work, we achieved similar results using the one-class isolation forest, the one-class support vector machine (SVM), an elliptic envelope, or a one-class extreme-learning machine (ELM) (34). We, therefore, report only results using the latter, an ELM with 50 neurons.

C. Datasets. We test the proposed approach on two open-source sound datasets. First, we test the model on an anomaly detection task of sound data of industrial machines with the sound dataset for malfunctioning industrial machine investigation and inspection (MIMII) (35). Second, we demonstrate that the network can learn decomposition specific to its training data by solving a dictionary learning classification task. We show that the approach performs equally well independently of the type of sound used as input to perform classification on the bird song dataset, as proposed in ref. 17. Finally, we show the consistency of the results by using this same dataset for anomaly detection. For both datasets, we analyze the obtained latent space and demonstrate that it can also be used to interpret the data at hand.

C.1. Malfunctioning industrial machine detection. The MIMII dataset (35) consists of audio recordings of four types of industrial machines, i.e., valves, pumps, fans, and slide rails, in normal and malfunctioning states. It is, therefore, a good benchmark for testing anomaly detection approaches on sound data.

The dataset has four individual machines of four machine types. For each machine, sound from normal and abnormal

Table 1. Characteristics of the MIMII dataset

Type	ID	Normal	Abnormal	Operating conditions and type of anomalies
Valve	0	991	119	Open/close repeat with
	2	708	120	different timing. More
	4	1,000	120	than two kinds of
	6	992	120	contamination.
Pump	0	1,006	143	Suction from/discharge to a
	2	1,005	111	water pool. Leakage,
	4	702	100	contamination, clogging,
	6	1,036	102	etc.
Fan	0	1,011	407	Normal operation.
	2	1,016	359	Unbalanced, voltage
	4	1,033	348	change, clogging, etc.
	6	1,015	361	
Slide rail	0	1,068	356	Slide repeat at different
	2	1,068	267	speeds. Rail damage,
	4	534	178	loose belt, no grease, etc.
	6	534	89	

operating conditions has been recorded without further label on the operating state or on the faults, making the dataset very suitable for unsupervised anomaly detection. Each machine has been recorded under three different signal-to-noise ratio (SNR) setups (0, 6, and –6 dB), where the noise denotes background noise of other industrial processes. This results in 48 experiments on which anomaly detection can be performed. It is interesting to note that various anomaly types are collected and that several anomalies can influence the same machine in different recording samples. Table 1 gives an overview of the dataset, presenting the number of samples for each machine and the conditions of operation.

The data were recorded with an eight-channel microphone array, at 16 kHz and 16 bits resolution. Each sample is 10 s long, or 160,000 timestamps. With this, we can set L to 17. In accordance with the work in ref. 35, the data recorded by the first microphone only are used.

C.2. Bird song dataset. For the second experiment, we use a different type of sound data to demonstrate the performance of the proposed framework in a different context, especially since industrial machines often make repetitive noise that is rather easier to characterize. The recordings of bird songs in their natural environment are also subject to environmental noise and differences in the recording hardware that influence the recorded sound. Since the data are labeled (contrary to the machine sound data), it allows us to test the proposed architecture both in an anomaly detection and in a classification setup.

The Xeno-canto Foundation collection bird song dataset (36) is a dataset of bird songs from all around the world that are collected by a large variety of participants. In ref. 17, the author proposes to focus on the following birds: corn bunting (CB), Eurasian skylark (ES), barn swallow (BS), sedge warbler (SW), and common nightingale (CN). These species were selected under the argument that all their recordings were recorded by the same person, implying similar recording conditions and probably similar and consistent hardware. This allows removing the hypothesis that detected fluctuations between recordings and bird species would be due to a change of recording hardware.

For each of the above species, three recordings of about 5 min are available. To establish a fair comparison, we apply the exact same preprocessing as in ref. 17: decimation of the signals by a factor of 4 since most of the signal energy is below 5 kHz and the original sampling rate is 41 kHz. The recordings are split into a collection of signals with 2^{18} samples ($\approx$ 24 s). This leads to the following number of signals:

- CB, 30 signals;
- ES, 24 signals;
- BS, 21 signals;
- SW, 20 signals;
- CN, 20 signals.

As in ref. 17, a fivefold cross-validation is used, meaning that 80% of the data are used for training and 20% for testing at each fold.

D. Ablation Study.

D.1. From FDWT to DeSpaWN. In this section, we aim to analyze how the different contributions impact the results compared to the traditional case where the coefficients from the FDWT would be used as inputs to subsequent machine-learning tasks, such as classification or anomaly detection. Thus, in this first evaluation, to demonstrate how all the steps of the transition from the traditional FDWT to our proposed framework impact the results, we compare the results for the following architectures:

- *db4* : Using the Daubechies-4 (db4) wavelets.
- *db4*+HT: Using the db4 wavelets with learnable hard thresholding (HT) of the coefficients.
- CWN (CQF wavelet network): Learning a single kernel h^0; using CQF to fix the other three g^0, $\bar{h}^0$, and $\bar{g}^0$; and using these kernels for all levels.
- LCWN (layer-wise CQF): Learning one kernel h^l per level in L, using CQF to fix the others.
- DeCWN (denoising CQF): Learning a single kernel h^0, using CQF to fix the other for all levels, and using the learnable hard-thresholding activation function.
- DeSpaWN: Learning one kernel h^l per level in L, using CQF to fix the others, and using the learnable hard-thresholding activation function.
- DeSpaWN-2: Learning two kernels, h^l and g^l, per level in L, using CQF to fix the others, and using the learnable hard-thresholding activation function.
- FreeWN: Learning all kernels of all levels independently and using the learnable hard-thresholding activation function.

