VECTOR: VERY DEEP CONVOLUTIONAL AUTOENCODERS FOR NON-RESONANT BACKGROUND REMOVAL IN BROADBAND COHERENT ANTI-STOKES RAMAN SCATTERING

A PREPRINT

Zhengwei Wang
School of Computer Science and Statistics
Trinity College Dublin
Ireland.
Current address: ByteDance AI Lab, Beijing, China villa.wang.zhengwei@gmail.com

Kevin O' Dwyer
Department of Electronic Engineering
Maynooth University
Co. Kildare, Ireland
kevin.odwyer@mu.ie

Ryan Muddiman
Department of Electronic Engineering
Maynooth University
Co. Kildare, Ireland
ryan.muddiman.2021@mumail.ie

Tomas Ward
Insight Centre for Data Analytics
School of Computing
Dublin City University
Dublin, Ireland
tomas.ward@dcu.ie

Charles. H. Camp Jr.
Biosystems and Biomaterials Division
National Institute of Standards and Technology
Gaithersburg, MD, USA
charles.camp@nist.gov

Bryan Hennelly
Department of Electronic Engineering
Maynooth University
Co. Kildare, Ireland
bryan.hennelly@mu.ie

March 29, 2022

ABSTRACT

Rapid label-free spectroscopy of biological and chemical specimen via molecular vibration through means of Broadband Coherent Anti-Stokes Raman Scattering (B-CARS) could serve as a basis for a robust diagnostic platform for a wide range of applications. A limiting factor of CARS is the presence of a non-resonant background (NRB) signal, endemic to the technique. This background is multiplicative with the chemically resonant signal, meaning the perturbation it generates cannot be accounted for simply. Although several numerical approaches exist to account for and remove the NRB, they generally require some estimate of the NRB in the form of a separate measurement. In this paper, we propose a deep neural network architecture called VECTOR (**V**ery d**E**ep **C**onvolutional au**TO**encode**R**s), which retrieves the Raman-like spectrum from CARS spectra through training of simulated noisy CARS spectra, without the need for an NRB reference measurement. VECTOR is comprised of an encoder and a decoder. The encoder aims to compress the input to a lower dimensional latent representation without losing critical information. The decoder learns to reconstruct the input from the compressed representation. We also introduce skip connection bypasses from the encoder to the decoder, which benefits the reconstruction performance for deeper networks. We conduct abundant experiments to compare our proposed VECTOR to previous approaches in the literature, including the widely applied Kramers-Kronig method, as well as two another recently proposed methods that also use neural networks.

1 Introduction

Coherent Raman scattering (CRS) microscopies and spectroscopies have long sought to acquire the same chemical-rich information as traditional spontaneous Raman methods but orders-of-magnitude faster. Such a transformative technology would enable a significant expansion of application, such as live-cell and large-area imaging. Broadly speaking, the two dominant CRS modalities based on either stimulated Raman scattering (SRS) or coherent anti-Stokes Raman scattering (CARS) have either surpassed traditional Raman spectroscopy/imaging in speed or matched its spectroscopic bandwidth and sensitivity, but not both simultaneously. In this work, we focus on broadband coherent anti-Stokes Raman scattering (B-CARS), a method capable of probing over 4000 cm^{-1} bandwidth. Like all CARS-based methods, though, the signal is affected by a co-generated coherent background signal, the so-called non-resonant background (NRB). A limiting factor of CARS is that the third-order susceptibility term, from which it derives the frequency domain response, has a resonant and non-resonant component, with respect to the Raman-active transition frequencies $\chi^{(3)} = \chi_R + \chi_{NR}$. The intensity of the CARS spectrum is quadratically proportional to the susceptibility term:

$$I_{CARS} \propto |\chi_R^2| + |\chi_{NR}^2| + 2\chi_{NR} Re[\chi_R] \tag{1}$$

Though much maligned, the NRB is a stable homodyne amplifier thus partially responsible for the signal strength of CARS methods [1], which can also be used as an internal reference [2]. The side effect, though, is a significant perturbation of the recorded spectral shapes. If the NRB could be independently measured, it could be removed with analytical methods [3], though this is not currently possible; thus, surrogate materials have traditionally been utilized that contain weak or no vibrational peaks within swaths of the spectroscopic window. Though recent work has mitigated some of the ramifications [2], removing the necessity of any NRB estimate while still extracting undistorted Raman features would be superior. In this work, we present a new deep learning method, VECTOR, based upon an autoencoder topology with the addition of skip connections that provides similar results to analytical methods with NRB estimates and superior results over an alternative neural network approach.

Many NRB removal techniques exist, both as experimental techniques[4, 5, 6, 7], and as post-processing approaches[2, 8, 3, 9]. The majority of approaches seek to suppress or eliminate the non-resonant contribution, so that the observed spectrum obtains the form of a Raman-like spectrum. Experimental approaches have the benefit of directly collecting a purely resonant signal at the sample level, but suffer from weak signal, added experimental complexity and analysis or reduced throughput owing to requiring multiple measurements per area-of-interest. Post-processing approaches use physics or information theoretic approaches to mathematically extract the imaginary component of the CARS susceptibility χ_R, as it takes the form of a Raman-like spectrum. These approaches, however, necessitate an independent measurement of the NRB, which is not currently possible; thus, reference materials such as water, glass slide, or salt[10] is used, which contain weak or isolated vibrational signatures over certain regions of the Raman spectrum.

Deep Neural Networks (DNNs) have achieved striking results in different areas such as natural language processing [11], and computer vision [12]. Recently, deep learning approaches have been employed to tackle the issue of NRB removal [13, 14]. Houhou *et al.* [14] used a long short-term memory (LSTM) topology while Valensise *et al.* [13] implemented a convolutional neural network (CNN), termed "SpecNet", for NRB removal in CARS. These developments represent a significant step forward in rapid extraction of Raman-like spectra from B-CARS measurements.

A CNN makes use of several "filtering" layers for which the weights that connect these layers are learned using the method of back propagation. CNNs have been shown to be particularly effective in learning spatial features in images. An LSTM is a sub class of Recurrent Neural Network (RNN), where the network is trained by looping through sequences of data as functions of time. CNNs employ filters within convolutional layers to transform data, while LSTMs/RNNs are predictive in nature, reusing activation functions from other data points in the sequence in order to generate the next output in a series. Convolutional Autoencoders use CNNs to form each of an encoder and decoder pair, which encodes the input signal into a lower dimensional latent space and then decode it to back to its original value.

In this study, we provide comprehensive experiments to not only show the superior performance of our proposed approach but also analyze the critical information for successfully training a DNN. For the first time, we investigate the use of Convolutional Autoencoders (CAE), which belong to the Deep Generative Models (DGMs) family, for removing the NRB from CARS. A DGM is a powerful model of learning any kind of data distribution, and it has achieved tremendous success in the past few years in the form of Autoencoders (AE) [15], Variational Autoencoders (VAE) [16], and Generative Adversarial Networks (GANs) [17, 18]. Regarding our case, we seek to solve the problem by finding a 'mapping' between a noisy CARS spectrum and the underlying Raman spectrum.

Based on previous successful deployment of using AE for denoising images [19] and speech signals [20], we explore the use of AE for extraction of the Raman-like signal from CARS in this work.

The contributions in this paper are summarized as follows: First, we propose a **V**ery d**E**ep **C**onvolutional au**TO**encode**R** (VECTOR) architecture specifically designed for removing the NRB in CARS. Extensive experiments were performed to identify the optimal hyperparameters of VECTOR. We also demonstrate that skip

connections are not only able to boost performance for deeper networks but also to speed up the training process. Second, training was performed for nine different datasets with increasing complexity in terms of the number of peaks and the range of peak width. These datasets have different spectral shapes that emulate different applications of Raman spectroscopy, with the most complex dataset representative of biochemical spectra, and the simplest representative of pure chemical spectra. We demonstrate for the first time that the performance of deep learning based NRB removal correlates with spectral complexity. Third, we demonstrate significantly superior performance of VECTOR when compared with a simpler architecture that was recently proposed for NRB removal for all of the datasets tested. Lastly, VECTOR is applied to an experimental B-CARS spectrum of glycerol, and the results are compared with the Kramers-Kronig (KK) method. These results highlight a shortcoming of the trained VECTOR networks in dealing with a real-world NRB profile, which is not ideally smooth, as for the cases of the NRB profiles used in training. This in turn points to new research directions for training future networks, which are discussed in Section 6.

