# Crossterm-Free Time-Frequency Representation Exploiting Deep Convolutional Neural Network

Shuimei Zhang, Md. Saidur Rahman Pavel, and Yimin D. Zhang

*Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19122, USA*

**Abstract**

Bilinear time-frequency (TF) analyses provide high-resolution time-varying frequency characterization of nonstationary signals. However, because of their bilinear natures, such TF representations (TFRs) suffer from crossterms. TF kernels, which amount to low-pass weighting or masking in the ambiguity function domain, are commonly used to reduce crossterms. However, existing fixed and adaptive kernels do not guarantee effective crossterm suppression and autoterm preservation, particularly for signals with overlapping autoterms and crossterms in the ambiguity function. In this paper, we develop a new method that offers high-resolution TFRs of nonstationary signals with desired autoterm preservation and crossterm mitigation capabilities, especially for signals with slowly time-varying instantaneous frequencies. The proposed method exploits a convolutional autoencoder network which is trained to construct crossterm-free TFRs. For the signals being considered, the proposed technique with properly trained networks offers the capability to outperform state-of-the-art TF analysis algorithms based on adaptive kernels and compressive sensing techniques.

*Keywords:* Crossterm mitigation, deep neural network, nonstationary signal, time-frequency analysis.

## 1. Introduction

Nonstationary signals are naturally observed in many real-world applications, such as radar, sonar, satellite navigation, seismology, and biomedical applications [1–11]. One important class of nonstationary signals is referred to as frequency-modulated (FM) signals that are characterized by

time-varying instantaneous frequencies (IFs). For such signals, joint time-frequency (TF) domain representations are most suited for their analyses and classification as they effectively provide time-varying spectra along the true signal IFs.

TF representations (TFRs) can be generally classified into linear and bilinear. For example, short-time Fourier transform is a commonly used linear TFR approach whose TF resolution is restricted to a fixed sliding time window. Compared to linear counterparts, bilinear TFRs generally provide higher TF concentration. However, due to the nonlinearity of the bilinear TFRs, crossterms unavoidably appear in between signal autoterm components in the case of nonlinear or multi-component FM signals. Heavy presence of crossterms prohibits accurate signal analysis and estimation of signal IF signatures [6].

The Wigner-Ville distribution (WVD) is commonly referred to as the prototype bilinear TFR with a high impact of crossterms. Various TF kernels have been developed to suppress crossterms while preserving autoterms [4, 6, 12, 13]. Essentially, a TF kernel acts as a two-dimensional (2-D) low-pass filter or mask multiplied in the ambiguity function (AF) domain, expressed with respect to time lag and frequency shift. TF kernels are designed based on the fact that, typically, autoterms have high values around the origin of the AF domain whereas crossterms tend to be scattered away from the origin. Because the AF and the TF domains are associated by a 2-D Fourier transform relationship, a kernel effectively becomes a 2-D convolution in the TF domain. Because autoterm preservation and crossterm mitigation are conflicting objectives, designing TF kernels that meet both objectives has been a challenging task in the past several decades and motivated the development of a high number of TF kernels. The shape and extent of the 2-D smoothing kernel can be predetermined (fixed kernel) or optimized (data-dependent or adaptive kernel). Compared to fixed TF kernels, adaptive TF kernels, such as the adaptive optimal kernel (AOK) [12], provide data-dependent optimization and thus generally yield better performance.

While all existing fixed and adaptive kernels utilize certain properties to mitigate crossterms while preserving autoterms, there is no assurance to achieve these two conflicting objectives. Because existing TF kernels do not have the capability to explicitly distinguish crossterms from autoterms, there is no guarantee that crossterms are effectively mitigated and autoterms are preserved. In particular, existing TF kernel designs generally assume that crossterms are well separated from signal autoterms in the AF domain. When

such assumption is not satisfied, e.g., autoterms and crossterms highly overlap, TF kernels would compromise one or both of their desired objectives. Inseparable autoterms and crossterms are commonly observed when signals involve highly nonlinear FM and intersecting FM components. Another problem associated with adaptive kernels is that the rendered kernels are highly impacted by strong signal components. As a result, multiplicative kernels further weaken weak signals, making them more easily obscured by strong ones.

A recent trend to achieve high-resolution TF analyses is through the utilization of the sparsity of FM signals in the TF domain. Depending on the domain of observations, the incorporation of TF sparsity has been implemented in two ways. The first class of approaches utilizes the 2-D Fourier transform relationship between the AF and the TF domains [14–17]. In these methods, a proper mask around the AF origin is selected to mitigate the effects of crossterms. The desired TFR is then found as the sparse solution of the vectorized TFR entries. On the other hand, the second class of approaches is based on the one-dimension (1-D) Fourier transform relationship that relates the instantaneous autocorrelation function (IAF) and the TF domains, and sparsity-based TFRs are obtained by performing compressive sensing on the prototype or kernelled IAF for each time instant [18–20]. In this case, TF kernels can still be applied either directly in the IAF domain or by converting kernelled AF to the IAF domain through 1-D Fourier transform. By using IAF domain observations to perform sparse TFR reconstruction, the second class of approaches offers multi-fold merits, including reduced complexity because the 1-D Fourier transform relationship between the IAF and the TF domains requires a much smaller dictionary matrix, insensitivity to time-dependent signal fluctuation or fading because the sparsity is considered locally for each time instant, and the capability to account for the continuity of frequency signatures over closely spaced time samples [21].

Recently, deep learning techniques [22] have achieved great success in many applications, such as image recognition [23], speech recognition [24], electroencephalogram (EEG) interpretation [25], crack detection [26], human motion recognition [27], and spectral recovery [28]. In the area of TF analysis, Jiang *et al* [17] recently developed a U-Net aided iterative shrinkage-thresholding algorithm to learn the structural sparsity in the TF domain. However, its TFR reconstruction performance is still restricted by the selection of the AF samples.

In [29], the concept of training a deep neural network (DNN) to achieve autoterm preservation and crossterm mitigation was briefly described, and fully convolutional neural network (FCNN) and convolutional autoencoder (CAE) were exploited and compared. It is shown that the CAE may achieve comparable performance as the FCNN using less layers, and the use of max-pooling further reduces the computational complexity. In this paper, we will further explore this concept to obtain high-resolution crossterm-free TFRs. Because of the clear advantage of CAE over FCNN as indicated in [29], only the CAE is considered in this paper. The CAE is trained to provide high-resolution autoterms while completely suppress crossterms, even when autoterms and crossterms highly overlap in the AF domain.

In this paper, ideal TF model constructed based on true signal IFs is served as the training label. The convolutional layers capture the abstraction of the autoterms while eliminating crossterms. Deconvolutional layers are used to upsample the feature maps and recover the autoterms details. Unlike TF kernel designs which are optimized based on the signal characteristics observed in the TF or AF domain, the proposed method offers optimized end-to-end learning with loss function minimized at the network output. Therefore, it provides significant improvement in the TFR reconstruction performance and it works well even when autoterms and crossterms highly overlap in the AF. It is worth noting that desired performance relies on adequately trained networks, and the performance may degrade when the test signal deviates from the training dataset. The superiority of the proposed method is demonstrated for a set of examples which represent different nonstationary signals with slowly time-varying IFs. Developing more comprehensive training strategies is out of the scope of this paper.

*Notations*: Lower-case (upper-case) bold characters are used to denote vectors (matrices). $(\cdot)^*$, $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ denote complex conjugation, transpose and the Hermitian transpose, respectively. $\mathcal{F}_s(\cdot)$ represents the discrete Fourier transform (DFT) with respect to $x$. $\mathrm{Re}(\cdot)$ represents the real part of a complex value, $\|\cdot\|_2$ denotes the $\ell_2$-norm and $\|\cdot\|_F$ denotes the Frobenius norm. $*$ denotes the 2-D convolution.

## 2. Signal Model and Time-Frequency Analysis

### *2.1. Signal Model*

Consider a $P$-component discrete-time FM signal

$$s(t) = \sum_{p=1}^{P} s_p(t) = \sum_{p=1}^{P} a_p e^{\jmath\phi_p(t)}, \quad t = 1, \cdots, T, \tag{1}$$

where $a_p$ and $\phi_p(t)$ respectively denote the amplitude and the phase law of the $p$th component for $p = 1, \cdots, P$. The IF of the $p$th signal component is given by $f_p(t) = \mathrm{d}\phi_p(t)/(2\pi\ \mathrm{d}t)$.

The IAF of $s(t)$ is defined as

$$\begin{aligned} R_{ss}(t,\tau) &= s\left(t+\tau\right) s^*\left(t-\tau\right) \\ &= \sum_{p=1}^{P} s_p\left(t+\tau\right) s_p^*\left(t-\tau\right) + \sum_{p=1}^{P} \sum_{\substack{q=1 \\ q\neq p}}^{P} s_p\left(t+\tau\right) s_q^*\left(t-\tau\right), \end{aligned} \tag{2}$$

where $\tau$ is the time lag. The first term in the right-hand expression implies autoterms of the $P$ components, whereas the last term represents their crossterms.

The DFT of the IAF $R_{ss}(t,\tau)$ with respect to $\tau$ is the well-known WVD, which is considered as the prototype bilinear TFR since it does not apply a TF kernel. The WVD of $s(t)$ is given as

$$W_{ss}(t,f) = \mathcal{F}_\tau[R_{ss}(t,\tau)] = \sum_{\tau} R_{ss}(t,\tau) e^{\jmath 4\pi f\tau}. \tag{3}$$

Note that, in the above expression, $4\pi$ is used because only integer values of $\tau$ is used in (2) and the actual lag is thus $2\tau$. The WVD can be divided into the autoterms and crossterms, given by

$$W_{ss}(t,f) = \underbrace{\sum_{p=1}^{P} W_{s_p s_p}(t,f)}_{\text{Autoterms}} + \underbrace{2\sum_{p=1}^{P} \sum_{q=p+1}^{P} \mathrm{Re}\left[W_{s_p s_q}(t,f)\right]}_{\text{Crossterms}}. \tag{4}$$

The two terms at the right-hand side respectively show the autoterms and crossterms. Similarly, the AF is obtained by applying 1-D DFT to the IAF

$R_{ss}(t,\tau)$ with respect to $t$, expressed as

$$\begin{aligned} A_{ss}(\theta,\tau) &= \mathcal{F}_t[R_{ss}(t,\tau)] \\ &= \underbrace{\sum_{p=1}^{P} A_{s_p s_p}(\theta,\tau)}_{\text{Autoterm AF}} + \underbrace{2\sum_{p=1}^{P}\sum_{q=p+1}^{P} \text{Re}\left[A_{s_p s_q}(\theta,\tau)\right]}_{\text{Crossterm AF}}. \end{aligned} \tag{5}$$

where $\theta$ denotes the frequency shift or Doppler. The AF has a 2-D Fourier transform relationship with the WVD.