For all experiments, we set L, the number of decomposition levels to the nearest second logarithm of the length of the time series. We set the kernel size to 8 and compare the results with those achieved using Daubechies db4 wavelets (since they also have eight coefficients) and with the baseline results on these datasets. In Eq. **2**, we set γ, the weight on the sparsity term in the loss arbitrarily to one. The architecture has, therefore, $(8+2)\cdot L$ learnable parameters (eight kernel coefficients, two thresholds).

For reference, we also report results from the baseline models [MIMII (35) and bird song (17)].

D.2. Results and discussion. From the comparative results presented in Table 2, it appears that the results are consistent between the three machine-learning tasks. Consequently, the impact of the different parameters and assumptions can be discussed at the general level.

DeSpaWN outperforms the baseline. On all tasks, the proposed architecture DeSpaWN significantly outperforms the baseline models found in the literature. Particularly on the MIMII dataset, it reaches globally a performance improvement of 16%. Also, compared to the baseline, DeSpaWN is much less impacted by the noise level; when the SNR changes from 6 to 0 dB, DeSpaWN experiences a drop of performance of 4% while the baseline has a 9% drop. When the noise level increases to a SNR at –6 dB, DeSpaWN performance diminishes while remaining well above the baseline (+16%). This suggests that DeSpaWN is learning a noise-independent representation of the signals, which makes it much more robust to noise than other approaches. This is a

Table 2. Comparative study on three machine-learning tasks: For the different architecture variations (one per column), comparative results on the three considered tasks

	DeSpaWN										
	$\gamma = 1$	$\gamma = 0.5$	$\gamma = 5$	*db4*	*db4*+HT	CWN	LCWN	DeCWN	DeSpaWN-2	FreeWN	Baseline
Anomaly Detection on MIMII											
Valve	92.8	92.7	91.0	92.7	92.7	92.7	92.7	92.9	**93.0**	**93.0**	61.3
Pump	**84.5**	82.0	72.6	77.9	78.3	78.2	78.1	78.0	84.2	75.8	72.3
Fan	**86.2**	84.8	84.9	84.6	84.7	84.9	84.8	85.3	85.5	83.8	79.0
Slider	**91.0**	89.4	78.7	89.8	90.0	89.4	89.6	89.7	90.1	89.3	78.6
6 dB	**94.6**	93.5	84.4	93.4	93.4	93.7	93.5	93.5	94.3	90.7	81.6
0 dB	**90.5**	87.7	81.9	82.6	88.1	87.7	87.8	88.5	90.2	87.0	72.3
-6 dB	80.8	79.6	76.5	77.6	77.9	77.5	77.5	77.5	80.0	78.8	64.4
Avg.	**88.6**	87.1	81.4	85.5	86.4	86.3	86.3	86.5	88.2	85.5	72.8
Anomaly detection (1 versus 4 bird species)											
Avg.	**99.8**	**99.8**	91.7	95.4	98.2	97.5	98.8	99.0	99.5	99.1	N.A.
#C: Classification with dictionary learning (bird song)											
2	**99.2**	98.2	91.3	50.0	87.0	73.3	92.7	93.4	99.1	89.0	97.2
3	**98.3**	96.3	88.7	33.3	77.3	55.1	86.0	87.4	97.5	78.0	88.0
4	**97.5**	94.5	87.0	25.0	70.0	43.5	80.0	81.9	97.3	67.0	74.7
5	**96.7**	92.7	85.3	20.0	64.0	37.0	74.7	77.0	95.6	56.0	70.4
Avg.	**97.9**	95.4	88.1	32.1	74.6	52.2	83.3	84.9	97.4	72.5	82.6

For the anomaly detection tasks, on MIMII and on the bird song, we report the average AUC (%). Finally, for the bird song classification by dictionary learning, we report the average accuracy (%). N.A., not applicable. Bold indicates best model for each experimental setup.

particularly important requirement in real applications that are typically impacted by different types of noise at different levels. Finally, in the baseline, the spectrogram with logarithmic mel scale (log-mel) is extracted to be used as input to an autoencoder. With DeSpaWN, the raw data are used directly, without requiring any preprocessing steps. This lightens the methodology significantly since extracting a spectrogram requires the choice of several hyperparameters such as the window type, the window length, the window stride, whether to compute the density or the magnitude, and whether to apply additional transformations (log-mel, decibels, etc.). All these choices can influence the results significantly.

Impact of the sparsity coefficient. From the first three columns of Table 2, it appears that the sparsity term in the loss of DeSpaWN (Eq. **2**) influences the results. In addition, setting γ to one, without fine-tuning, always seems to be a near optimal choice. This can be explained by our definition of the loss as the average of the ℓ_1 values, first of the residuals, and second of the coefficients moduli. Using the average makes them comparable. Using smaller γ influences the results slightly. However, increasing γ can have quite a strong impact on the performance. This signifies that even though sparsity is helping to get some robustness to external factors, too much of it would be at the expense of the reconstruction loss and at the expense of the ability to distinguish variations in the signals, including anomalies or class-specific coefficient behaviors.

Impact of hard-threshold learning. The second noteworthy observation is that the architectures without hard threshold are performing significantly worse compared to others (*db4* versus *db4*+HT or LCWN versus DeSpaWN). This highlights the importance of the denoising part of the architectures. The strength of the architecture is its ability to learn the best thresholds for the wavelet coefficients to become robust to small variations in the signal. This strength is independent of whether the wavelets are learned or not.