The breakdown of the paper is as follows: Section 2 briefly reviews B-CARS in terms of the experimental optical recording system, as well as the current state-of-the art in numerical NRB removal. In Section 3, the VECTOR architecture is considered in detail, including the skip connections. Section 4 details how simulated data was generated to train the networks and presents an ablation study and cross-validation to find the optimal hyperparameters. In Section 5, the results are provided. This includes a quantitative comparison with another CNN over the aforementioned datasets, and experimental results on a glycerol B-CARS spectrum. Finally, in Section 6 we offer a conclusion with an emphasis on the direction of future work.

2 Broadband Coherent Anti-Stokes Raman Spectroscopy

2.1 Theory and System Design

Coherent anti-Stokes Raman scattering is a third-order nonlinear optical process in which a “pump" and a “Stokes" photon coherently excite a molecular vibration at their beat frequency from which a “probe" photon is able to inelastically scatter, gaining energy (blue shifting) equal to the vibrational frequency – the anti-Stokes photon – as depicted in Fig. 1(b). Many of the earliest CARS microscopy systems were narrowband (∼1 ps to 10 ps pulse duration), able to probe single vibrational modes at high-speed, but capturing hyperspectral imagery required slow, often unstable laser tuning. To capture single-shot spectra, one approach termed broadband or multiplex CARS, uses a broadband Stokes source to stimulate multiple vibrations simultaneously with the now broadband anti-Stokes signal captured on a spectrometer. Due to practical limitations in the intensity of laser light that can be applied to [biological] samples and detector technology, B-CARS methods collect images relatively slowly but spectra extremely quickly. It should be noted that in most implementations of CARS methods, the probe source and the pump source are the same (and narrowband) – sometimes specified as “interpulse” excitation or “two-color" CARS. It is also possible to have a degenerate pump and Stokes source, “intrapulse", “impulsive", or “three-color" CARS.

Figure 1(a) presents a simplified schematic of a basic B-CARS system. Here the Stokes source (supercontinuum) is necessarily broadband, the probe source is necessarily narrowband (i.e., it is a determining factor in the spectroscopic resolution), and the pump source is purposely left ambiguous. The broadband pulsed source and the narrowband probe are combined using a dichroic and focused on the sample. Though there is a spatial phase-matching condition for CARS excitation, the use of high numerical aperture (NA) objective lenses effectively presents a plethora of permutations of photon vectors; thus, enabling collinear excitation. The generated anti-Stokes light is collected with another objective lens and separated from the remaining excitation beam by low-pass dichroic filters and recorded with a spectrometer. It should be highlighted that “epi-detected" systems with a reflective geometry have also been developed as have systems with beam-scanning capabilities; though, sample raster scanning is the most prevalent setup.

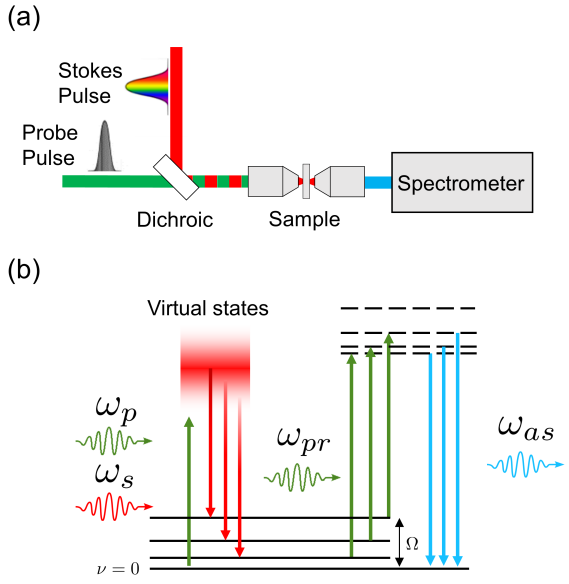

Figure 1: (a) Diagram of the setup of the B-CARS RMS, (b) CARS energy level diagram (ω_p: Pump frequency, ω_s: Stokes frequency, ω_{pr}: Probe frequency, ω_{as}: anti-Stokes frequency).

2.2 State-of-the-art for Non-Resonant Background Removal

There are two established methods for removing the NRB from a CARS spectrum: the KK method and the Maximum Entropy Method (MEM). The KK method utilizes a Hilbert transform to recover the susceptibility phase from the CARS spectrum and a reference NRB[3]. The Maximum Entropy Method is an information theoretic approach that attempts to maximize the spectral entropy of the susceptibility, given the measured constraints of the CARS spectrum[21]. These post processing methods used to remove the NRB contribution from CARS signals take a reference NRB signal and the resonant CARS signal as inputs.

As χ_{NR} cannot be readily extracted from a CARS spectrum, a CARS signal from a sample with no active Raman transitions in the region of interest is obtained. Glass or water are commonly used for this purpose, as they can be easily measured alongside the sample of interest. However, these NRB spectra are not perfect analogies for the non-resonant component of the sample CARS signal as they do contain some Raman vibrational features. This discrepancy is the major cause of error in the KK and MEM methods. The KK method is briefly reviewed in the following subsection. The KK method was chosen as the current best practice for analytical NRB removal since it requires less parameter optimization in the computation compared to the MEM. The KK method is also known to be more computationally efficient than the MEM approach[22], which will be pertinent for high-speed applications of B-CARS.

The KK relations generally relate the real and imaginary parts of any complex quantity which describes a causal (analytic) system. Under certain conditions, the KK relations may be extended to relate the intensity modulus and phase[23, 24]. It has been shown that the KK relation could be applied to CARS[3], enabling the retrieval of the spectral phase of the susceptibility (ϕ) from the CARS intensity:

$$\varphi(\omega) = -\frac{P}{\pi}\int_{-\infty}^{+\infty}\frac{\ln|\chi^{(3)}(\omega')|}{\omega'-\omega}\,\mathrm{d}\omega', \tag{2}$$

where P is the Cauchy principle value.

In this work, we have implement this method from Camp et al.[2] and apply wavelet denoising and a baseline correction using asymmetrically reweighted penalized least squares (arPLS) smoothing [25] on the phase prior to obtaining the complex quantity through Eq. 3.

$$\chi^{(3)} = \sqrt{I_{\mathrm{CARS}}}\exp\left[i\varphi(\omega)\right] \tag{3}$$

3 Very Deep Convolutional Autoencoders

The Autoencoder (AE) learns how to efficiently encode the input and learns how to reconstruct the input back from the reduced representation. Its ability to reduce data dimensions by learning how to ignore the noise in the data by design, has precipitated extensive study in the area of speech signals [20], computer vision [15, 19]. A typical AE normally comprises three main components (see Fig. 2): 1. The encoder, in which the model learns how to reduce the input dimensions and compress input data into an encoded representation; 2. A latent space or bottleneck, which is the layer that contains the compressed representation of the input data. This is the lowest possible dimension among the AE; 3. The decoder, in which the model learns how to reconstruct the input data from the encoded representation. The ordinary AE incorporates all fully-connected layers for both encoder and decoder [26]. However, training a fully-connected deep neural networks suffers from long computation times, over-fitting, and being difficult to optimize [27] especially for high dimensional input data. CNNs [28] were proposed to mitigate the problems above by introducing sparse connections among neurons. In terms of our case, the input data has a very high dimension i.e., length of 1000. We adopt Convolutional Autoencoders (CAEs) instead of fully-connected AE in this work as our backbone.

3.1 Architecture

The proposed VECTOR (**V**ery d**E**ep **C**onvolutional Au**TO**encode**R**s) consists of an encoder and a decoder as seen in Fig. 2. The structure of the encoder and the decoder is symmetric. This design will enable conducting a symmetric skip connection [12, 19, 20] between each paired convolutional layer and transposed convolutional layer in the encoder and the decoder respectively. We will introduce the benefits of employing skip connections in Section 3.2. Here we present notations used in this section: N is the batch size, K is the size of the 1D convolutional kernel, C is number of channels, S is the stride number, T is the input or feature length and $\mathbf{X} \in \mathbb{R}^{N\times T}$ is the input.