*2.2. Kernel Design*

The last term in (4) represents undesirable crossterms, which are byproducts induced by the bilinear nature of the WVD and appear in the midway between any pair of autoterm components. TF kernels for crossterm mitigation are often implemented in the AF domain as a 2-D multiplicative filter preserving the region around the origin since, typically, autoterms are centered around the origin whereas the crossterms are dislocated from the origin. However, as the exact characteristics of autoterms and crossterms vary with each signal, there is no single TF kernel that fits all signals. Optimized design of TF kernels has been an important task in TF analyses in the past several decades.

As we discussed earlier, TF kernels can be classified into two general types, i.e., data-independent (fixed) and data-dependent (adaptive). Adaptive kernels are designed to maximize certain performance measure under some constraints and generally provide better performance compared to fixed kernels. To depict the nature and the limitation of adaptive kernels, we briefly describe a commonly used AOK [12].

The AOK is designed based on radially Gaussian windows with an angle-dependent window size, given by:

$$\begin{aligned} &\Phi_{\text{opt}}(r,\psi) = \underset{\Phi(r,\psi)}{\arg\max} \int_0^{2\pi}\int_0^{\infty} |A(r,\psi)\Phi(r,\psi)|^2 r\text{d}r\text{d}\psi \\ &\text{s.t.} \quad \Phi(r,\psi) = \exp\left(-\frac{r^2}{2\sigma^2(\psi)}\right), \\ &\qquad \frac{1}{4\pi^2}\int_0^{2\pi} \sigma^2(\psi)d\psi \le \alpha, \end{aligned} \tag{6}$$

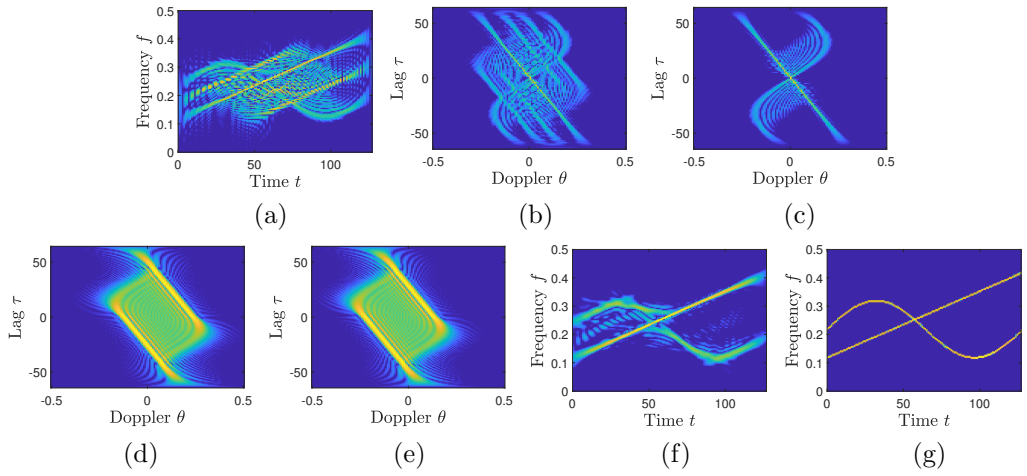



Figure 1: Results for a noiseless signal consisting of one SFM and LFM. (a) WVD of $s(t) = s_1(t) + s_2(t)$; (b) AF of $s(t) = s_1(t) + s_2(t)$; (c) auto AF of $s(t) = s_1(t) + s_2(t)$; (d) cross AF between $s_1(t)$ and $s_2(t)$; (e) cross AF between $s_2(t)$ and $s_1(t)$; (f) TFR obtained by the AOK for $s(t) = s_1(t) + s_2(t)$; (g) proposed method.

where $A(r, \psi)$ represents the AF of the signal in polar coordinates, $r$ and $\psi$ denote the radius and radial angle, respectively, and $\alpha$ denotes the kernel volume constraint. In short, the AOK optimizes the width of radial Gaussian function in different directions to maximize the overall volume of the kernelled AF, $\bar{A}(r, \psi) = A(r, \psi)\Phi_{\text{OPT}}(r, \psi)$ in the entire polar coordinate system.

Denote $\bar{A}(\theta, \tau)$ as the corresponding AF in the rectangular coordinate system. Then, the corresponding TFR is given by

$$\rho_{\text{AOK}}(t, f) = \mathcal{F}_\tau \left\{ \mathcal{F}_\theta^{-1}[\bar{A}(\theta, \tau)] \right\}. \tag{7}$$

### *2.3. Demonstration Example*

As an example, we consider a two-component signal $s(t) = s_1(t) + s_2(t)$ consisting of sinusoidal FM (SFM) component $s_1(t) = e^{\jmath\phi_1(t)}$ and linear FM (LFM) component $s_2(t) = e^{\jmath\phi_2(t)}$. In this case, the instantaneous phase laws of the two components are as respectively given as:

$$\begin{aligned} \phi_1(t) &= 2\pi \left( T/(20\pi) \cos\left(2\pi t/T + \pi\right) + 0.22t \right), \\ \phi_2(t) &= 2\pi \left( 0.12t + 0.15t^2/T \right). \end{aligned} \tag{8}$$

No noise is considered in this example.

Figure 1(a) depicts the WVD of $s(t)$ in which severe crossterms exist due to the intersection of two signal components and the nonlinearity of the SFM component. As a result, recognition of true IFs becomes difficult. Figure 1(b) shows the AF of $s(t)$. To clearly understand how the autoterms and crossterms of this signal overlap in the AF domain, we depict in Figure 1(c) the autoterm AF, i.e., the superposition of the AF of $s_1(t)$ and that of $s_2(t)$. Figure 1(d) shows the crossterm AF between $s_1(t)$ and $s_2(t)$ and Figure 1(e) shows the crossterm AF between $s_2(t)$ and $s_1(t)$. Note that, although these crossterm plots in Figures 1(e) and 1(d) look similar, they differ in their shapes and positions. It is clear from Figures 1(b)–1(e) that it is difficult to separate the crossterm AFs from the autoterm AF. As a result, obtaining a desired TFR with preserved autoterms and mitigated crossterms is challenging. As shown in Figure 1(f), the AOK is not able to handle such challenging situation, and renders poor TFR particularly for the SFM component. Figure 1(g) shows the result of the proposed method which successfully reconstructs both components without distortion.

## 3. Convolutional Autoencoder-based Crossterm-Free TFR

In this section, we describe the proposed deep learning-based crossterm-free TFR reconstruction by exploiting the CAE architecture. In the proposed method, obtaining crossterm-free TFR is considered as a generative learning problem which, in essence, provides supervised TF kernel optimization capability to minimize the reconstruction error at the network output. As such, the CAE acts as an optimized TF kernel which, unlike any existing TF kernel, is trained to output TFRs which are conceptually ensured to preserve autoterms and mitigate crossterm through the minimization of the loss function evaluated at the network output. It is noted that, while the CAE architecture is used in this paper to provide high flexibility to optimize the TF kernel with a low complexity, other deep learning network architectures can also be used to implement the proposed crossterm-free TFR method.

In the following, we first describe the CAE architecture, and then present the network training process.

### *3.1. Proposed Convolutional Autoencoder Architecture*

A high-level diagram of the CAE architecture used in this paper is depicted in Figure 2. We adopt the WVD of a signal as the input image $\mathbf{X}$ of the CAE, and the corresponding training label $\mathbf{Y}$ is constructed from the

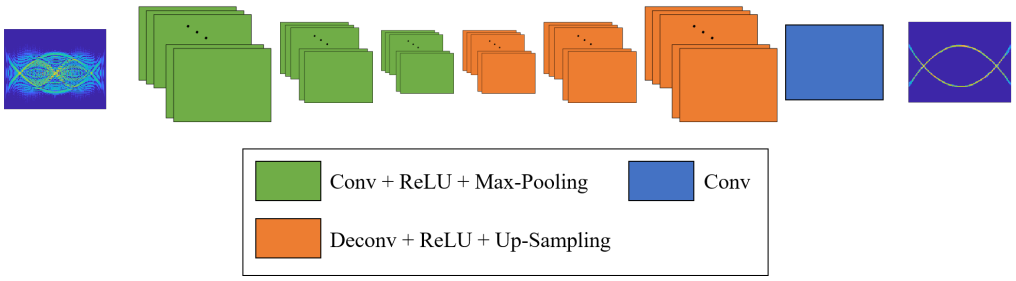



Figure 2: CAE architecture to achieve the high-resolution crossterm-free TFR.

IF law of the signal components scaled by their respective power. It is clear that the label $\mathbf{Y}$ is crossterm-free and reflects high-resolution autoterms. The CAE extracts features from the WVD to reconstruct the TFR, and minimizes the difference between the reconstructed TFR and the corresponding label in the minimum mean square error sense. The CAE can be considered as the concatenation of two sections, i.e., an encoder and a decoder.

*3.1.1. Encoder*

The encoder consists of $N$ network layers. Each network layer includes several functional layers, namely, a convolutional layer followed by a rectified linear unit (ReLU, $\max(0, \cdot)$) and a max-pooling layer.