Impact of wavelet learning. For the classification task in particular, learning the right wavelet is pivotal for the architecture accuracy. This is to be expected since the class attribution is done based on the network loss minimization. When fixing the wavelets or even both the wavelets and the thresholding function (*db4* and *db4*+HT), there are not many parameters left to optimize. The architectures become generic and not fine-tuned per class, making the class attribution based on the loss not much better than a random guess (*db4* has random guess attribution since all architectures are identical). For anomaly detection, this effect is mitigated by the property of the wavelets: Due to the CQF property, wavelets are designed for good reconstruction and relative sparsity (wavelets are used in signal processing since they inherently tend to produce sparse signal representations). Hence, they are already good candidates to create relevant signal description and, thus, anomaly detection. Yet, learning the right wavelets still brings some nonnegligible improvements (additional +1 or 2%, averaged over several tens of experiments).

Impact of the CQF. The impact of constraining the wavelet basis can be observed by comparing *db4*+HT (fixed Daubechies-4 wavelets), the DeCWN (learning of one global wavelet), the DeSpaWN (learning one wavelet per layer), the DeSpaWN-2 (learning one wavelet and one scaling function per layer), and the FreeWN (learning all kernels). As expected, constraining the kernels tends to make the network less specific to the characteristics of the training class and affects the classification performance strongly. The most constrained architecture (*db4*+HT) has the worst results (52%); DeCWN performs better (84.9%) but not as good as DeSpaWN or DeSpaWN-2 (97.9 and 97.4%). These two architectures are in fact quite equivalent in terms of results. Leaving the kernel completely free (FreeWN) also leads to a drop in performance. This is likely due to training instabilities as explained in *2. Learnable Denoising Sparse Wavelet Network*.

E. Additional Diagnostics Potential.

E.1. Insights on MIMII. The good detection performance of DeSpaWN indicates again that the signals are probably forming well-defined clusters in the $Res \circ MaxRes \circ \{\hat{\ell}_1^l\} \circ \{\bar{\ell}_1^l\}_{l\in[1..L]}$ latent space. Anomalies would then appear outside of these clusters and finding anomaly clusters could help to diagnose the different conditions of the system. This can be visualized in two dimensions, by performing a t-distributed stochastic neighbor

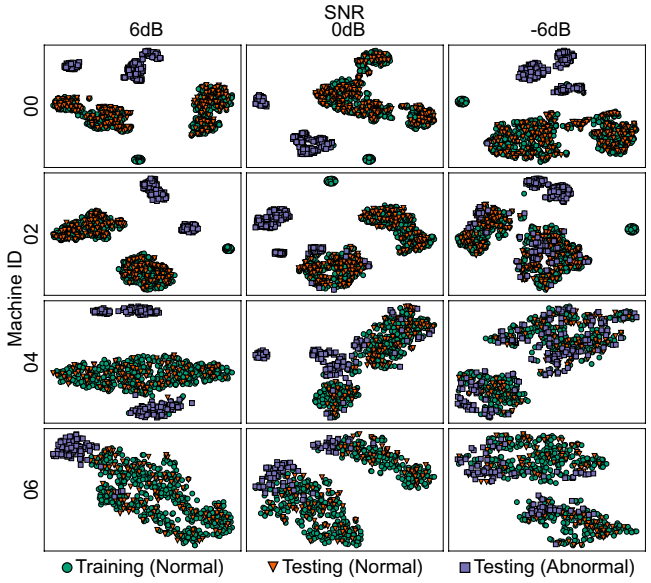


Fig. 3. t-SNE of the slide rail latent space. Shown is representation in two dimensions of the slide rail latent space for the four machines (one machine per row), at the three different experimental SNRs (columns), using t-SNE (perplexity of 30). One can distinguish different clusters, most likely representing different operating conditions and anomalies.

embedding (t-SNE) on the latent space, as illustrated in Fig. 3. In Fig. 3, the latent spaces of the different slide rail experiments are shown after a t-SNE transformation with the default perplexity of 30. Clusters can be clearly identified in this representation. They are likely to be formed by different anomaly types and operating conditions. In all experiments, one can distinguish at least two normal operating condition clusters, indicating different conditions of operation and at least two anomaly clusters, well separated from the normal conditions. With this representation, only when the SNR decreases to –6 dB, some of the anomalous points become less separated from the normal conditions. This decrease in separability could be expected from the lower area under the curve (AUC) 4 as reported in Table 2.

The diagnostics possibilities offered by extracting characteristic patterns of the learned coefficients of the proposed approach for the different clusters analysis are illustrated for the slide rail 0 at 0 dB SNR in Fig. 4. Two different signals extracted from each of the two normal measurement clusters are shown. In Fig. 4, the raw signals, their log-spectrogram, and the distribution of the learned coefficients over the 18 levels are shown. The first signal has its major components around the 10th level with some quite large coefficients at the highest level of detail, while the other signal is mostly "active" at the 12th level, with very little activation of the last two levels (highest level of details). This indicates that the operating mode has likely changed between these two samples: The main information content changed from level 10 to 12, that is, a factor of 4 in the spectrum of the original signal. Similarly, in Fig. 5, two unhealthy signals of the slider rail drawn from two different clusters are depicted. The first signal has its 12th level more activated than healthy signals; the other signal distinguishes itself with its much larger high-level coefficients. These are likely two different types of anomalies.