The encoder and the decoder are fully convolutional (1D) and fully transposed convolutional (1D) respectively. Batch normalization [29, 30] and ReLU [31] are added to each convolutional layer and transposed convolutional layer except for the last layer in the decoder. The encoder acts as a feature extractor that preserves primary information and properties of the input CARS spectra. To this end, the input is compressed to the latent representation (as seen in the middle of the architecture in Fig. 2). This compression process could have two benefits: 1. The latent representation holds as much critical information as input contains but in a much lower dimension; 2. The compressed representation removes noise and nonessential information contained in the input. The decoder is then combined to recover details of input contents subject to the corresponding set of 'clean' Raman spectra. We use the Mean Absolute Error (MAE), also known as the $\mathcal{L}_1$ norm, between CARS spectra and clean spectra as the loss (see Equation (4)) to train the VECTOR network in an

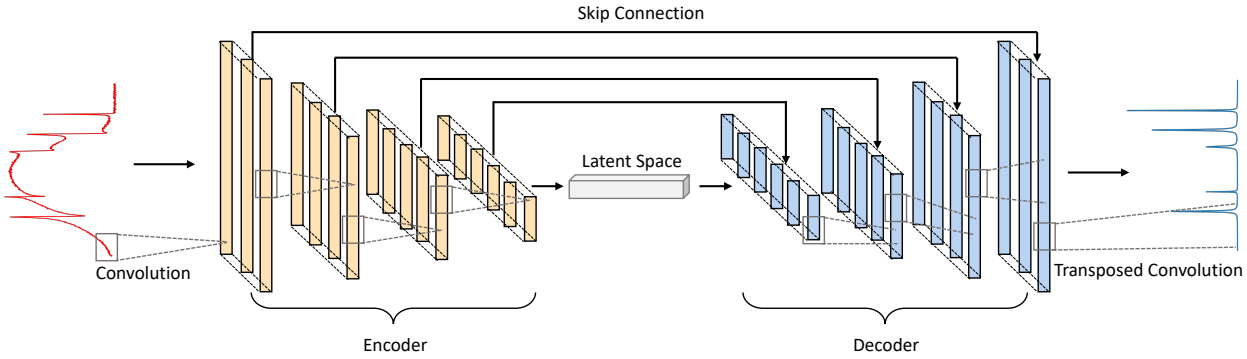

Figure 2: The example of VECTOR-8 architecture used in this study. Convolutional layers in the encoder and transposed convolutional layers in the decoder are symmetric i.e., the input dimension to the convolutional layer and the output dimension to the paired transposed convolutional layer are the same.

end-to-end manner.

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i}^{N} |f(\mathbf{X}_i;\theta) - \mathbf{Y}_i|, \tag{4}$$

where $\mathbf{X}$ is CARS input spectra, $\mathbf{Y}$ is Raman output spectra, f is the network, θ is the network weight.

Moreover, skip connections are also added from each convolutional layer to its corresponding paired transposed convolutional layer. The convolutional feature maps are passed to, and summed to, the transposed convolutional feature maps in an element-wise manner because those two feature maps have the exact same dimension. Details of VECTOR-8 are provided in Table 1 and the details of the other VECTOR architectures are provided in Appendix A.

3.2 Skip Connections

Skip connections have been widely applied in the area of computer vision. The most well known CNN architecture with skip connection is ResNet [12]. Fig. 3 shows an example of skip connection. The output of a residual block

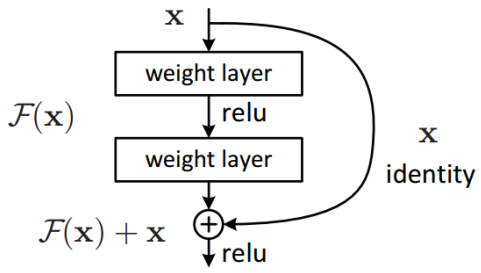

Figure 3: Skip connection in residual block [12].

can be easily formulated as $\mathbf{y} := F(\mathbf{x}) + \mathbf{x}$. He *et al.* [12] showed that skip connections are easier to optimize compared to those 'plain' networks without skip connections, in which skip connections mitigate the vanishing gradient descent to bottom layers [32] and skip connections are able to boost performances for the networks with greatly increased depth.

The vanishing gradient is a well known problem in training neural networks. Gradients are used to update the network weights and are in the range (0,1]. Backpropagation computes gradients by the chain rule, which the effect of multiplying n of these small numbers to compute gradients of the early layers in an n-layer network, meaning that the gradient (error signal) decreases exponentially with n while the early layers train slowly. Skip connections offer a solution to this problem by connecting the early layers to the later layers, which enables gradients to pass to early layers directly, instead of passing n layers.

Apart from the vanishing gradient problem, the growing depth of the AE architecture could also lead to the critical input information being lost such that they can not be recovered via the encoding/decoding process transposed convolution. Skip connections also enable the recovery of such information.

To overcome this issue, we design the VECTOR network in a symmetric manner i.e., the convolutional layer in the encoder and the transposed convolutional layer in the decoder are paired with each other. This design allows connections bypassing from one convolutional layer to the paired transposed convolutional layer. An additional benefit of the symmetrical skip connection is that padding can be avoided. Fig. 2 shows an example of skip connection for VECTOR-8. In this work, the number of skip connections is equal the the number of layers in the encoder/decoder. To address benefits of using skip connection, we will show that skip connections are able to 1. speed-up the training and 2. boost the performance for the deeper networks compared to the 'plain' networks in Section 4.3.

Table 1: The architecture of VECTOR-8. The operation refers to 1D convolution and 1D transposed convolution for encoder and decoder respectively, where K is the kernel size, C_{in} represents number of input channels, C_{out} represents number of output channels and S refers to the stride. The output size refers to the dimension of feature maps produced by each stage, where T is the length and C is the number of channels. Rest of VECTOR models can be referred to *Appendix A*.

Stage		Operation (K, C_{in}, C_{out}, S)	Output size ($T \times C$)
Input		-	1000×1
Encoder	layer 1	8, 1, 64, 1	993×64
	layer 2	8, 64, 128, 2	493×128
	layer 3	8, 128, 256, 2	243×256
	layer 4	8, 256, 512, 2	118×512
Latent space		-	118×512
Decoder	layer 1	8, 512, 256, 2	243×256
	layer 2	8, 256, 128, 2	493×128
	layer 3	8, 128, 64, 2	993×64
	layer 4	8, 64, 1, 1	1000×1
Output layer		Sigmoid	1000×1

4 Training

4.1 Datasets for Training

VECTOR was trained by a dataset of simulated CARS spectra. The efficacy of the algorithm was tested by training it with a number of different datasets of varying complexity. The datasets were varied based on the number of Raman peaks per spectrum, and the variation in full-width-half-maximum (FWHM) of the peaks themselves. This is principally of interest owing to the different kinds of Raman spectra found in different applications. Chemical and pharmaceutical specimens have sharper peaks qualitatively observed as distinct peaks[33, 34], in comparison to biological samples, for example cellular spectra, which have broader peaks and appear as more complex amalgam of broad to narrow peaks[35, 36].