- Convolutional layer: In a convolutional layer, a neuron is only connected to a local region called the receptive field. Neurons in a convolutional layer compute the inner product between their weights and the receptive field of the input image to generate the activation map or feature map. For each convolutional layer, $C$ filters of size $D \times D$ are used to generate $C$ feature maps. We use the "same" padding for the convolutional layers so the image size of a convolutional layer output is kept the same as its input size. The use of ReLU introduces nonlinearity and pushes negative outputs to actual zeros, thus further enhancing the TF sparsity at the network output.

  Denote $\mathbf{W}_c^{[n]}$ and $b_c^{[n]}$ as the weight coefficient matrix and the bias of the $c$-th channel at the $n$-th layer. Then, the activation from the $n$-th

convolutional layer is given by:

$$\begin{aligned} \mathcal{L}^{[0]}(\mathbf{X}) &= \mathbf{X}, \\ \mathcal{L}_c^{[n]}(\mathbf{X}) &= \text{ReLU}\left(\mathbf{W}_c^{[n]} * \mathcal{L}^{[n-1]}(\mathbf{X}) + b_c^{[n]}\right), \\ & \qquad n = 1, \cdots N, \quad c = 1, \cdots, C, \end{aligned} \tag{9}$$

where $\mathcal{L}_c^{[n-1]}$ is the $c$-th output feature map from the $(n-1)$-th network layer from the max-pooling layer as describe below, and is fed into the $n$-th network layer as the input. The first layer ($n = 1$) takes the input TF image $\mathbf{X}$ as the network input which is defined as $\mathcal{L}^{[0]}(\mathbf{X})$ for mathematical convenience.

- Max-Pooling layer: The ReLU output is followed by a max-pooling layer, which performs three functions, namely, reducing the size of the input image by a factor determined by the filter size and stride, adding translational invariancy to the feature maps thus enhancing the model's generalization capability, and keeping the most prominent feature values by avoiding trivial solutions [30].

  The max-pooling layer divides the feature map into several non-overlapping regions, and maps the largest values from each region to its output feature map. The $i$-th element of the $c$-th feature map output from the $n$-th max-pooling layer is given as

$$\mathcal{L}^{[n_i]} = \max(\mathcal{L}_c^{[n_i]}), \tag{10}$$

  where $\mathcal{L}_c^{[n_i]}$ is the $i$-th region of the $c$-th feature map resulted from the $n$-th convolutional layer.

After repeating the above procedures for all $N$ network layers in the encoder section, we obtain a number of feature maps with a reduced size. Accordingly, the input TFR is encoded to the feature maps with a significantly reduced dimension.

*3.1.2. Decoder*

The decoder section reconstructs the TFR from the encoded feature map from the encoder section. A deconvolutional layer, a ReLU, and an upsampling layer are stacked together to form each network layer of the decoder section. The hyperparameters such as filter size and number of filters used for the decoder section are kept the same as those in the encoder section.

- Deconvolutional layer: A deconvolutional layer performs the reverse operation of the corresponding convolutional layer and restores the TFR from the captured features. It performs the transposed convolutional operation to distribute the features of a feature map within its neighborhood.

  The output from the $n$th deconvolutional layer can be expressed as,

$$\mathcal{L}_{D_c}^{[n]}(\mathbf{X}) = \text{ReLU}\left(\mathbf{W}_{D_c}^{[n]} * \mathcal{L}_{D_c}^{[n-1]}(\mathbf{X}) + b_{D_c}^{[n]}\right), \quad n = 1, \cdots N, \quad c = 1, \cdots, C. \tag{11}$$

- Up-sampling layer: The up-sampling layer is introduced to reconstruct the original dimension of the TF images. The up-sampling operation preserves the location of the maximum values from pooling and zeros for the rest.

In addition to the encoder and decoder sections, one additional convolutional layer consisting of a single filter of size $D \times D$ is utilized to reconstruct the output TFR.

*3.2. Neural Network Training*

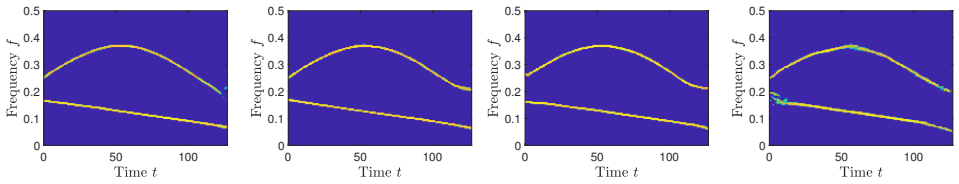
As examples, we consider two-component FM signals, expressed as:

$$x(t) = e^{\jmath\phi_1(t)} + e^{\jmath\phi_2(t)}, \tag{12}$$

for $t = 0, 1, \cdots, T-1$. We assume $T = 128$, and the resulting size of each input TFR image is $128 \times 128$. Two types of signals are considered for training. The first one consists of two NLFM signal components, whereas the second one consists of an LFM component and an SFM component. $2,000$ samples are randomly generated for each class with different parameters, such as the respective initial frequencies, frequency slope, and frequency difference. 90% of the samples are utilized for training and the remaining 10% are utilized for validation.

We adopt the mean square error between the estimated TFR $\hat{\mathbf{Y}}$ and label $\mathbf{Y}$ as the loss function, described as

$$\text{Loss}^{\{i\}} = \frac{1}{2M} \sum_{m=1}^{M} \|\hat{\mathbf{Y}}_m^{\{i\}} - \mathbf{Y}_m^{\{i\}}\|_F^2, \tag{13}$$




(a) 2 layers (b) 3 layers (c) 4 layers (d) 6 layers

Figure 3: Performance comparison with respect to the number of layers.

where $M$ is the number of samples in each batch, and $i$ denotes the batch index. Minimizing the mean square error will force $\hat{\mathbf{Y}}_m^{\{i\}}$ to match $\mathbf{Y}_m^{\{i\}}$ as close as possible and the minimum value is achieved when $\hat{\mathbf{Y}}_m^{\{i\}} = \mathbf{Y}_m^{\{i\}}$. Minimizing the mean square error is equivalent to maximizing the peak signal-to-noise ratio (PSNR), which is a popular image quality measure.

In this paper, we empirically set $N = 3$, $C = 40$, $D = 5$, and $M = 113$ that well balance the complexity and the performance. The optimizer implements the Adam algorithm [32], with all its hyperparameters set to their default values. Discussions on parameter selection are provided in Section 4.

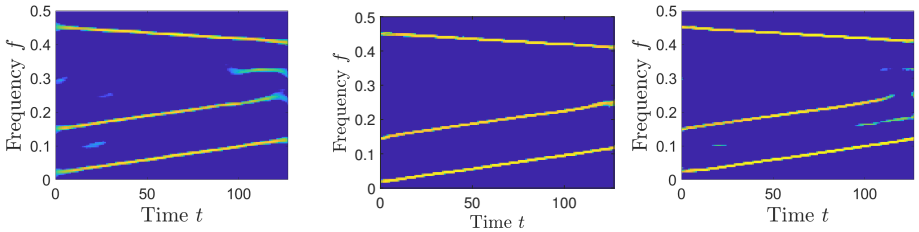
To verify the noise robustness of our proposed method, we consider four noise levels, i.e., noise-free ("inf" dB), 10 dB, 5 dB, and 0 dB. The same parameters are shared to generate the training dataset at different noise levels.

## 4. Parameter Selection

In this section, we discuss parameter selection in the proposed CAE to obtain the crossterm-free TFR from the WVD.

### *4.1. Number of Layers*

When processing a TF image, low level features like horizontal lines, vertical lines, and edges are learned in the lower layers, whereas more complex features are learned in the higher layers [31]. Considering crossterm-free TFR reconstruction, therefore, signals with a highly time-varying TFR signatures require a higher number of network layers to describe its features [31]. As a general rule, when the number of the network layers is insufficient, the network may not have the capability to learn all the useful features to describe the input TF image. Therefore, there is high probability that the training dataset is underfitted. On other hand, using a higher number of network layers will not only increase the computation cost, but may also




(a) 5 filters (b) 40 filters (c) 55 filters

Figure 4: Performance comparison with respect to the number of filters.

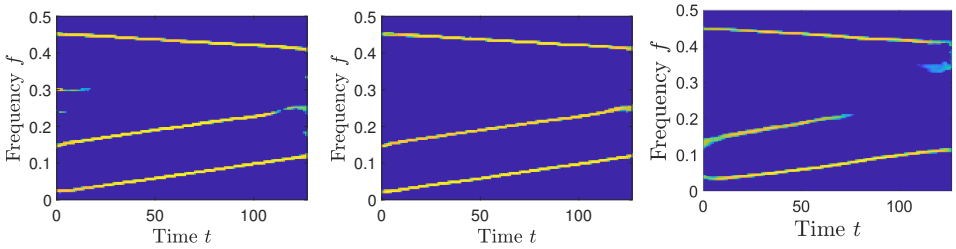
risk the overfitting problem and does not necessarily further improve the performance.

Figure 3 compares the TFR reconstruction performance of the proposed method that handles an LFM component and an SFM component using the CAE architecture with different numbers of layers. Figure 3(a) shows the results for the case of $N = 2$ network layers, where we see that the SFM component is distorted at the right edge. Increasing the number of network layers improves its performance as the network can capture more useful features. For the case of $N = 3$, Figure 3(b) shows improved reconstruction performance with smooth and continuous TFRs. The result for $N = 4$ depicted in Figure 3(c) shows comparable reconstruction performance. For the case of $N = 5$, however, we notice in Figure 3(d) that the LFM component becomes blurred at the edges. In fact, adding more layers might lead to overfitting, which indicates that the model cannot be well generalized to the testing data.

### *4.2. Number of Filters*

The deeper convolutional layers stack low-level features from its previous layers to make meaningful abstract shapes [31]. As the features detected at the higher layers are a combination of those at the prior layers, any features that are previously undetected can no longer be detected in later layers. In each layer, a plurality of filters are used to detect different types of features. A sufficient number of filters must be used in each layer to ensure that no useful features are lost. Therefore, if the number of filters is too small at a layer, it takes a high risk of not adequately capturing all useful features, thus degrading the performance. On the other hand, using too many filters in a layer may introduce the overfitting problem.