Finally, Fig. 6 shows exemplarily the learned filters and hard-threshold coefficients for a slider rail. The first layers, corresponding to the high frequencies, have high hard-thresholding values (up to 0.3 in absolute value). One can observe that the corresponding wavelet coefficients in Fig. 4 are almost all zero. This makes the filter of these layers irrelevant and this further explains why the corresponding filters observed in Fig. 6 are very close to the original Daubechie filters. It is probably where most of the noise is concentrated. For lower frequencies, the filters are farther away from the Daubechie wavelets and the hard thresholding is much lower. It is probably at these scales that the information required for the reconstruction is concentrated. This interpretation matches the coefficient distributions observed in Fig. 4. This gives a strong indication that the proposed architecture can indeed selectively filter and threshold the layers based on their information content.

E.2. Insights on the bird song dataset. Fig. 7 illustrates the $l1$ -residual space for all birds for the different cases. For each plot, DeSpaWN is trained on the main bird class where all other bird samples would have to be detected as anomalies. In each case, one can observe that the samples corresponding to the bird used for training form a well-defined and separated cluster, allowing the one-class classifier to identify easily all other birds

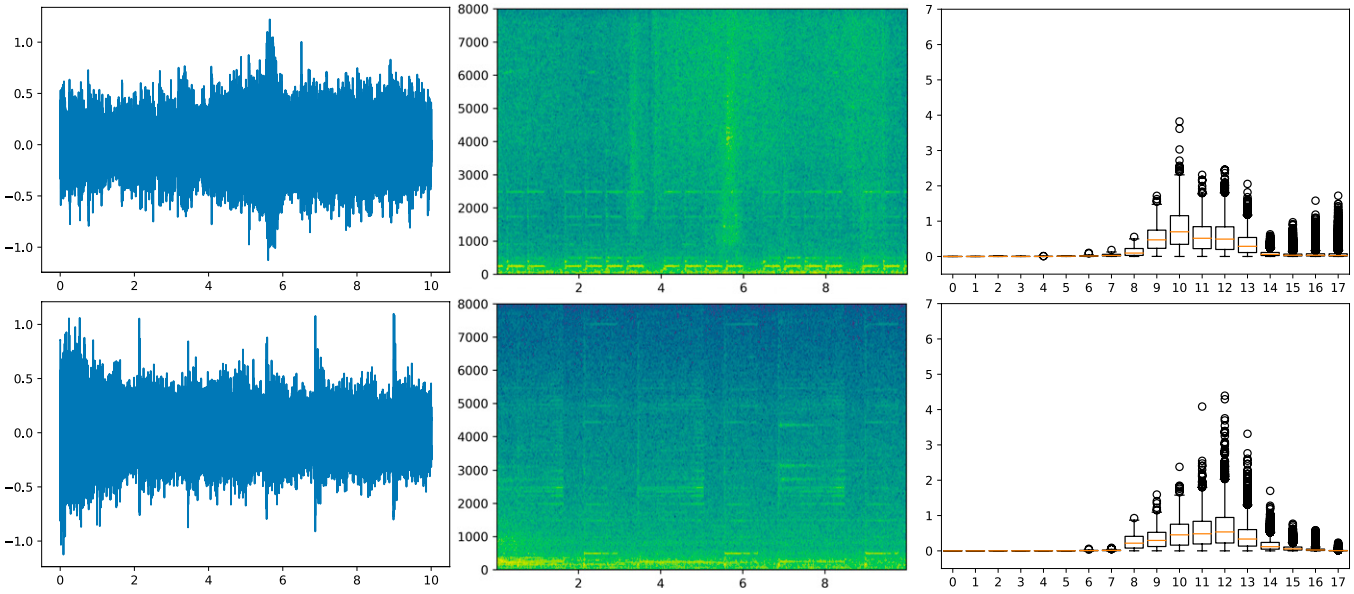


Fig. 4. Examples of normal signals. Shown are raw data, log-spectrogram, and obtained coefficient distribution per level for two normal measurements of the slider rail 0 at SNR 0 dB.