The code used to generate the simulated datasets was adapted from [13], which we detail below. Further details are also given in Appendix B. The CARS spectra were generated as per Equation 1. The resonant susceptibility is expressed as a sum of Lorentzian functions, as per its physical basis, see Equation 5. For each Lorentz function, the peak amplitude A is uniform random value between 0 and 1, with its resonant peak at Ω between 300 cm^{-1} and 1700 cm^{-1} with FWHM $2\times\Gamma$. Nine datasets were generated of varying ranges of peaks widths and number of peaks per spectrum, designated (i - ix). The spectra are generated as 1000 datapoints across 2000 cm^{-1}. FWHM vary from low (2-10 cm^{-1}) in (i,ii,iii) , to moderate (2-25 cm^{-1})in (iv,v,vi), to high (2-75 cm^{-1}) in (vii,viii,ix). The datasets are further split into three sections by number of spectra. Each spectrum generated in datasets (i,iv,vii) have between 1 and 15 peaks. Similarly (ii,v,viii) contain between 15-30 peaks per spectrum, and (iii,vi,ix) have between 30-50 peaks.The non-resonant susceptibility is typically expressed as an arbitrary, slowly varying function. In this case, it is expressed as the product of two randomised, countervailing sigmoid functions, per Equation 6.

$$\chi_R(\omega) = \sum_{N_{peaks}} \frac{A}{\Omega - \omega - i\Gamma} \tag{5}$$

$$\chi_{NR}(\omega) = \frac{e^{-c_3(\omega - c_4)}}{(1 + e^{-c_1(\omega - c_2)})(1 + e^{-c_3(\omega - c_4)})} \tag{6}$$

A training set of 200,000 spectra and an evaluation set of 30,000 spectra was generated for each dataset. Alongside the CARS spectrum, a corresponding Raman spectrum is created to function as an ideal reference to train the network.

4.2 Implementation Details

Experiments were conducted on nine synthetic datasets using the formulation presented in Section 4.1. When comparing to previous studies [13], we select VECTOR-16 by default as an acceptable trade-off between computation time and performance, more details on this selection are given in Section 4.3. The models were trained on one TITAN Xp GPU. We adopted Stochastic Gradient Descent (SGD) as an optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. The batch size was set to 256 for the training. The initial learning rate was 0.1 and was reduced by a factor of 10 at 25, 50, 75 epochs. Training was stopped at 100 epochs.

4.3 Model Dimensionality

Fig. 4 compares the performance of six architectures of VECTOR with different numbers of layers including (red) and excluding (blue) skip connections. The MAE between the synthetic Raman spectrum and the recovered Raman spectrum from the corresponding CARS spectrum in the validation datasets was used as the performance metric for this study. It should be noted that architectures without skip connections achieve comparable performance for lower

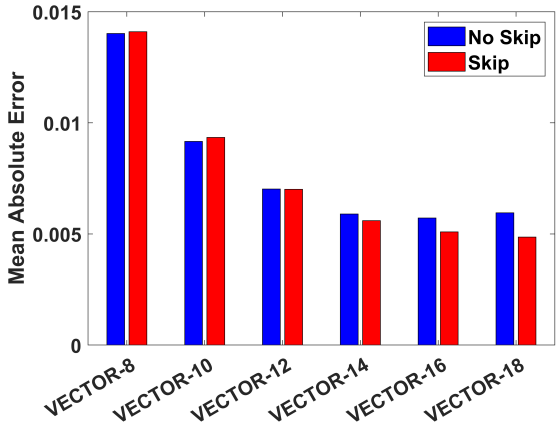

Figure 4: MAE performance of VECTOR for different encoder-decoder depths,applied to the most complex dataset (dataset ix) .

sets of layers i.e., VECTOR-8, VECTOR-10, VECTOR-12. Starting from VECTOR-14, performance of architectures without skip connection saturates, which indicates that the critical information contained in the bottom layers gets lost and does not pass to the transposed convolutional layers successfully. For VECTOR-18, the performance is even worse than VECTOR-14 and VECTOR-16, which indicates the vanishing gradient problem may arise for the bottom layers leading to the bottom layer weights receiving very small gradient and has not been fully optimized. Increasing number of layers of the VECTOR will increase

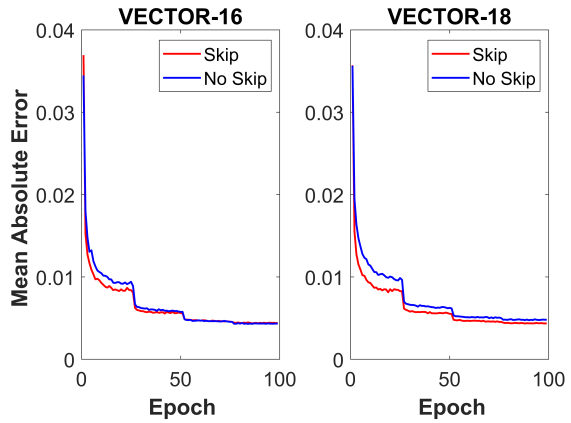

Figure 5: Training loss for VECTOR-16 and VECTOR-18 with and without skip connections.

computation times leading to slower training but only provide very limited performance enhancement. As such, VECTOR-16 was determined to be optimal for our purposes. Shown in Fig. 5 is the training loss for VECTOR-16 and VECTOR-18 with and without skip connections. It can seen that skip connections are able to accelerate the training process especially for the first 25 epochs. This is most pronounced in VECTOR-18 where the skip connection architecture produces consistently better performance throughout all training epochs.

5 Results

5.1 Performance on Simulated Datasets

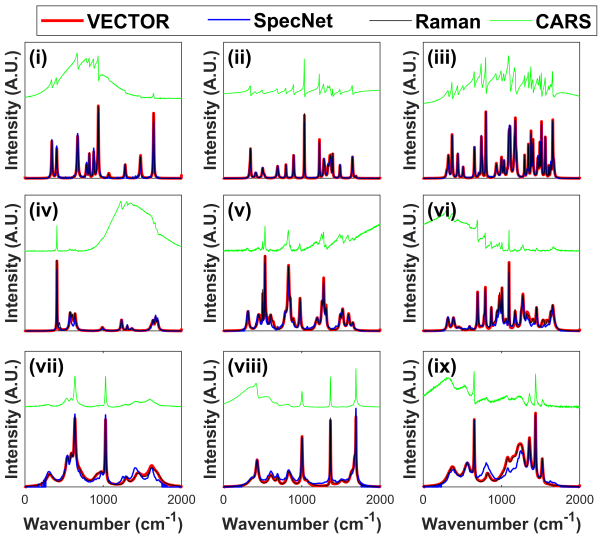

Figure 6: Example of recovered spectra for VECTOR-16 (red) and SpecNet architecture (blue) for each algorithm trained and evaluated on datasets (i-ix). True Raman spectrum is overlaid (black) and input CARS spectrum (green) is plotted with offset.

We trained and performed validation on nine different networks, where each network was trained/validated using the nine different training/validation datasets described in Section 4.1. These nine networks were then tested using nine test datasets of size 4096 spectra. The test dataset used for each network matched the parameters of the training/validation datasets for that given network in terms of peak number and width.

The SpecNet network[13] was trained/validated using the same nine datasets as described in Section C, and were then subject to the identical test datasets. In contrast only a single pre-trained network from SpecNet[13] was tested on the same nine test datasets. This network was provided by the authors having been trained using up to 15 peaks of width 1-20 cm^{-1}. Visualizations of spectra recovered by VECTOR-16 and the SpecNet network are shown in Fig.6. Fig. 7 illustrates the average MAE (as defined in Equation (4)) for both networks on each dataset. It can be noticed that the VECTOR network significantly outperforms SpecNet for all nine datasets. For both networks, MAE increases proportionally with number of peaks. Datasets (i, iv, vii) have between 1-15 peaks, datasets (ii,v,viii) between 15-30, and datasets (iii,vi,ix) have between 30-50 peaks per spectrum (these correspond to left, centre and right columns in Fig. 6). Similarly, an increase in the ranges of

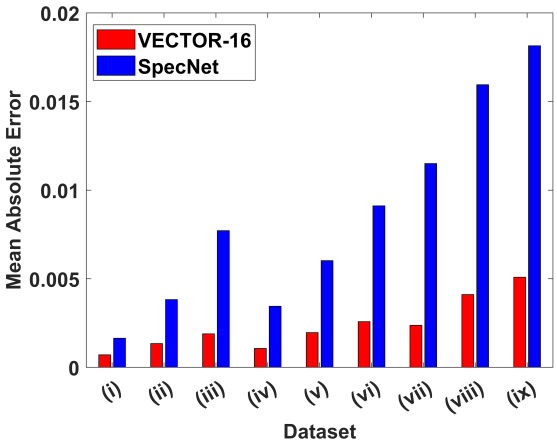

Figure 7: Average MAE from each of the datasets.

peak widths produces an increase in MAE. Datasets (i-iii) contain narrow peak widths between 2-10 cm^{-1}, widening to between 2-25 cm^{-1} for datasets (iv-vi), with the widest range of 2-75 cm^{-1} in the case of datasets (vii-ix) (corresponding to top, middle and bottom rows in Fig. 6). MAE increases most sharply for the most complex datasets containing the broadest range of peaks. The relative change in MAE is broadly consistent across datasets for both networks. Valensise et al. trained their network on spectra between 1-15 peaks and between 1-20 cm^{-1}, which is most closely matched by dataset (iv), and for this case it is outperformed by our network by a factor of 4. Similar out-performance is seen for all datasets, though it must be noted that SpecNet cannot be expected to perform well on datasets that are significantly different to those on which it was trained, in particular for the case of datasets (vi-ix) for which the peak width has the greatest variation. For more complex datasets such as dataset ix, the improvement of VECTOR-16 is much more significant.