Comparing Figures 4(a) and 4(b), it is observed that, when using a small number of $C = 5$ filters, the performance of the CAE architecture is poor, whereas using an adequate number of $C = 40$ filters provides desired performance. On the other hand, in Figure 4(c) where $C = 55$ filters are used, the crossterms are not effectively mitigated due to the over-trained model. Consequently, it is vital to find an optimal number of filters that achieve high TFR reconstruction performance with a low computational complexity.




(a) Filter size: $3 \times 3$ (b) Filter size: $5 \times 5$ (c) Filter size: $11 \times 11$

Figure 5: Performance comparison for different filter sizes.

### *4.3. Filter Size*

In general, filters with a small size are useful for extracting meaningful local information and finer details, whereas large filters are used to extract global and more generalized information. However, using a large filter size will require a high complexity and may risk the overfitting problem.

In this paper, we select $D = 5 \times 5$ as the filter size since this size is suitable to capture the continuity of frequency signatures over a short time period. Figure 5 presents a performance comparison for three different filter sizes, i.e., $3 \times 3$, $5 \times 5$, and $11 \times 11$. We can see that the filter of size $D = 5 \times 5$ provides the best reconstruction performance in terms of crossterm mitigation and autoterm preservation as compared with the filters of size $D = 3 \times 3$ and $D = 11 \times 11$. When we have filters of size $D = 11 \times 11$, one LFM component cannot be fully detected, indicating that larger filter size might lead to overfitting since more parameters are utilized during the training procedure.

## 5. Simulation Results

In this section, we compare the proposed CAE-based learning method with the prototype WVD and five TFR reconstruction algorithms, namely,

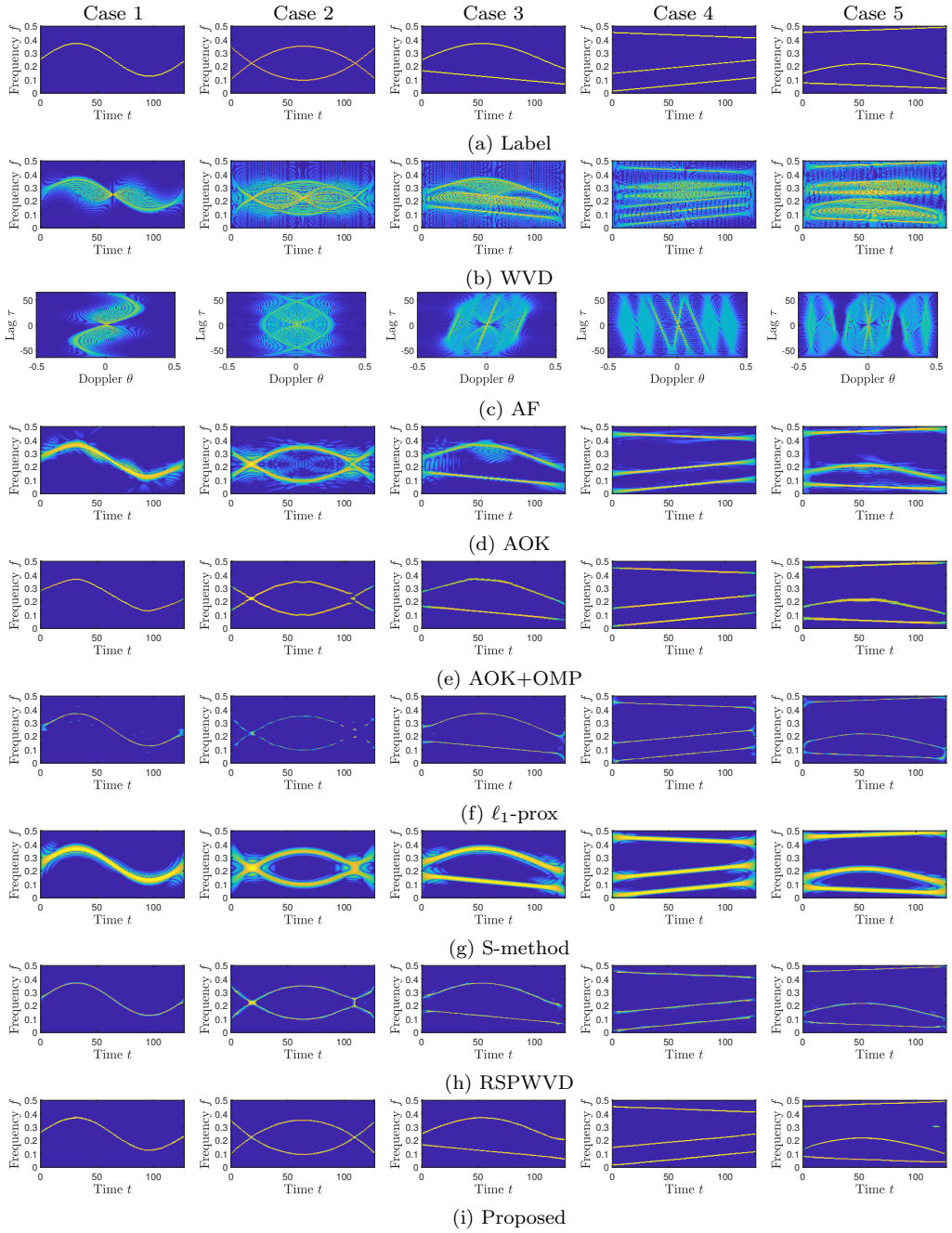



Figure 6: Reconstructed TFR results for the five cases. For all results, amplitudes are coded logarithmically with a dynamic range of 20 dB.

AOK[12], AOK+OMP [18], $\ell_1$-prox [16], S-method [33], and reassigned smoothed pseudo WVD (RSPWVD) [34]. The $\ell_1$-prox algorithm [16] takes a rectangular sampling area of size $13 \times 13$ in the AF domain centered at the AF-domain origin.

*5.1. Case Studies*

To demonstrate the effectiveness of the proposed method, both synthetic and real-world signals are taken into account. For synthetic signals, we consider five different cases which include signals with one component, two components, and three components. These comparisons verify the generalized learning capability and effectiveness of the proposed approach for crossterm mitigation irrespective of the number of signal components and types used in testing to some extent. The TFR reconstruction results of different methods are depicted in Figure 6 in a noise-free case. In this figure, columns 1 through 5 are respectively associated with cases 1 through 5. For real-world signals, the bat echolocation signal is considered in Section 5.1.6.

*5.1.1. Case 1: One SFM Component*

We first consider the following signal with a single SFM component whose phase law is given as,

$$\phi(t) = 2\pi \left(0.06T/\pi \cos(2\pi t/T + \pi) + 0.25t\right). \tag{14}$$

This case is represented in the first column of Figure 6. The AOK and AOK+OMP methods do not accurately capture the local information as AOK attempts to linearize the IF law of the reconstructed TFR. Comparing with the AOK-based method, the $\ell_1$-prox method, S-method, and RSPWVD provide much better capability to preserve the smoothness of the sinusoidally time-varying IF signature. In particular, RSPWVD provides an accurate and high-resolution TF signature which is very close to the ground truth except minor smearing at both edges. The result presented in the first column of Figure 6(i) shows that the proposed method provides accurate and high-resolution TFR including both edges.

*5.1.2. Case 2: Two Intersecting NLFM Components*

Two intersecting NLFM components are considered, and their instantaneous phase laws are respectively given by:

$$\begin{aligned} \phi_1(t) &= 2\pi \left(0.35t - 0.50t^2/T + 0.33t^3/T^2\right) \\ \phi_2(t) &= 2\pi \left(0.10t + 0.50t^2/T - 0.33t^3/T^2\right). \end{aligned} \tag{15}$$

Compared to case 1, the signal consisting of two intersecting NLFM components generates much higher crossterms and, as a result, is much more challenging to handle. In the plot depicted at the second column of Figure 6(e), we observe that the AOK+OMP substantially eliminates crossterms, but it cannot correctly detect the overlapped spectrum around $t = 110$. Compared with AOK+OMP, the $\ell_1$-prox approach renders TFR with less IF distortions, especially in the middle section. However, it fails to reconstruct IFs in the overlapped portion on the right hand side because, in this case, the autoterms and crossterms are difficult to be separated in the AF domain. Both S-method and RSPWVD obtain a cleaner and more consistent TFR compared with the AOK method. However, the TFR is distorted at the intersections of the two signal components. The proposed method is the only method among those compared to accurately obtain TFR for the entire time period including the intersections.

*5.1.3. Case 3: One SFM and One LFM Components*

In this case, we consider a signal consisting of one SFM and one LFM components. Their instantaneous phase laws are given as:

$$\begin{aligned} \phi_1(t) &= 2\pi \left(0.12T/(1.20\pi)\cos(1.20\pi t/T + \pi) + 0.25t\right), \\ \phi_2(t) &= 2\pi \left(0.17t - 0.05t^2/T\right). \end{aligned} \tag{16}$$

The results for case 3 are represented in the third column of Figure 6. The AOK+OMP method well reconstructs the TFR of the LFM component, but the SFM component is heavily distorted since it is difficult for the AOK to handle highly nonlinear IF signatures. Compared with the AOK+OMP, the $\ell_1$-prox method provides better capability to handle the high IF nonlinearity, but it does not preserve the shape of the autoterms at the edges. Both S-method and RSPWVD also maintain the curvature well for the NLFM signal component, and RSPWVD provides a high TFR resolution. In comparison, the proposed TFR method provides cleaner and smoother TFR for both components.