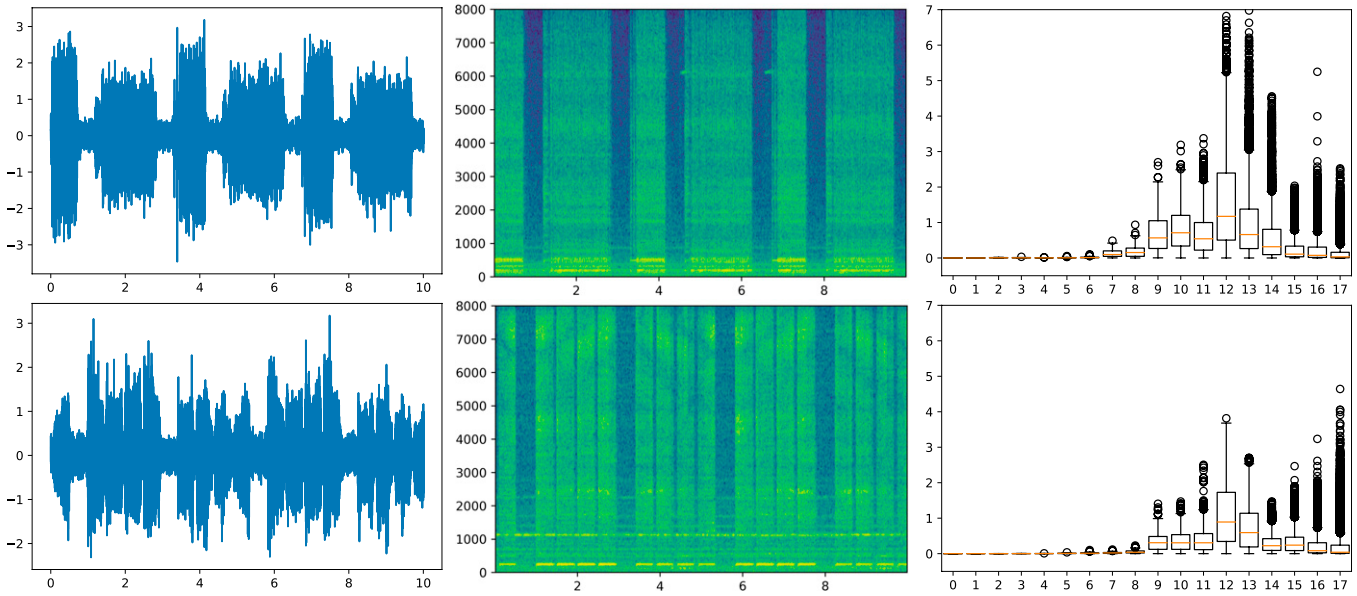


Fig. 5. Examples of abnormal signals. Shown are raw data, log-spectrogram, and obtained coefficient distribution per level for two abnormal measurements of the slider rail 0 at SNR 0 dB.

as anomalies. The only exception is the corn bunting case (Fig. 7, *Right*), which is mixed with the sedge warbler after training, particularly with this specific representation. These results explain the classification accuracy drop observed when adding the sedge warbler as the fifth species (Table 2).

4. Comparison to Other Frameworks

A. Scattering Transform, U-Net, and Autoencoders. In this section, we propose to compare the results of DeSpaWN to other state-of-the art frameworks found in the literature: the scattering transform, vanilla convolutional autoencoders (CAE), and U-Net.

First, we propose to compare to the scattering transform (21), which has been extensively used in the context of audio signal processing. It is a signal representation that is stable to small deviations in its inputs and is able to characterize transient phenomena like amplitude modulation. We use the Kymatio library (37) to compute the coefficients from a two-layer scattering transform. It requires the selection of two parameters (38): J the maximum scale of the filters used, implying that the transform will capture only frequencies superior to 2^J, and Q the number of filters per octave. We propose to analyze two combinations of parameters: 1) $J = 17$ and $Q = 1$, which will result, for the first layer, in a decomposition close to the FDWT with one filter per octave and able to characterize low frequencies up to 2^{-17}, and 2) $J = 10$ and $Q = 8$ for the first layer (which defines wavelets having the same frequency resolution as mel-frequency filters) and $Q = 1$ for the second layer. The second choice of parameters is motivated by previous research that proposed to consider mel-spectrogram features on frames of around 60 ms (35). Similar to the approach used with DeSpaWN, for the anomaly detection task, we use a one-class ELM on the scattering coefficients. For the bird song classification task, since the scattering transform does not create a signal-specific decomposition, the dictionary learning approach cannot be mimicked. Therefore, we use, for all approaches, the coefficients as input to an SVM classifier to make the results comparable.

In addition, to highlight the relevance of the proposed architectural choice of DesPaWN, we compare it numerically to standard CNN autoencoders (CAE) and CNN U-Nets. The considered autoencoders (AE) are based on the work of ref. 39. The architecture has been used for fault diagnosis of rotating machinery. We consider four encoding and decoding layers with eight coefficients per kernel and a kernel size of N_{AE} for each layer. The impact of the addition of trainable parameters is studied by considering the range of $N_{AE} = [4, 8, 16, 32]$. DeSpaWN architecture has skip-connections at each level. Therefore, it can be considered as a special case of a U-Net model. We then compare our method to another U-Net architecture based on ref. 22 that was applied to electrocardiogram detection. We replace the concatenation of the skipped connection with the addition to

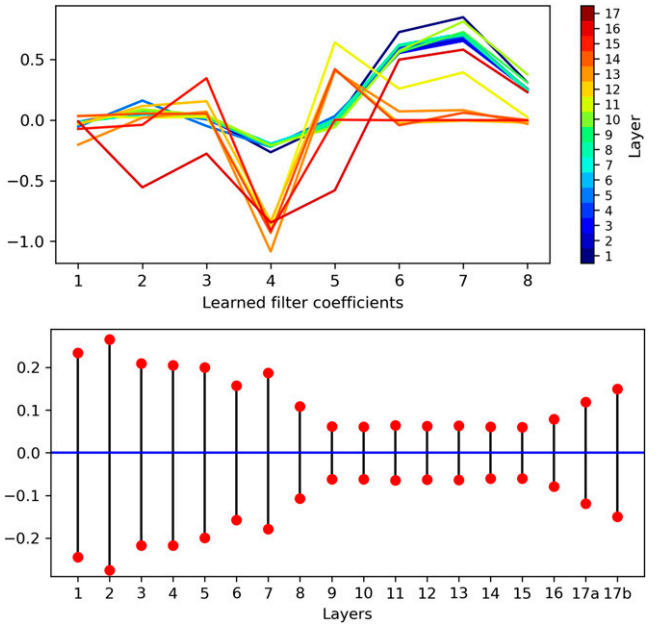


Fig. 6. Learned coefficients and biases. (*Top*) Learned kernel for all layers of a slider rail. Each color represents the filter from a different layer, from the first (high-frequency) layer in dark blue to the last layers in red. (*Bottom*) Learned positive and negative biases for each layer. The black lines represent the range of values that are set to zero by the corresponding hard-thresholding layer.