Both Fig. 6 and Fig. 7 clearly demonstrate that the VECTOR-16 network is able to recover spectra with high quality for the case of all nine datasets. For datasets (i-ii) both SpecNet and our three individual networks perform well in recovering peak position and shape. However, close inspection of the peak values reveals that VECTOR is more successful at recovering peak height and also at removing noise in the baseline. For the case of datasets (vii-ix) SpecNet returns erroneous values at the ends of the spectrum, likely due to the fact that this network has not been trained on data of this nature. It is likely that this is the same reason that VECTOR evidently performs so much better at retrieving the broader baseline in these cases.

We also highlight the capability of the VECTOR network to reduce the effect of noise in the CARS intensity. Fig.8 shows the result of applying VECTOR-16 (trained on dataset ix) for two test cases also corresponding to dataset ix. In the first case, on the left side of the figure, we see the result of processing a high SNR CARS spectrum, with high quality output as expected. In the second case on the right a lower SNR signal is processed. In this case, the random noise fluctuations are clearly seen in the input CARS spectrum. The recovered Raman spectrum is high quality and contains no such noisy signal. However, the insert in the figure highlights that small spectral features, with similar amplitude to the noise signal have been lost. These results show that the VECTOR network is robust to handle low SNR inputs, with the caveat that spectral features matching the noise floor may be lost. However, dealing with very low SNR signals would require further training with datasets of similar quality. A more detailed discussion of SNR is provided in Section D.

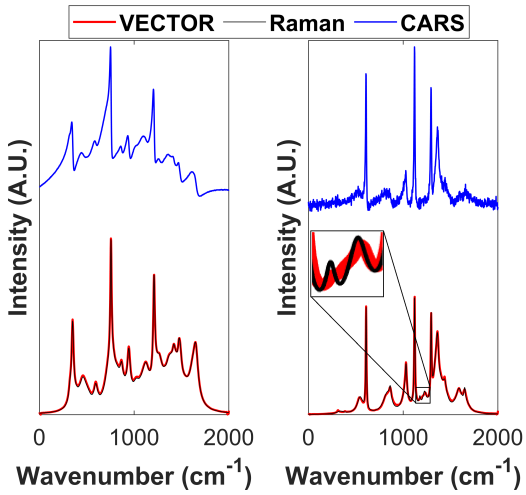

Figure 8: An example of high and low SNR B-CARS spectra processed by VECTOR-16 trained on dataset ix. Although the network effectively removes the noise in the retrieved Raman spectrum, the insert on the right figure highlights that spectral features under the noise floor may be lost.

5.2 Performance on Experimental Data

A B-CARS spectrum of glycerol was recorded using an Er-Fibre system as described in Section 2.1. Neat glycerol was applied into a 120 μm thick imaging spacer sandwiched between a microscope slide and coverslip glass. A spectrum of a glass coverslip was also recorded, as an NRB reference spectrum. The spectrum was processed independently by VECTOR-16, trained on datasets (iii), (vi) and (ix). The recovered spectrum is shown in Fig. 9. While VECTOR-16 shows robust performance for simulated datasets, performance with real spectra is found to be sensitive to the training data used for the network.

The form of the NRB seen in an Er-Fibre CARS system is not simply modeled as a slowly varying polynomial such as the countervailing sigmoids used here, as is more appropriate for a conventional two-colour broadband CARS platform. The combination of two-colour and three-colour excitation domains produces a complex NRB signal that is highly subject to the laser properties, such as spectral

envelope and phase delay. This is particularly difficult for an NRB agnostic system such as VECTOR, in comparison to KK or MEM, which incorporate an NRB reference in their process. As a result, the high degree of variation in the NRB intensity means the peak heights of the C-H band in glycerol are not as intense as would be expected, as achieved by KK. As a result, components of the NRB profile are interpreted by the network as resonant features, and attempts to construct peaks, as can be seen most acutely in datasets vi and ix, which are trained on broader sets of peak widths. However, VECTOR does not produce many

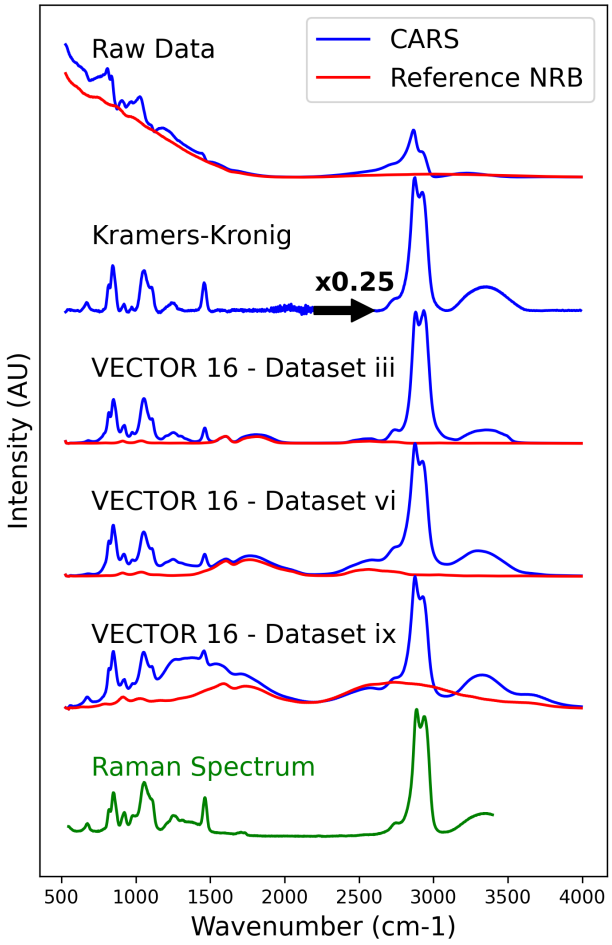

Figure 9: Recovery of experimental CARS spectrum. The top spectrum shows the CARS spectrum recorded from glycerol, and a corresponding NRB reference spectrum of glass. The successive spectra shown are the recovered Raman-like spectra from the KK method, followed by VECTOR-16 trained on different dataset configurations, all shown in blue. The NRB spectra was also processed for the case of VECTOR-16 and is shown in red.

of the artifacts of the KK method common to low intensity regions, typically the Raman silent region, where quotients of small values produce noisy results. VECTOR also seeks to smooth the extracted Raman-like output.

For comparison a Raman spectrum of glycerol was also recorded and is shown in green in Fig. 9. This spectrum was recorded using a coomercial Raman microspectrometer (Horiba Jobin Yvon LabRAM HR, grating 300 lines/mm, excitation 532 nm). The spectrum was wavenumber calibrated using 4-Acetamidophenol (Sigma) and intensity calibrated using a NIST calibrated White-Light source (Ocean-Optics) as described in Ref.[37]. Interestingly the spectrum recovered from the various VECTOR architectures appears to be more consistent regarding the relative intensity of the peaks in the higher wavenumbers.