*5.1.4. Case 4: Three LFM Components*

In this case, the instantaneous phase laws of the three LFM components are:

$$\begin{aligned} \phi_1(t) &= 2\pi \left(0.02t + 0.05t^2/T\right), \\ \phi_2(t) &= 2\pi \left(0.15t - 0.05t^2/T\right), \\ \phi_3(t) &= 2\pi \left(0.45t - 0.02t^2/T\right). \end{aligned} \tag{17}$$

We observe in the fourth column of Figure 6 that, for this relatively simple case with three LFM components, all methods provide good TFR reconstruction results. Among them, the compressive sensing-based methods, the RSPWVD, and the proposed method provide higher TFR resolution as compared to the AOK and the S-method.

*5.1.5. Case 5: One SFM and Two LFM Components*

Now, we consider a three-component signal with one SFM component and two LFM components. Their respective instantaneous phase laws are given as,

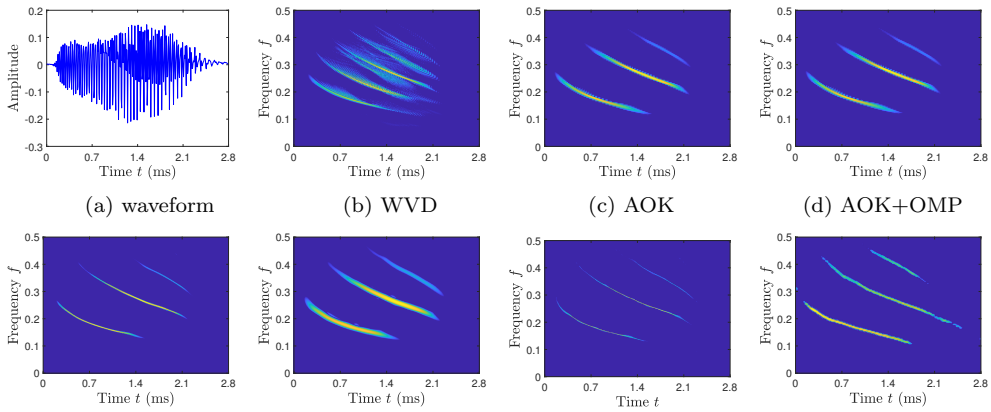
$$\begin{aligned}\phi_1(t) &= 2\pi\left(0.07T/(1.20\pi)\cos(1.20\pi t/T+\pi)+0.15t\right),\\ \phi_2(t) &= 2\pi\left(0.45t+0.02t^2/T\right),\\ \phi_3(t) &= 2\pi\left(0.08t-0.02t^2/T\right).\end{aligned} \tag{18}$$

The AOK+OMP results depicted in the ((e)th row, 5th column) of Figure 6 does not detect the SFM component faithfully, especially when $t$ is around 50. Compared to the AOK+OMP, the $\ell_1$-prox method and S-method fail to cleanly eliminate the crossterms. Moreover, they cannot separate the SFM and the lower-frequency LFM component at the edges. The RSPWVD provides a comparable performance as the proposed method, which detects all IF components successfully. We notice that the proposed method has some residual crossterms during $115 \leq t \leq 121$. Nevertheless, the proposed method maintains a clean and high-resolution spectrum for the entire time period including both edges.

In all the above five cases, the proposed method consistently provides near-ideal TFRs irrespective of the number of signal components and signal types. These examples also verify that, while two-component FM signals are used in our examples for training, the proposed method is not restricted only to two-component FM signals but could be applied to general cases with more or less than two components. Moreover, the proposed method is insensitive to the finite sampling effect and maintains the signal energy along the true signal IFs.

*5.1.6. Real-life Bat Echolocation Signal*

To demonstrate the superior performance of the proposed method in handling real-life signals, we consider the commonly compared bat echolocation




(a) waveform (b) WVD (c) AOK (d) AOK+OMP

(e) $\ell_1$-prox (f) S-method (g) RSPWVD (h) Proposed

Figure 7: Performance of the model in case of reconstructing real-life bat echolocation signal

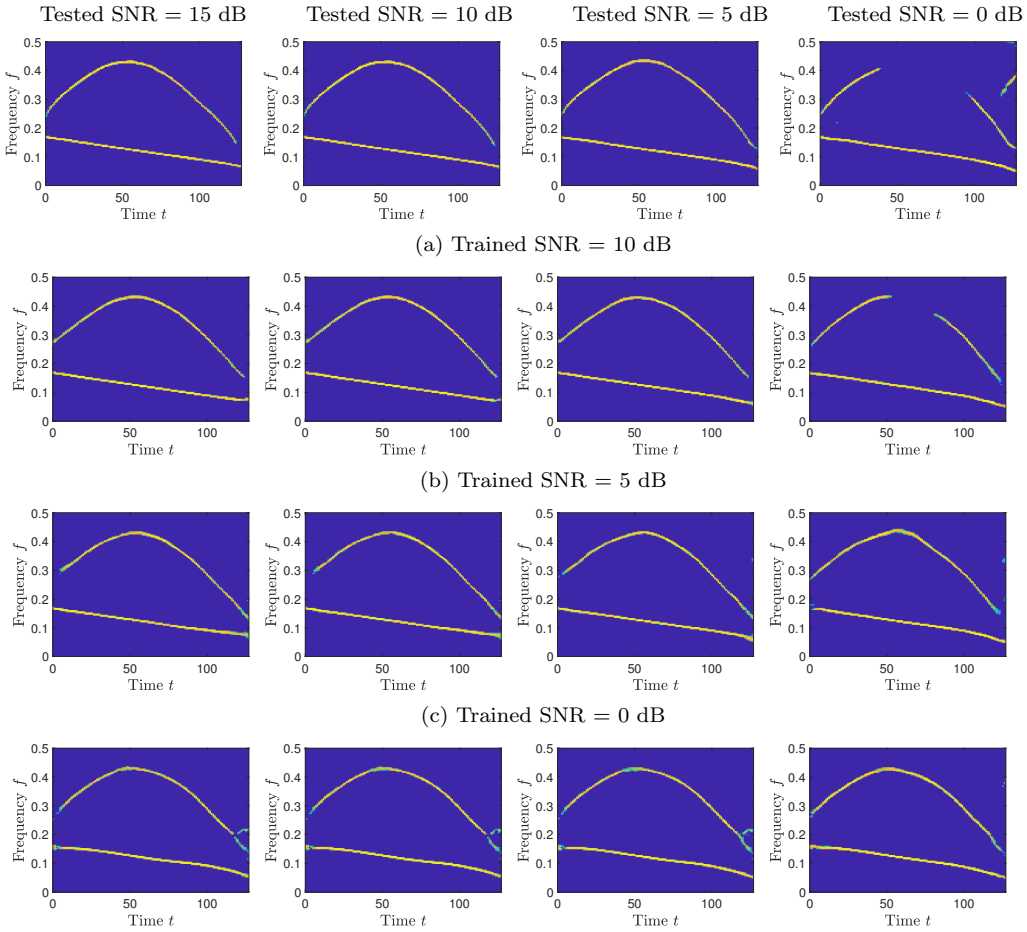
exponential chirp signal emitted by Eptesicus fuscus [1]. This 2.8 ms data contains 400 samples with a sampling period of 7 $\mu$s. Particularly, this pulse contains three gently curved harmonics, which are nearly linear if expressed with respect to the logarithmic time [35].

Figure 7(a) depicts the waveform for the bat signal, which reflects the signal energy distribution over a time period of 2.8 ms. Figure 7(b) represents the WVD of the bat echolocation signal, which is obscured by the excessive crossterms. AOK, AOK+OMP, $\ell_1$-prox, S-method, and RSPWVD all provide good TFR reconstruction performance. However, the TFR result obtained from the proposed method not only offers a high resolution, but also well maintains the signal energy, especially for $t > 2.1$ ms.

### *5.2. Robustness Analysis*

To quantitatively compare the performance of the proposed CAE-based crossterm-free TFR reconstruction other TFR reconstruction methods, we evaluate the TFRs from two different perspectives, i.e., fidelity and energy concentration. We consider noisy signal measurements with different levels of input SNR. Except for the noise-free case, $K = 50$ independent trials are performed to obtain the average value.

[1]The authors wish to thank C. Condon, K. White, and A. Feng of the Beckman Institute of the University of Illinois for the bat data and for permission to use it in this paper.




(a) Trained SNR = 10 dB

(b) Trained SNR = 5 dB

(c) Trained SNR = 0 dB

(d) Trained SNR = 0 dB (one additional convolutional and deconvolutional layer in the model)

Figure 8: Effect of noises on the performance

At first, the normalized mean square error (NMSE) is adopted to measure the fidelity, which is defined as follows:

$$\text{NMSE} = 10\log_{10}\left(\|\mathbf{Y}-\hat{\mathbf{Y}}\|_2^2/\|\mathbf{Y}\|_2^2\right). \tag{19}$$

Table 1 summarizes the NMSE results. We observe that the case of three LFM components obtains the lowest NMSE among the five cases since LFM components are relatively easier to be reconstructed. The S-method performs better than the AOK method in most cases, especially when the signal consists of NLFM components because AOK tends to linearize the TF signature.