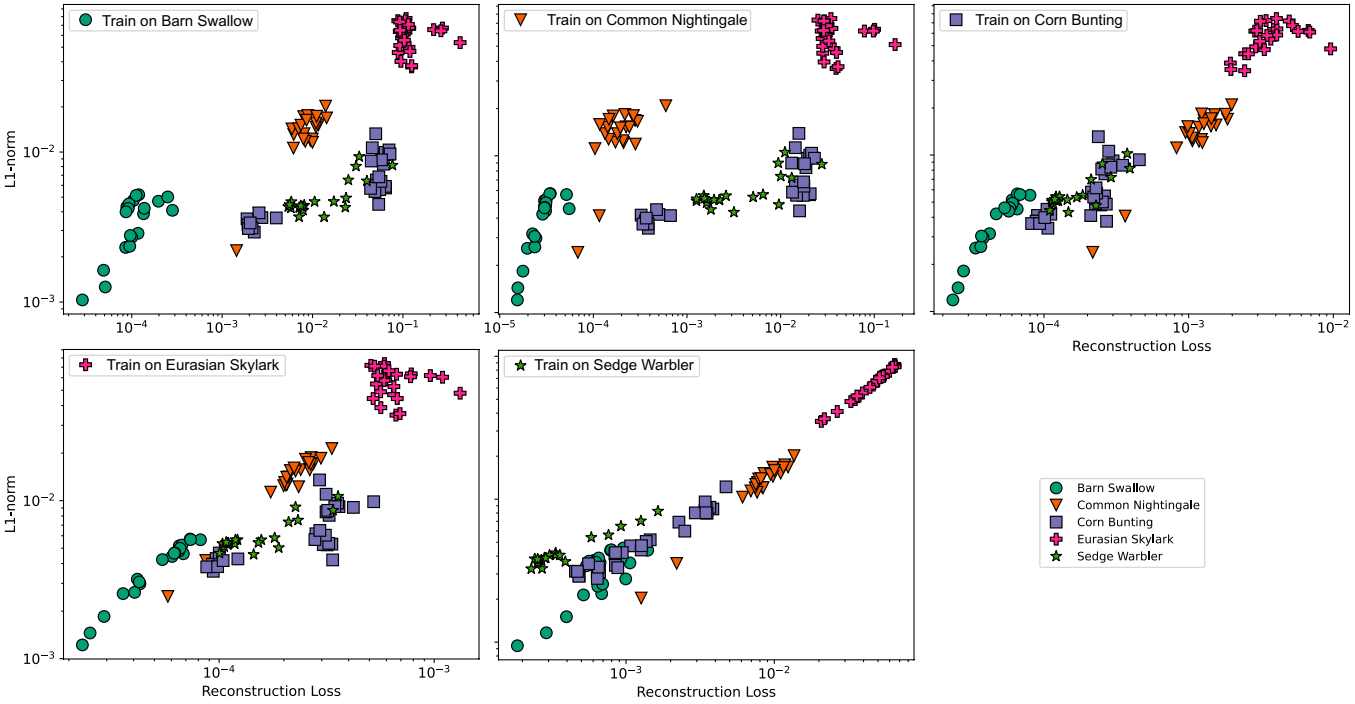


Fig. 7. *Res* $\circ \hat{\ell}_1$ space. For each bird, DeSpaWN is trained and all birds are tested and plotted together. When DeSpaWN is trained on one species, the species tends to be well separated from others and has generally a more compact representation in the *Res* $\circ \hat{\ell}_1$ than when tested with another DeSpaWN.

have an architecture closer to DesPaWN. Furthermore, we consider $L = 17$ layers with a stride of 2 for each CNN layer and eight coefficients per kernel. The kernel size N_{Unet} of each filter is studied in the range $N_{Unet} = [1, 2, 4, 8]$. This grid was chosen since larger kernel sizes led to decreasing performances. The main differences between our model and this U-Net architecture are one more filter at each skip connection, initialization of each pair of filters as high-pass and low-pass, and replacing the rectified linear unit (ReLU) activation function with the proposed learnable hard-thresholding function. For each method, the same loss function as in Eq. **2** is used for training. The exact same process is followed for anomaly detection and for classification as for the results achieved with DeSpaWN (cf. *B. Results of the Benchmark*).

B. Results of the Benchmark. The results of the benchmark are presented in Table 3, for two versions of the scattering transform, four variations of the CAE, and four variations of the U-Net. DeSpaWN and the scattering transform both provide very competitive results and are both very solid candidates to solve the tasks on the dataset studied here. One can note, however, a bigger drop in performance for the scattering transform when the noise in the data increases (–20% and –16% when comparing MIMII 6 dB with –6 dB). It is also worth noting that the strength of each approach depends on the machine type, where the scattering transform seems to be the most adapted to tackle the pump and the slider while DeSpaWN performs better on the valve and on the fan. For the bird song classification, very similar results are achieved.

The autoencoders, both the traditional CAE and the U-Nets, provide very competitive results for the valve system but not for the other machines and are overall significantly worse performing compared to the scattering transform and to DeSpaWN. It is worth noting, however, that all approaches outperform the reported baseline on these datasets.

5. Conclusion

In this paper, we proposed an architecture for learning a meaningful and sparse representation of high-frequency signals in an unsupervised manner without requiring neither preprocessing (feature extraction) nor postprocessing (e.g., denoising). This architecture achieves very good results that are well above the baselines and are competitive compared to other state-of-the-art approaches on three machine-learning tasks for anomaly detection and classification. We designed an end-to-end deep-learning architecture, mimicking the cascade algorithm of the FDWT but making it fully learnable. Using the deep-learning framework, we demonstrated the benefits of learning the right wavelets at each level of the decomposition. One of the additional contributions is the introduction of a learnable hard-thresholding function for automatic signal denoising.