6 Conclusion

In this paper, we have proposed a general architecture called VECTOR specifically designed for removing the NRB from CARS spectra. The VECTOR network is symmetric and consists of an encoder and decoder. Each convolutional layer and transposed convolutional layer are paired to each other. More importantly, the VECTOR network contains the skip connection from one convolutional layer and the paired transposed convolutional layer. We design abundant experiments to demonstrate the skip connection is able to help on recovering clean spectra and tackles the optimization difficulty caused by vanishing gradient for deeper networks. The proposed VECTOR network outperforms the previous works on recovering the clean spectra and is also robust to low SNR input spectra.

Compared to the traditional AE (Autoencoder), we adopt two critical designs in the VECTOR: 1. Replace fully connected layers by convolutional layers; 2. Skip connections are bypassed from a convolutional layer to a paired transposed convolutional layer. We design ablation studies to demonstrate that skip connections are able to improve the reconstruct performance and speed up the training. The key difference between VECTOR and CNNs is that VECTOR is symmetric. It contains a symmetric encoder-decoder architecture to reconstruct spectra in a "squeeze-and-unsqueeze" manner. This symmetric design enables the skip connection between paired convolutional layer and transposed convolutional layer. In Fig. 4, we have shown that deeper VECTOR has better reconstruction performance. However, we only investigate the impact of the network depth scaling in this study. Other expansion for networks such as width scaling, resolution scaling and compound scaling [38] can be further investigated beyond the VECTOR network for CARS-related research.

We have shown the superior performance of our VECTOR compared to the previous CNNs approach [13] for all nine datasets. We trained our network on different configurations of simulated spectra, which demonstrated the importance of training the network for the specific types of spectra that it is intended to be used on. While the network created by Valensise et al. [13] performed well on the simulated datasets that were similar or simpler to their training set (i-iv), its qualitative and quantitative performance suf-

fered from more complex datasets, as seen in Fig.6 and Fig.7.

Although VECTOR performs well when processing synthetic spectra that is used to train it, further work is required for use with real-world experimental data. The use of experimental spectra would be cumbersome for training due to the large volume of data required to train the network, and therefore, improvements must be made to the training set to better emulate the spectra recorded from the CARS platform. Constraining and tailoring the peak shapes, heights and frequencies to emulate those observed in the fingerprint and C-H stretching regions of the spectrum is a natural next step. This would be achieved by a wider range of peak amplitudes, SNR ranges, and generating taller, sharper peaks in the C-H range with smaller, denser, wider peaks in the fingerprint region. At present the double sigmoid function used to model the slowly varying frequency response of the NRB is not sufficient to model a three-colour CARS system. In order to train VECTOR on data that better emulates the types of experimental data seen in 5.2, rather than use an arbitrary function it may be advantageous to model the excitation profile of the specific laser system to create the two and three colour NRB domains. In this configuration the excitation profile would dominate over the NRB shape, assuming it is non-zero across the frequency domain, as is physically appropriate.

An important point of discussion is the relationship between the proposed architecture (VECTOR) and U-Net [39], which is a convolutional neural network developed for 2-D image segmentation, and which has become the gold-standard in this area. VECTOR and U-Net are both specific types of the more general AE architecture [40, 41] for which the encoder learns an efficient representation of unlabelled data in the form of key features, which can then be used by the decoder to reconstruct the data in the absence of unwanted data such as noise. Both VECTOR and U-Net use an "hourglass architecture," and both use skip connections. The skip in VECTOR is different from that used in U-Net. Our skip is similar to the ResNet skip connection [12] in which the original input data is also added to the output of the convolution block, while U-Net concatenates feature maps through the channel dimension. The skip connection in VECTOR can be described as (a+b), compared to U-Net's [a, b]. Another distinguishing feature is that U-Net uses max pooling, while VECTOR uses striding convolution.

In this paper, we have successfully demonstrated the application of the AE architecture as an effective method to solve the CARS problem for the first time. Further, we have demonstrated the use of skip connections to boost the performance of deeper networks. This work sets the baseline for autoencoder research in this application area and any future research can incrementally build on this work. Examples of future work could include an investigation of U-Net (it should be noted that our VECTOR architecture is easily adapted to U-Net), attention mechanism [42], transformer [42]etc., all of which belong to the autoencoder family.

7 Acknowledgements

It should be noted that the first two authors listed on this article contributed equally to the work presented. We acknowledge the support of Science Foundation Ireland under Grant Numbers 15/CDA/3667, 19/FFP/7025, and 16/RI/3399. We wish to thank Swarnavo Sarkar, Michael Majurski, Nancy Lin, Sheng Lin-Gibson, and Donald Atha of the National Institute of Standards and Technology for their helpful discussions and insights towards the preparation of this manuscript.

8 Disclaimer

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

[1] Müller M, Zumbusch A. Coherent anti-Stokes Raman Scattering Microscopy. ChemPhysChem. 2007;8(15):2156–2170. doi:10.1002/cphc.200700202.

[2] Camp CH, Lee YJ, , Cicerone MT. Quantitative, Comparable Coherent Anti-Stokes Raman Scattering (CARS) Spectroscopy: Correcting Errors in Phase Retrieval. Journal of Raman spectroscopy : JRS. 2016;47(4):408–415. doi:10.1002/jrs.4824.

[3] Liu Y, Lee YJ, Cicerone MT. Broadband CARS spectral phase retrieval using a time-domain Kramers–Kronig transform. Optics Letters. 2009;34(9):1363. doi:10.1364/OL.34.001363.

[4] Cheng JX, Book LD, Xie XS. Polarization coherent anti-Stokes Raman scattering microscopy. Optics Letters. 2001;26(17):1341. doi:10.1364/OL.26.001341.

[5] Potma EO, Evans CL, Xie XS. Heterodyne coherent anti-Stokes Raman scattering (CARS) imaging. Optics Letters. 2006;31(2):241. doi:10.1364/OL.31.000241.

[6] Littleton B, Kavanagh T, Festy F, Richards D. Spectral Interferometric Implementation with Passive Polarization Optics of Coherent Anti-Stokes Raman Scattering. Physical Review Letters. 2013;111(10):103902. doi:10.1103/PhysRevLett.111.103902.

[7] Volkmer A, Book LD, Xie XS. Time-resolved coherent anti-Stokes Raman scattering microscopy:

Imaging based on Raman free induction decay. Applied Physics Letters. 2002;80(9):1505–1507. doi:10.1063/1.1456262.

[8] Vartiainen EM, Rinia HA, Müller M, Bonn M. Direct extraction of Raman line-shapes from congested CARS spectra. Optics Express. 2006;14(8):3622. doi:10.1364/OE.14.003622.

[9] Okuno M, Kano H, Leproux P, Couderc V, Day JPR, Bonn M, et al. Quantitative CARS Molecular Fingerprinting of Single Living Cells with the Use of the Maximum Entropy Method. Angewandte Chemie International Edition. 2010;49(38):6773–6777. doi:10.1002/anie.201001560.

[10] Karuna A, Masia F, Borri P, Langbein W. Hyperspectral volumetric coherent anti-Stokes Raman scattering microscopy: quantitative volume determination and NaCl as non-resonant standard. Journal of Raman Spectroscopy. 2016;47(9):1167–1173. doi:https://doi.org/10.1002/jrs.4876.

[11] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018;.

[12] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society; 2016. p. 770–778. Available from: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.90.

[13] Valensise CM, Giuseppi A, Vernuccio F, De la Cadena A, Cerullo G, Polli D. Removing non-resonant background from CARS spectra via deep learning. APL Photonics. 2020;5(6):061305. doi:10.1063/5.0007821.

[14] Houhou R, Barman P, Schmitt M, Meyer T, Popp J, Bocklitz T. Deep learning as phase retrieval tool for CARS spectra. Optics Express. 2020;28(14):21002. doi:10.1364/OE.390413.

[15] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA, Bottou L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research. 2010;11(12).

[16] Kingma DP, Welling M. Auto-encoding variational bayes. arXiv preprint arXiv:13126114. 2013;.

[17] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. arXiv preprint arXiv:14062661. 2014;.

[18] Wang Z, She Q, Ward TE. Generative adversarial networks in computer vision: A survey and taxonomy. arXiv preprint arXiv:190601529. 2019;.