Table 1: NMSE (dB) comparison among WVD, AOK[12], AOK+OMP[18], $\ell_1$-prox [16], S-method [33], reassigned smoothed pseudo WVD (RSPWVD) [34], and the proposed CAE-based method for different SNR values

| Case | SNR(dB) | WVD | AOK | AOK+OMP | $\ell_1$-prox | S-method | RSPWVD | Proposed |
|---|---|---|---|---|---|---|---|---|
| One SFM | Inf | −0.33 | 3.14 | 0.34 | −2.66 | 2.38 | −2.95 | −4.29 |
| | 10 | −0.10 | 2.99 | 0.22 | −1.29 | 1.83 | −1.49 | −3.69 |
| | 5 | 0.45 | 2.95 | 0.17 | −0.71 | 1.72 | −0.82 | −3.08 |
| | 0 | 2.33 | 2.98 | 0.12 | −0.38 | 2.08 | −0.39 | −1.29 |
| Two NLFMs | Inf | 2.57 | 3.72 | −0.35 | −0.76 | 2.37 | −1.78 | −5.82 |
| | 10 | 2.91 | 3.74 | −0.10 | −0.57 | 2.50 | −1.59 | −5.10 |
| | 5 | 3.62 | 3.83 | 0.21 | −0.38 | 2.76 | −1.18 | −3.23 |
| | 0 | 5.22 | 4.42 | 1.06 | −0.23 | 3.56 | −0.51 | −0.56 |
| One SFM and one LFM | Inf | −0.38 | 0.36 | −2.49 | −2.20 | −0.11 | −2.31 | −5.82 |
| | 10 | −0.20 | 0.37 | −2.47 | −1.41 | −0.52 | −1.59 | −5.95 |
| | 5 | 0.16 | 0.40 | −2.42 | −0.82 | −0.63 | −0.89 | −5.32 |
| | 0 | 1.07 | 0.57 | −2.25 | −0.44 | −0.42 | −0.77 | −2.49 |
| Three LFMs | Inf | −0.97 | 0.81 | −4.44 | −2.60 | −0.65 | −2.61 | −8.02 |
| | 10 | −0.88 | 0.79 | −4.36 | −1.53 | −1.23 | −1.53 | −7.76 |
| | 5 | −0.71 | 0.77 | −4.22 | −0.93 | −1.43 | −1.30 | −6.54 |
| | 0 | −0.16 | 0.72 | −3.83 | −0.47 | −1.34 | −1.12 | −2.94 |
| One SFM and two LFMs | Inf | 0.35 | 1.49 | −3.56 | −1.76 | −0.73 | −2.03 | −5.71 |
| | 10 | 0.31 | 1.46 | −3.52 | −1.28 | −1.20 | −1.48 | −5.36 |
| | 5 | 0.43 | 1.43 | −3.42 | −0.83 | −1.38 | −1.37 | −4.26 |
| | 0 | 0.84 | 1.41 | −3.09 | −0.43 | −1.39 | −0.91 | −1.77 |

Both $\ell_1$-prox and RSPWVD provide good TFR reconstruction performance, with the RSPWVD consistently offering slightly better performance than $\ell_1$-prox. Generally, the TFR reconstruction performance is improved as the input SNR increases. However, for AOK, AOK+OMP, and S-method, the NMSE does not necessarily decrease with the input SNR since they fail to reconstruct a high-fidelity TFR in some challenging cases even when no noise is present. We notice in Table 1 that the proposed method consistently outperforms the other methods being compared in most scenarios. The effectiveness and robustness of the proposed method is thus evidently verified.

The energy concentration measure is defined as [36]:

$$M_2^2 = \left( \sum_{t=1}^{T} \sum_{f'=1}^{N_f} |\rho(t, f')|^{\frac{1}{2}} \right)^2 \tag{20}$$

Table 2: $M_2^2$ comparison among WVD, AOK[12], AOK+OMP[18], $\ell_1$-prox [16], S-method [33], reassigned smoothed pseudo WVD (RSPWVD) [34], and the proposed CAE-based method for different SNR values

| Case | SNR(dB) | WVD | AOK | AOK+OMP | $\ell_1$-prox | S-method | RSPWVD | Proposed |
|---|---|---|---|---|---|---|---|---|
| One SFM | Inf | 9692 | 5404 | 247 | 553 | 3630 | 1440 | 312 |
| | 10 | 11979 | 7123 | 247 | 1547 | 6733 | 2052 | 315 |
| | 5 | 12837 | 8649 | 246 | 4100 | 8571 | 2991 | 325 |
| | 0 | 13270 | 10830 | 245 | 7606 | 10618 | 4077 | 392 |
| Two NLFMs | Inf | 11142 | 8270 | 743 | 762 | 5845 | 1369 | 610 |
| | 10 | 12231 | 8865 | 742 | 1088 | 7251 | 1841 | 605 |
| | 5 | 12823 | 9677 | 741 | 2398 | 8354 | 2467 | 630 |
| | 0 | 13252 | 11071 | 738 | 6154 | 9902 | 5683 | 579 |
| One SFM and one LFM | Inf | 10824 | 6857 | 719 | 784 | 4784 | 1289 | 567 |
| | 10 | 11992 | 8196 | 719 | 1438 | 6800 | 1720 | 562 |
| | 5 | 12662 | 9300 | 719 | 2838 | 8129 | 2441 | 566 |
| | 0 | 13155 | 10919 | 720 | 6212 | 9885 | 3442 | 556 |
| Three LFMs | Inf | 11496 | 6215 | 1090 | 1073 | 5149 | 1909 | 819 |
| | 10 | 12115 | 6575 | 1090 | 1852 | 6885 | 2113 | 819 |
| | 5 | 12627 | 7267 | 1090 | 3232 | 8056 | 2645 | 819 |
| | 0 | 13101 | 8878 | 1091 | 6365 | 9657 | 3588 | 819 |
| One SFM and two LFMs | Inf | 11674 | 6751 | 1443 | 1171 | 4992 | 1622 | 823 |
| | 10 | 12299 | 7225 | 1445 | 1708 | 6259 | 1912 | 829 |
| | 5 | 12769 | 7997 | 1446 | 3034 | 7335 | 2503 | 861 |
| | 0 | 13163 | 9641 | 1450 | 6115 | 8962 | 3489 | 861 |

where $\rho$ is the normalized TFD such that

$$\sum_{t=1}^{T}\sum_{f'=1}^{N_f}|\rho(t,f')| = 1. \tag{21}$$

Here, $f' = 1, \cdots, N_f$ denotes the frequency bin index and $N_f$ stands for the total number of frequency bins. A lower value of $M_2^2$ indicates higher energy concentration which corresponds to a smaller support region occupation. As depicted in Table 2, the proposed method consistently provides the lowest value of $M_2^2$ among all the compared TFR reconstruction methods, thereby demonstrating the highest energy concentration.

### *5.3. Generalization Capability Analysis*

In this subsection, we investigate the generalization capability, i.e., the capability to handle unseen data, of the proposed method. In other words, we will look at different scenarios where the test signals deviate from the training signals. Unless otherwise specified, a signal consisting of one LFM

signal component and one SFM signal component is considered for illustration purpose, and the input SNR is 10 dB for both training and testing signals.

*5.3.1. Effect of Noise Levels*

First, we examine how noise on training as well as test signals affect the performance of the proposed method. The model is trained using WVD images with 10 dB, 5 dB and 0 dB input SNR, as respectively depicted in Figures 8(a), 8(b), and 8(c). The columns from the left to right in Figure 8 represent the output of the model for test signals with 15 dB, 10 dB, 5 dB, and 0 dB input SNR, respectively.

Figure 8(a) represents the case when the model is trained on 10 dB WVD images. It is clear that the proposed model almost perfectly reconstructs the model TFR for the cases of 15 dB, 10 dB and 5 dB input SNR. However, for the test signal with 0 dB input SNR it performs poorly and the sinusoidal component cannot be fully detected and the LFM component is distorted at the right edge. Figure 8(b) shows that, when the network is trained using signals with 5 dB input SNR, the TFR reconstruction result of the test signals with 0 dB input SNR is improved, but the estimated signature for the sinusoidal component is still discontinuous.

The performance of the CAE model trained using signals with 0 dB input SNR is illustrated in Figure 8(c). In this case, the performance for the test signal of 0 dB is substantially improved. However, it still does not completely detect the SFM component especially at the right edge, whereas the left edge of the LFM component is slightly deformed. When the network model is trained using 0 dB input SNR training images, the performance improvement is because the training and test signals have closer statistics.

We further add one pair of additional convolutional and deconvolutional layers in the network model aiming to better detect noisy signals at 0 dB input SNR, and the results are shown in Figure 8(d) where the network model is trained using 0 dB SNR signals as well. In this case, it is noted that, the CAE model with one pair of additional layers performs better in the 0 dB case and detects both SFM and LFM components completely. Therefore, increasing the number of layers enhances the reconstruction capability of our model for extremely noisy circumstances.

It can be summarized from Figure 8 that it is best to match the input SNR of the training signal with that of the tested signal. When the input SNR of the training signal does not match that of the tested signal, the

performance tends to degrade.

*5.3.2. Effect of Amplitude Difference between Signal Components*

Next, we investigate the effect of the mismatch in terms of the amplitude difference on the performance of the proposed method. In Figure 9, the results of three different amplitude ratios are presented. For the three columns, the magnitude of the first signal, $a_1$, is kept as unity, whereas that of the second signal, $a_2$, is respectively set to 0.8, 0.5, and 0.4. It is observed in this figure that the weak signal remains clearly detectable when the amplitude ratio is 0.5. However, further reduction of the signal strength of the weak signal component renders the resulting TFR reconstruction performance inconsistent. On the other hand, we observe that the resulting TFR shows closer amplitude ratio. It is mainly caused by the fact that the same magnitudes are assumed in the two signal components of the training dataset.

*5.3.3. Effect of Variation Speed of the IFs*

Now we consider the test signal with different speed of variation of the IFs. This is considered by exploiting the SFM signal component with different number of cycles of frequency variation, namely, $N_{\text{sine}} = 0.3$, 1, 1.5, and 1.8 cycles. It is noted that in the training dataset, SFM signals with $N_{\text{sine}} \in [0.5, 1.5]$ are considered.

As depicted in Figure 10, the proposed model works well when $N_{\text{sine}}$ is 0.3 and 1. Note that $N_{\text{sine}} = 1$ is within the variation range of the training signals whereas $N_{\text{sine}} = 0.3$ is outside of this range. On the other hand, the reconstructed TFR starts degradation for $N_{\text{sine}} = 1.5$ and becomes worse for when $N_{\text{sine}} = 1.8$. As such, the results indicate that the proposed method can tolerate certain degrees of model deviation, but it becomes more challenging to handle rapidly time-varying SFM signals.

*5.3.4. Effect of Signal Fading*

To consider the effect of signal fading, we set the signal magnitude of one portion of the SFM signal component different from other portions. In Figure 11, we set the magnitude of both components to be 0.5, 0.3, and 0.1 for $t \in [41, 60]$, whereas the magnitude for the other portions to be 1. It is observed that the proposed method is able to recover the faded portion when the fading magnitude is 0.5 and 0.3, but the result does not correctly reflect the actual fading magnitudes. When the fading magnitude is 0.3, the estimated SFM signature is not properly recovered for the fading portion.