The proposed methodology combines 1) a thorough theoretical foundation on the wavelet properties, including cascade, perfect reconstruction, and antialiasing filter basis with the CQF property, and denoising with coefficient thresholding with 2) the learning ability of deep learning. The proposed architecture could demonstrate a significant improvement on sound data processing, both for classification and for anomaly detection tasks. Our approach allows the use of the raw HF data as input to a deep-learning architecture, a setup usually avoided in the literature due to the difficulty of designing efficient architectures that are robust to changes in the input lengths. The proposed architecture takes root in spectral analysis and can replace the usual preprocessing steps such as spectrogram or wavelet coefficient extraction. Since it is unsupervised, it can be used as an input to subsequent learning methods. In addition, compared to other deep-learning architectures, it is a very light framework with only a few hundred learnable parameters, mitigating in that way the high risk of overfitting. With its spectral interpretation, it also provides diagnostics information to the domain experts that can potentially improve the interpretation capabilities.

This work opens several doors for future directions. First, given the high information content of the proposed latent space, other unsupervised machine-learning tasks could be explored such as system degradation monitoring; e.g., a drop in the sparsity of the decomposition could be a sign of an increased signal complexity

Table 3. Comparative study on three machine-learning tasks: For the different methodologies (one per column), comparative results on the three considered tasks

	DeSpaWN	Scattering transform		Auto Encoder				U-Net			
	γ 1	(J, Q) (17, 1)	(J, Q) (10, 8)	N_{AE} 4	N_{AE} 8	N_{AE} 16	N_{AE} 32	N_{Unet} 1	N_{Unet} 2	N_{Unet} 4	N_{Unet} 8
Anomaly detection on MIMII											
Valve	92.8	65.5	78.6	91.5	92.4	91.5	92.0	92.3	92.7	**93.0**	91.1
Pump	84.5	88.5	**90.6**	74.5	74.5	74.0	73.4	75.4	69.6	70.5	74.4
Fan	86.2	**88.7**	84.9	71.0	76.6	80.7	76.1	77.1	75.6	76.5	73.6
Slider	91.0	86.9	**96.8**	79.2	81.0	81.7	78.1	78.6	80.6	83	87.4
6 dB	94.6	91.2	**95.4**	83.4	87.5	86.6	85.0	86.3	84.0	84.7	85.2
0 dB	**90.5**	84	88.6	79.7	81.3	81.5	80.8	82.5	79.8	83.0	83.8
–6 dB	**80.8**	72	79.1	74.2	74.6	73.9	74.0	74.0	75.0	74.0	75.0
Avg.	**88.6**	82.4	87.7	79.1	81.15	80.3	80.0	81.0	79.6	80.9	81.4
Anomaly detection (1 versus 4 bird species)											
Avg.	**98.6**	93.8	94.0	85.4	86.1	86.3	86.3	89.6	88.9	89.2	90.0
#C: Classification with SVM (bird song)											
5	97.7	**97.9**	97.3	87.6	92.2	92.3	90.2	83.4	85.7	87.1	88.4

For the anomaly detection tasks, on MIMII and on the bird song, we report the average AUC (%). Bold indicates best model for each experimental setup.

or of the presence of disturbing components due to system wear. The architecture could be further analyzed in conditions with controlled noise and signals to better understand its denoising and stability properties. The architecture could also be extended, such as in particular with the imbrication of this architecture in stacked architecture to solve supervised machine-learning tasks. In these cases, the learned wavelets and thresholding coefficients could be learned not only for sparsity and for reconstruction but also to maximize a supervised objective. The use of parallel wavelets (number of kernels in a convolution layer) and the handling of multichannel inputs are further exciting potential developments.

Data and Code Availability. All study data are included in the main text. Our code is available at GitHub, https://github.com/MichauGabriel/DeSpaWN. Previously published data were used for this work (https://zenodo.org/record/3384388#.YG1g150zaUk https://archive.org/details/xccoverbl_2014).

ACKNOWLEDGMENTS. This work was supported by the Swiss National Science Foundation Grant PP00P2-176878 and by the Innosuisse Grant 27662.1 PFES-ES. We thank Christoph Preisinger for his preliminary explorations of the proposed methodology.