[19] Mao XJ, Shen C, Yang YB. Image restoration using convolutional auto-encoders with symmetric skip connections. arXiv preprint arXiv:160608921. 2016;.

[20] Park SR, Lee J. A fully convolutional neural network for speech enhancement. arXiv preprint arXiv:160907132. 2016;.

[21] Yang PK, Huang JY. Phase-retrieval problems in infrared–visible sum-frequency generation spectroscopy by the maximum-entropy method. Journal of the Optical Society of America B. 1997;14(10):2443. doi:10.1364/josab.14.002443.

[22] Cicerone MT, Aamer KA, Lee YJ, Vartiainen E. Maximum entropy and time-domain Kramers-Kronig phase retrieval approaches are functionally equivalent for CARS microspectroscopy. Journal of Raman Spectroscopy. 2012;43(5):637–643. doi:10.1002/jrs.3169.

[23] Toll JS. Causality and the Dispersion Relation: Logical Foundations. Physical Review. 1956;104(6):1760. doi:10.1103/PhysRev.104.1760.

[24] Smith DY. Dispersion relations for complex reflectivities*†. Journal of the Optical Society of America. 1977;67(4):570. doi:10.1364/josa.67.000570.

[25] Baek SJ, Park A, Ahn YJ, Choo J. Baseline correction using asymmetrically reweighted penalized least squares smoothing. The Analyst. 2015;140(1):250–257. doi:10.1039/c4an01061b.

[26] Schmidhuber J. Deep learning in neural networks: An overview. Neural networks. 2015;61:85–117.

[27] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. vol. 1. MIT press Cambridge; 2016.

[28] LeCun Y, Bengio Y, et al. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1995;3361(10):1995.

[29] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Bach F, Blei D, editors. Proceedings of the 32nd International Conference on Machine Learning. vol. 37 of Proceedings of Machine Learning Research. Lille, France: PMLR; 2015. p. 448–456. Available from: https://proceedings.mlr.press/v37/ioffe15.html.

[30] Santurkar S, Tsipras D, Ilyas A, Madry A. How does batch normalization help optimization? arXiv preprint arXiv:180511604. 2018;.

[31] Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. ICML'10. Madison, WI, USA: Omnipress; 2010. p. 807–814.

[32] Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 1998;6(02):107–116.

[33] Das RS, Agrawal YK. Raman spectroscopy: Recent advancements, techniques and applica-

tions. Vibrational Spectroscopy. 2011;57(2):163–176. doi:10.1016/j.vibspec.2011.08.003.

[34] Vankeirsbilck T, Vercauteren A, Baeyens W, Van der Weken G, Verpoort F, Vergote G, et al. Applications of Raman spectroscopy in pharmaceutical analysis. TrAC Trends in Analytical Chemistry. 2002;21(12):869–877. doi:10.1016/S0165-9936(02)01208-6.

[35] Krafft C, Schie IW, Meyer T, Schmitt M, Popp J. Developments in spontaneous and coherent Raman scattering microscopic imaging for biomedical applications. Chemical Society Reviews. 2016;45(7):1819–1849. doi:10.1039/C5CS00564G.

[36] Movasaghi Z, Rehman S, Rehman IU. Raman Spectroscopy of Biological Tissues. Applied Spectroscopy Reviews. 2007;42(5):493–541. doi:10.1080/05704920701551530.

[37] Hutsebaut D, Vandenabeele P, Moens L. Evaluation of an accurate calibration and spectral standardization procedure for Raman spectroscopy. Analyst. 2005;130(8):1204–1214.

[38] Tan M, Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th International Conference on Machine Learning. vol. 97 of Proceedings of Machine Learning Research. PMLR; 2019. p. 6105–6114. Available from: https://proceedings.mlr.press/v97/tan19a.html.

[39] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Cham: Springer International Publishing; 2015. p. 234–241.

[40] Shrestha A, Mahmood A. Review of deep learning algorithms and architectures. IEEE Access. 2019;7:53040–53065.

[41] Deng L. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing. 2014;3.

[42] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017.Available from: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[43] Barton SJ, Hennelly BM. Signal to noise ratio of Raman spectra of biological samples. In: Popp J, Tuchin VV, Pavone FS, editors. Biophotonics: Photonic Solutions for Better Health Care VI. vol. 10685. International Society for Optics and Photonics. SPIE; 2018. p. 698 – 708. Available from: https://doi.org/10.1117/12.2307591.

A Summaries for the rest of CAE models

Table 1: The architecture of VECTOR-10

Stage		Operation (K, C_{in}, C_{out}, S)	Output size ($T \times C$)
Input		-	1000×1
Encoder	layer 1	8, 1, 64, 1	993×64
	layer 2	8, 64, 128,2	493×128
	layer 3	8, 128, 256, 2	243×256
	layer 4	8, 256, 512, 2	118×512
	layer 5	8, 512, 1024, 2	56×1024
Latent space		-	56×1024
Decoder	layer 1	8, 1024, 512, 2	118×512
	layer 2	8, 512, 256, 2	243×256
	layer 3	8, 256, 128, 2	493×128
	layer 4	8, 128, 64, 2	993×64
	layer 5	8, 64, 1	1000×1
Output layer		Sigmoid	1000×1

Table 2: The architecture of VECTOR-12

Stage		Operation (K, C_{in}, C_{out}, S)	Output size ($T \times C$)
Input		-	1000×1
Encoder	layer 1	8, 1, 64, 1	993×64
	layer 2	8, 64, 128, 2	493×128
	layer 3	8, 128, 256, 2	243×256
	layer 4	8, 256, 512, 2	118×512
	layer 5	8, 512, 1024, 2	56×1024
	layer 6	8, 1024, 2048, 2	25×2048
Latent space		-	25×2048
Decoder	layer 1	8, 2048, 1024, 2	56×1024
	layer 2	8, 1024, 512, 2	118×512
	layer 3	8, 512, 256, 2	243×256
	layer 4	8, 256, 128, 2	493×128
	layer 5	8, 128, 64, 2	993×64
	layer 6	8, 64, 1, 1	1000×1
Output layer		Sigmoid	1000×1

Table 3: The architecture of VECTOR-14

Stage		Operation (K, C_{in}, C_{out}, S)	Output size ($T \times C$)
Input		-	1000×1
Encoder	layer 1	8, 1, 64, 1	993×64
	layer 2	8, 64, 128, 2	493×128
	layer 3	8, 128, 256, 2	243×256
	layer 4	8, 256, 512, 2	118×512
	layer 5	8, 512, 1024, 2	56×1024
	layer 6	8, 1024, 2048, 2	25×2048
	layer 7	8, 2048, 2048, 2	18×2048
Latent space		-	18×2048
Decoder	layer 1	8, 2048, 2048, 2	25×2048
	layer 2	8, 2048, 1024, 2	56×1024
	layer 3	8, 1024, 512, 2	118×512
	layer 4	8, 512, 256, 2	243×256
	layer 5	8, 256, 128, 2	493×128
	layer 6	8, 128, 64, 2	993×64
	layer 7	8, 64, 1, 1	1000×1
Output layer		Sigmoid	1000×1

Table 4: The architecture of VECTOR-16

Stage		Operation (K, C_{in}, C_{out}, S)	Output size ($T \times C$)
Input		-	1000×1
Encoder	layer 1	8, 1, 64, 1	993×64
	layer 2	8, 64, 128, 2	493×128
	layer 3	8, 128, 256, 2	243×256
	layer 4	8, 256, 512, 2	118×512
	layer 5	8, 512, 1024, 2	56×1024
	layer 6	8, 1024, 2048, 2	25×2048
	layer 7	8, 2048, 2048, 2	18×2048
	layer 8	8, 2048, 2048, 2	11×2048
Latent space		-	11×2048
Decoder	layer 1	8, 2048, 2048, 2	18×2048
	layer 2	8, 2048, 2048, 2	25×2048
	layer 3	8, 2048, 1024, 2	56×1024
	layer 4	8, 1024, 512, 2	118×512
	layer 5	8, 512, 256, 2	243×256
	layer 6	8, 256, 128, 2	493×128
	layer 7	8, 128, 64, 2	993×64
	layer 8	8, 64, 1, 1	1000×1
Output layer		Sigmoid	1000×1