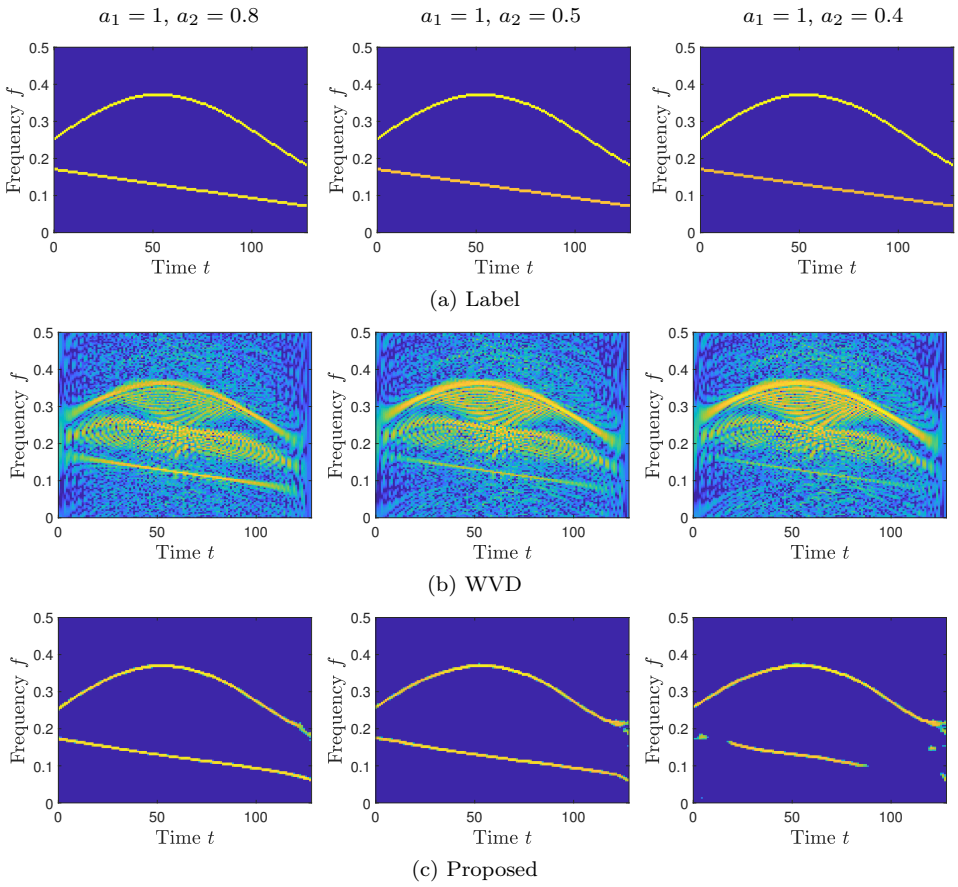


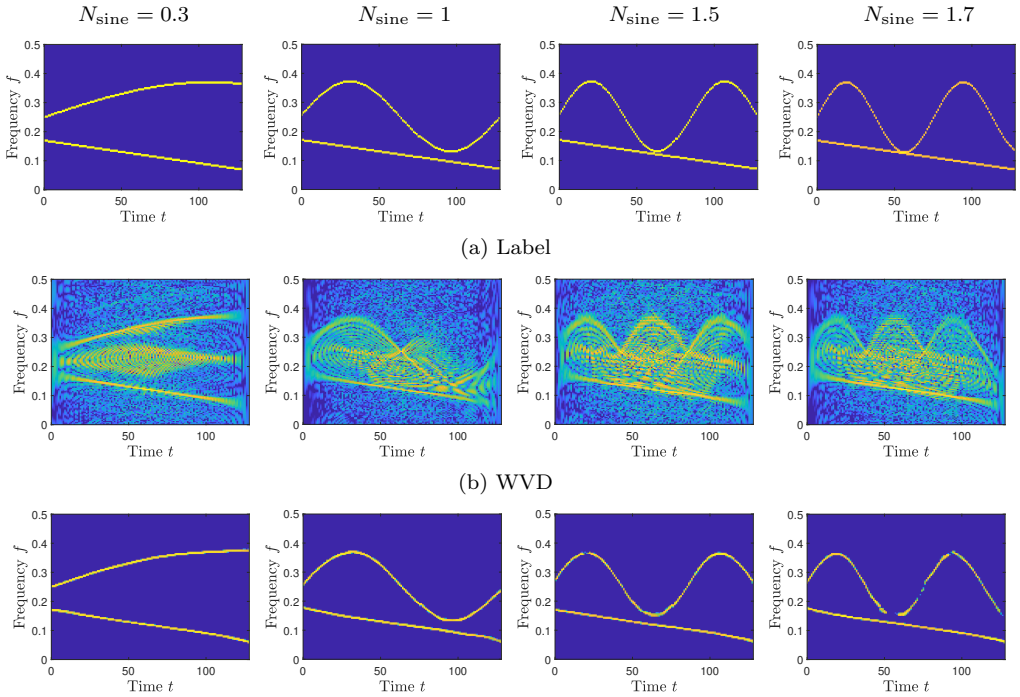

Figure 9: Effect of amplitude differences on the performance

*5.3.5. Effect of the Number of Signal Components*

In the training dataset, we only utilized signals consisting of two components. Here, we consider the performance of the proposed method when the number of signal components is higher than 2. In Figure 12, the three columns illustrate the results for 3-, 4-, and 5 component LFM signals, respectively. It is observed that the proposed method still performs well for the 3-component LFM signal case, but it starts to degrade when more than 3 signal components exist.

*5.3.6. Effect of Spreading Effect*

In the last example, we consider the scenario when one of the signal components has frequency spreading. In Figure 13(a), we model this effect




(a) Label

(b) WVD

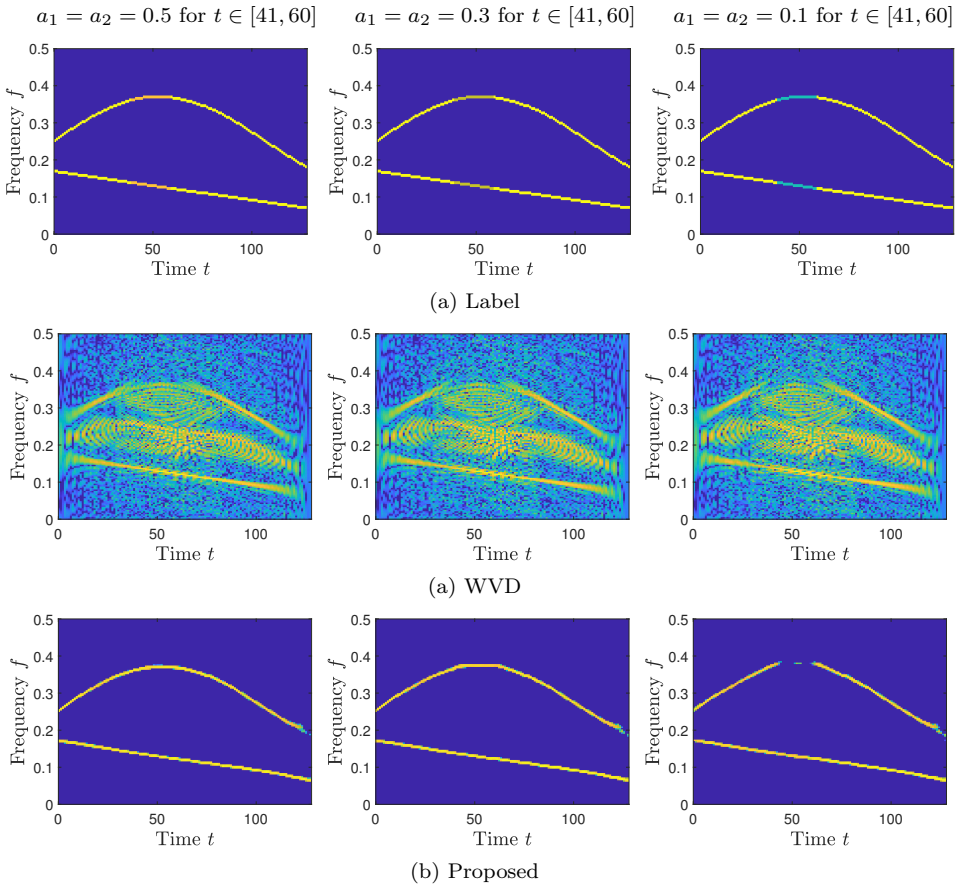
(b) Proposed

Figure 10: Effect of variation speed of the IFs on the performance

by including three closely spaced parallel LFM components and the results are depicted in Figure 13(c). It is observed that both signal components are successfully detected, but the result does not correctly reflect the spreading.

## 6. Conclusion

In this paper, we proposed a machine learning-based method using the CAE architecture to obtain crossterm-free TFRs. In the proposed method, a DNN is trained to provide effective generative learning capability to completely mitigate undesired crossterms while preserving signal autoterms. The performance of the proposed method is not restricted by the geometrical shapes in any domain as in the conventional TF kernel design. Simulation results provided for different nonstationary signals with slowly time-varying IFs show that, provided that the neural networks are adequately trained, the proposed method works robustly and provides significant performance improvement compared to existing TFR reconstruction algorithms.