1. Z. Peng, F. Chu, Application of the wavelet transform in machine condition monitoring and fault diagnostics: A review with bibliography. *Mech. Syst. Signal Process.* **18**, 199–221 (2004).
2. P. Gangsar, R. Tiwari, Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review. *Mech. Syst. Signal Process.* **144**, 106908 (2020).
3. Q. Wang, G. Michau, O. Fink, Missing-class-robust domain adaptation by unilateral alignment. *IEEE Trans. Ind. Electron.* **68**, 663–671 (2020).
4. Z. Li, Y. Wang, K. Wang, A deep learning driven method for fault classification and degradation assessment in mechanical equipment. *Comput. Ind.* **104**, 1–10 (2019).
5. H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P. A. Muller, Deep learning for time series classification: A review. *Data Min. Knowl. Discov.* **33**, 917–963 (2019).
6. H. Wang, Q. Zhang, J. Wu, S. Pan, Y. Chen, Time series feature learning with labeled and unlabeled data. *Pattern Recognit.* **89**, 55–66 (2019).
7. O. Fink *et al.*, Potential, challenges and future directions for deep learning in prognostics and health management applications. *Eng. Appl. Artif. Intell.* **92**, 103678 (2020).
8. G. Michau, O. Fink, Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer. *Knowl. Base. Syst.* **216**, 106816 (2021).
9. S. Kiranyaz *et al.*, 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **151**, 107398 (2021).
10. W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **100**, 439–453 (2018).
11. V. Papyan, Y. Romano, M. Elad, Convolutional neural networks analyzed via convolutional sparse coding. *J. Mach. Learn. Res.* **18**, 2887–2938 (2017).
12. S. Liu, "Fourier neural network for machine learning" in *2013 International Conference on Machine Learning and Cybernetics* (IEEE, 2013) vol. 1, pp. 285–290.
13. M. Uteuliyeva *et al.*, Fourier neural networks: A comparative study. *Intell. Data Anal.* **24**, 1107–1120 (2020).
14. S. Luan, C. Chen, B. Zhang, J. Han, J. Liu, Gabor convolutional networks. *IEEE Trans. Image Process.* **27**, 4357–4366 (2018).
15. T. Li *et al.*, Waveletkernelnet : An interpretable deep neural network for industrial intelligent diagnosis. *IEEE Trans. Syst. Man Cybern. Syst.*, 10.1109/TSMC.2020.3048950 (2021).
16. J. Wang, Z. Wang, J. Li, J. Wu, "Multilevel wavelet decomposition network for interpretable time series analysis" in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, New York, 2018), pp. 2437–2446.
17. D. Recoskie, "Learning sparse orthogonal wavelet filters," PhD thesis, University of Waterloo, Waterloo, ON, Canada (2018).
18. P. Liu, H. Zhang, W. Lian, W. Zuo, Multi-level wavelet convolutional neural networks. *IEEE Access* **7**, 74973–74985 (2019).
19. M. Khalil *et al.*, An end-to-end multi-level wavelet convolutional neural networks for heart diseases diagnosis. *Neurocomputing* **417**, 187–201 (2020).
20. D. L. Donoho, I. M. Johnstone, "Threshold selection for wavelet shrinkage of noisy data" in *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 1994), vol. 1, pp. A24–A25.
21. J. Andén, S. Mallat, Deep scattering spectrum. *IEEE Trans. Signal Process.* **62**, 4114–4128 (2014).
22. G. Jimenez-Perez, A. Alcaine, O. Camara, "U-net architecture for the automatic detection and delineation of the electrocardiogram " in *2019 Computing in Cardiology (CinC)* (IEEE, 2019).
23. S. Mallat, *A Wavelet Tour of Signal Processing, The Sparse Way* (Elsevier Science/Academic Press, 3rd ed., 2009).
24. A. Croisier, "Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques" in *Proceedings of the International Symposium on Information, Circuits and Systems* (Proceedings of the International Symposium on Information Circuits and Systems, Patras, Greece, 1977), pp. 443–446.
25. M. Smith, T. Barnwell, "A procedure for designing exact reconstruction filter banks for tree-structured subband coders" in *ICASSP'84, IEEE International Conference on Acoustics, Speech, and Signal Processing* (IEEE, 1984), vol. 9, pp. 421–424.
26. F. Mintzer, Filters for distortion-free two-band multirate filter banks. *IEEE Trans. Acoust. Speech Signal Process.* **33**, 626–630 (1985).
27. M. Alfaouri, K. Daqrouq, ECG signal denoising by wavelet transform thresholding. *Am. J. Appl. Sci.* **5**, 276–281 (2008).
28. F. M. Bayer, A. J. Kozakevicius, R. J. Cintra, An iterative wavelet threshold for signal denoising. *Signal Processing* **162**, 10–20 (2019).
29. D. L. Donoho, "Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data" in *Proceedings of Symposia in Applied Mathematics* (American Mathematical Society, 1993), pp. 173–205.
30. H. Bolcskei, P. Grohs, G. Kutyniok, P. Petersen, Optimal approximation with sparsely connected deep neural networks. *SIAM J. Math Data Sci.* **1**, 8–45 (2019).
31. D. L. Donoho, J. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**, 425–455 (1994).
32. J. Liebetrau, S. Grollmisch, Predictive maintenance with airborne sound analysis. *Process. Mag.* **1**, 15587140 (2017).
33. J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, "Discriminative learned dictionaries for local image analysis" in *2008 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2008), pp. 1–8.
34. G. Michau, Y. Hu, T. Palmé, O. Fink, Feature learning for fault detection in high-dimensional condition monitoring signals. *Proc. Inst. Mech. Eng. O. J. Risk Reliab.* **234**, 104–115 (2020).
35. H. Purohit *et al.*, MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection. arXiv [Preprint] (2019). https://arxiv.org/abs/1909.09347 (Accessed 3 February 2022).
36. X canto Foundation, *Dataset of bird songs* (2004). https://archive.org/details/xccoverbl_2014. Accessed 10 January 2021.
37. M. Andreux *et al.*, Kymatio: Scattering transforms in python. *J. Mach. Learn. Res.* **21**, 1–6 (2020).
38. G. Destouet, C. Dumas, A. Frassati, V. Perrier, "Wavelet scattering transform and ensemble methods for side-channel analysis" in *International Workshop on Constructive Side-Channel Analysis and Secure Design* (Springer, Cham, Switzerland, 2021), pp. 71–89.
39. X. Liu, Q. Zhou, J. Zhao, H. Shen, X. Xiong, Fault diagnosis of rotating machinery under noisy environment conditions based on a 1-d convolutional autoencoder and 1-d convolutional neural network. *Sensors (Basel)* **19**, 972 (2019).