(a) Label

(a) WVD

(b) Proposed

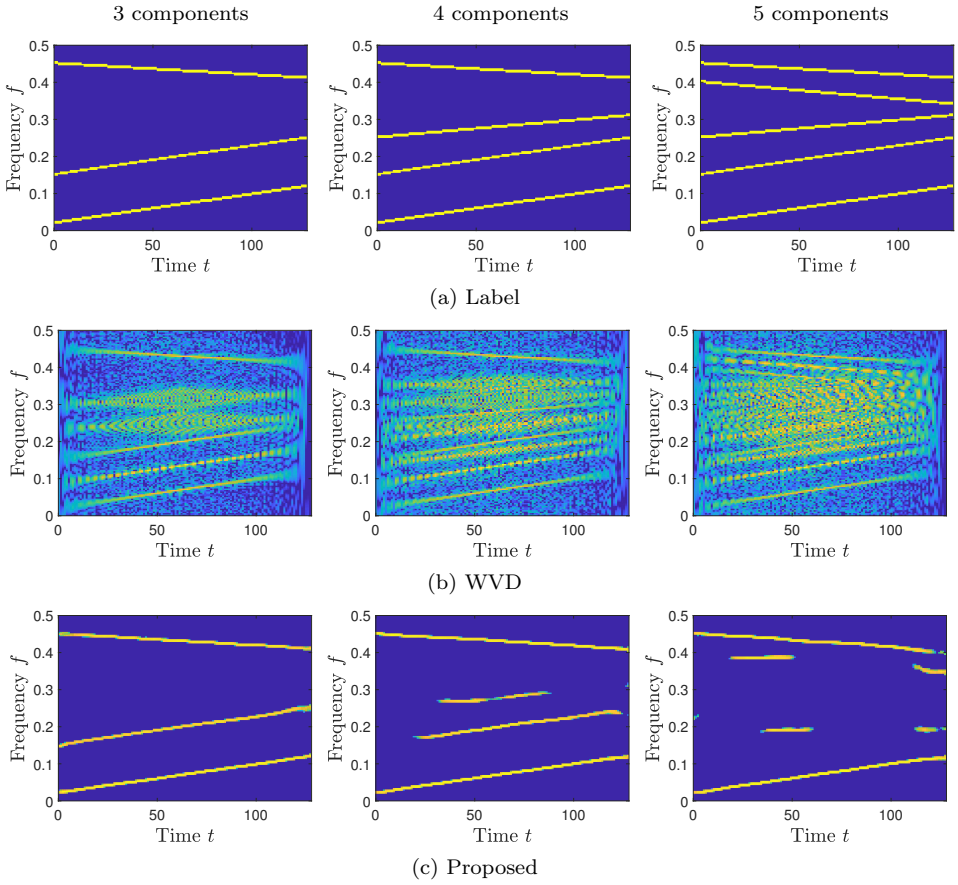
Figure 11: Effect of signal fading on the performance

## References


[1] A. Papandreou-Suppappola (Ed.), *Applications in Time-Frequency Signal Processing*. CRC Press, 2002.

[2] V. C. Chen and H. Ling, *Time-Frequency Transforms for Radar Imaging and Signal Analysis*. Artech House, 2002.

[3] Y. Zhang, M. G. Amin, and F. Ahmad, "A novel approach for multiple moving target localization using dual-frequency radars and time-frequency distributions," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 2007, pp. 1817–1821.

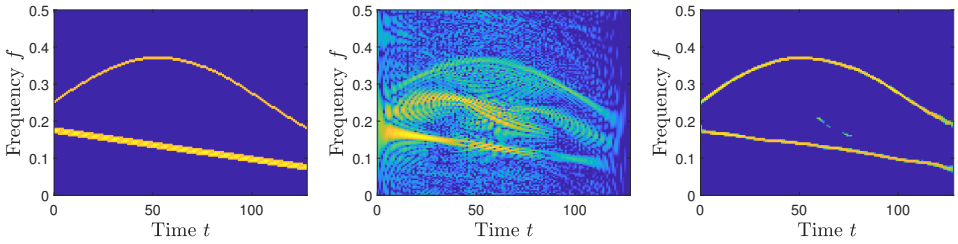



(a) Label

(b) WVD

(c) Proposed

Figure 12: Effect of number of signal components on the performance




(a) Label (b) WVD (c) Proposed

Figure 13: Effect of TF spreading on the performance.

[4] L. Stankovic, M. Dakovic, and T. Thayaparan, *Time-Frequency Signal Analysis with Applications*. Artech House, 2013.

[5] A. Belouchrani, M. G. Amin, N. Thirion-Moreau, and Y. D. Zhang, "Source separation and localization using time-frequency distributions," *IEEE Signal Process. Mag.*, vol. 30, no. 6, pp. 97–107, Nov. 2013.

[6] B. Boashash (ed.), *Time-Frequency Signal Analysis and Processing, 2nd Ed.* Academic Press, 2015.

[7] G. Liu, S. Fomel, and X. Chen, "Time-frequency analysis of seismic data using local attributes," *Geophysics*, vol. 76, no. 6, pp. 23–34, 2011.

[8] B. Boashash, G. Azemi, and J. M. O'Toole, "Time-frequency processing of nonstationary signals: Advanced TFD design to aid diagnosis with highlights from medical applications," *IEEE Signal Process. Mag.*, vol. 30, pp. 108–119, Nov. 2013.

[9] M. G. Amin, D. Borio, Y. D. Zhang, and L. Galleani, "Time-frequency analysis for GNSS: From interference mitigation to system monitoring," *IEEE Signal Process. Mag.*, vol. 34, no. 5, pp. 85–95, Sept. 2017.

[10] V. Shah, R. Anstotz, I. Obeid, and J. Picone, "Adapting an automatic speech recognition system to event classification of electroencephalograms," in *Proc. IEEE Signal Process. Medicine and Biology Symp.*, Philadelphia, PA, Dec. 2018, pp. 1–5.

[11] S. Zhang, A. Ahmed, and Y. D. Zhang, "Sparsity-based time-frequency analysis for automatic radar waveform recognition," in *Proc. IEEE Radar Conf.*, Washington DC, April 2020, pp. 548–553.

[12] D. L. Jones and R. G. Baraniuk, "An adaptive optimal-kernel time-frequency representation," *IEEE Trans. Signal Process.*, vol. 43, no. 10, pp. 2361–2371, Oct. 1995.

[13] N. Khan and B. Boashash, "Multi-component instantaneous frequency estimation using locally adaptive directional time frequency distributions," *Int. J. Adaptive Control Signal Process.*, vol. 30, pp. 429–442, March 2016.

[14] P. Flandrin and P. Borgnat, "Time-frequency energy distributions meet compressed sensing," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 2974–2982, June 2010.

[15] Z. Deprem and A. E. Çetin, "Cross-term-free time-frequency distribution reconstruction via lifted projections," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 479–491, Jan. 2015.

[16] P. Flandrin, N. Pustelnik, and P. Borgnat, "On Wigner-based sparse time-frequency distributions," in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, Cancun, Mexico, Dec. 2015, pp. 65–68.

[17] L. Jiang, H. Zhang, and L. Yu, "Robust time-frequency reconstruction by learning structured sparsity," April 2020. Available at https://arxiv.org/abs/2004.14820.

[18] Y. D. Zhang, M. G. Amin, and B. Himed, "Reduced interference time-frequency representations and sparse reconstruction of undersampled data," in *Proc. European Signal Process. Conf.*, Marrakech, Morocco, Sept. 2013, pp. 1–5.

[19] M. G. Amin, B. Jakonovic, Y. D. Zhang, and F. Ahmad, "A sparsity-perspective to quadratic time-frequency distributions," *Digital Signal Process.*, vol. 46, pp. 175–190, Nov. 2015.

[20] S. Zhang and Y. D. Zhang, "Robust time-frequency analysis of multiple FM signals with burst missing samples," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1172–1176, June 2019.

[21] Q. Wu, Y. D. Zhang, and M. G. Amin, "Continuous structure based Bayesian compressive sensing for sparse reconstruction of time-frequency distributions," in *Proc. Int. Conf. Digit. Signal Process.*, Hong Kong, China, Aug. 2014.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, pp. 1097–1105, Lake Tahoe, Nevada, Dec. 2012.

[24] G. Hinton, L. Deng, *et. al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups", *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[25] I. Obeid and J. Picone, "Machine learning approaches to automatic interpretation of EEGs," in E. Sejdik and T. Falk (Eds.), *Biomedical Signal Processing in Big Data*. CRC Press, 2017.

[26] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection with deep convolution neural network," in *Proc. IEEE Int. Conf. Image Process.*, Phoenix, AZ, Sept. 2016.

[27] M. Wang, Y. D. Zhang, and G. Cui, "Human motion recognition exploiting radar with stacked recurrent neural network," *Digital Signal Process.*, vol. 87, pp. 125–131, April 2019.

[28] L. H. Nguyen and T. D. Tran, "Deep CNN for extraction of sidelobes from SAR imagery in spectrally restricted environment," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 2019, pp. 2044–2047.

[29] S. Zhang, S. Pavel, and Y. D. Zhang, "Crossterm-free time-frequency analyses exploiting deep neural networks," (poster abstract) in *Proc. IEEE Signal Process. Medicine and Biology Symp.*, Philadelphia, PA, Dec. 2020.

[30] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 52–59.

[31] F. Chollet, *Deep Learning With Python*. Manning, Shelter Island, NY, USA, Nov. 2017.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, May 2015, pp. 1–15.

[33] L. J. Stankovič, "A method for time-frequency analysis," *IEEE Trans. Signal Process.*, vol. 42, no. 1, pp. 225–229, Jan. 1994.

[34] X. Wu and T. Liu, "Spectral decomposition of seismic data with reassigned smoothed pseudo Wigner–Ville distribution," *J. Appl. Geophys.*, vol. 68, no. 3, pp. 386–393, Jul. 2009.

[35] A. R. Wheeler, K. A. Fulton, J. E. Gaudette, R. A. Simmons, I. Matsuo, and J. A. Simmons, "Echolocating big brown bats, Eptesicus fuscus, modulate pulse intervals to overcome range ambiguity in cluttered surroundings," *Frontiers Behav. Neurosci.*, vol. 10, no. 125, June 2016.

[36] L. Stanković, "A measure of some time-frequency distributions concentration," *Signal Process.*, vol. 81, no. 3, pp. 621–631, Mar. 2001.