Table 5: The architecture of VECTOR-18

Stage		Operation (K, C_{in}, C_{out}, S)	Output size ($T \times C$)
Input		-	1000×1
Encoder	layer 1	8, 1, 64, 1	993×64
	layer 2	8, 64, 128, 2	493×128
	layer 3	8, 128, 256, 2	243×256
	layer 4	8, 256, 512, 2	118×512
	layer 5	8, 512, 1024, 2	56×1024
	layer 6	8, 1024, 2048, 2	25×2048
	layer 7	8, 2048, 2048, 2	18×2048
	layer 8	8, 2048, 2048, 2	11×2048
	layer 9	8, 2048, 2048, 2	4×2048
Latent space		-	4×2048
Decoder	layer 1	8, 2048, 2048, 2	11×2048
	layer 2	8, 2048, 2048, 2	18×2048
	layer 3	8, 2048, 2048, 2	25×2048
	layer 4	8, 2048, 1024, 2	56×1024
	layer 5	8, 1024, 512, 2	118×512
	layer 6	8, 512, 256, 2	243×256
	layer 7	8, 256, 128, 2	493×128
	layer 8	8, 128, 64, 2	993×64
	layer 9	8, 64, 1	1000×1
Output layer		Sigmoid	1000×1

B Simulation of bCARS spectra

The simulated spectra used for training/validation in this paper were produced in an almost identical manner to those in Ref. [13]. Given that there are some difference, we provide a detailed overview of the simulation here.

We begin by first approximating the ideal Raman spectrum, which is equivalent to the imaginary part of the resonant susceptibility, $\chi_R^{(3)}$. The spectrum is defined as a function of wavenumber (ν) with units cm^{-1}, over the region from 0.1 to 2000 cm^{-1}, which approximately corresponds to the fingerprint region. The susceptibility is defined as a superposition of n Lorentzian functions:

$$\chi_R^{(3)}(\nu) = \sum_1^n \frac{A}{\nu - \Omega - j\Gamma} \tag{7}$$

where n is the total number of resonances in the CARS spectrum and which varies randomly across the datasets; A is the resonance amplitude; Ω is the resonance position; and Γ is the half-width. These spectral parameters are generated randomly over a uniform distribution according to the following constraints:

$$\begin{aligned} A &\sim U(0,1) \\ \Omega &\sim U(\Omega_{min}+300, \Omega_{max}-300) \\ \Gamma &\sim U(\Gamma_{min}, \Gamma_{max}) \\ n &\sim U(n_{min}, n_{max}) \in \mathbb{Z} \end{aligned} \tag{8}$$

where U represents a uniform distribution from a minimum value (the first parameter) to a maximum value (the second parameter); Ω_{min} and Ω_{max} were 0.1 cm^{-1} and 2000 cm^{-1}, respectively, and the other parameters vary for the nine datasets, as defined in the table: The non-resonant part of

Table 6: Spectral parameters of each dataset

Simulation dataset	Γ_{min} (cm^{-1})	Γ_{max} (cm^{-1})	n_{min}	n_{max}
i	2	10	1	15
ii	2	10	15	30
iii	2	10	30	50
iv	2	25	1	15
v	2	25	15	30
vi	2	25	30	50
vii	2	75	1	15
viii	2	75	15	30
ix	2	75	30	50

the susceptibility $\chi_{NR}^{(3)}$ was simulated as a product of two sigmoid functions as presented in Ref. [13]:

$$\chi_{NR}^{(3)} = S_1(\nu) \cdot S_2(\nu) \tag{9}$$

where S is given as

$$S(\nu) = \frac{1}{1+\exp[-b(\nu+c)]} \tag{10}$$

The parameters c and b were calculated randomly from

$$c \sim \mathcal{N}(a_1 \nu_{max}, a_2 \nu_{max}) \tag{11}$$

$$b \sim \mathcal{N}(10\nu_{max}^{-1}, 5\nu_{max}^{-1}) \tag{12}$$

where $\mathcal{N}$ denotes a normal distribution, where the first parameter is the mean value, and the second parameter is the standard deviation. The constants a_1 and a_2 are given in the following table: After generating $\chi_{NR}^{(3)}$, the

Table 7: Sigmoid constants

Sigmoid	a_1	a_2
S_1	0.2	0.3
S_2	0.7	0.3

total third-order susceptibility is then calculated as the sum of the resonant and non-resonant parts,

$$\chi^{(3)} = \alpha \frac{\chi_R^{(3)}}{\max\left(\left|\chi_R^{(3)}\right|\right)} + \chi_{NR}^{(3)} \tag{13}$$

where the resonant susceptibility term has been normalised by dividing by the maximum of its magnitude, and is then scaled by a parameter α, which is defined by a uniform distribution as follows:

$$\alpha \sim U(k_{min}, k_{max}) \tag{14}$$

For all of the datasets used in the main body of the paper, $k_{min} = 0.3$, and $k_{max} = 1$. The bCARS intensity spectrum was then calculated as the magnitude square of the total third-order susceptibility, with additive Gaussian noise ε,

$$I_{\text{BCARS}} = \frac{|\chi^{(3)}|^2}{2} + \varepsilon \tag{15}$$

where the factor of 2 in the denominator ensures that the noise-free intensity is will always have values < 1, since both terms in Equation 13 have values < 1. The noise term ε is generated in the same way as for Ref [13]. We define an array the same size as I, where each element is sampled randomly according to a standard normal distribution (zero mean and standard deviation of 1), which is then scaled by a parameter β, which is defined by a normal distribution with mean q_1 and standard deviation q_2 as follows:

$$\varepsilon(n) = \beta \mathcal{N}(0,1)\beta = \mathcal{N}(q_1, q_2) \tag{16}$$

For all of the datasets used in the main body of the paper, $q_1 = 0.0005$, and $q_2 = 0.003$. The Raman spectrum is given by:

$$I_{\text{Raman}} = \text{Im}(\chi_R^{(3)}) \tag{17}$$

It should be noted that for the nine datasets used to generate results for the main body of this paper, the noise parameters q_1 and q_2, and the parameter α, which controls the relative amplitude between the resonant and non-resonant components were given the values defined above. However, for the SNR investigation presented in Section D, dataset (ix) was simulated with two further cases of these noise related

Table 8: Noise parameters used to generate datasets

	k_{min}	k_{max}	q_1	q_2
Case 1	0.3	1	0.0005	0.003
Case 2	0.3	1	0.0005	0.03
Case 3	0.01	1	0.0005	0.03

parameters. Case 1 refers to the set of parameters used to generate all result in the main body of the paper. Case 2 and Case 3 related to two additional sets of parameters that were applied to dataset (ix) only in order to investigate the effect of noise in training sets on the capacity of the network to deal with especially noisy spectra. For more details on these specific cases and how they, see Section D.

C Training SpecNet

The Specnet network [13] was adapted in order to process spectra of length 1000 samples; the original network was designed for an input length of 640 samples. Other than this change in input length all other parameters in the original Specnet architecture were left unchanged. The full details of the structure and hyper-parameters are given in Table 9. As in Ref. [13], the loss function is the mean squared error (MSE) between the true target vector and the prediction. An Adam optimizer was employed with a batch size of 256 examples. In total 50 epochs were used. For all nine datasets, a dataset size of 220000 was used with a validation split of 0.1. This ensures that the size of the training and validation sets match those used for the nine datasets used with the VECTOR architecture. The code used to train and test SpecNet using these nine datasets is provided in Ref. [**?**]

D SNR analysis

In this section, we provide information on an additional study of the VECTOR architecture relating to its performance on noisy bCARS spectra. Noise in CARS can manifest from two main sources; the first relates to the relative amplitude of the resonant and non-resonant susceptibilities and the second relates to an additive noise that models read noise, dark current, and shot noise [43]. The relative amplitude of the two susceptibilities is ambiguous in the context of noise. On one hand, the NRB can be viewed as an amplifier of the resonant term. However, when the NRB is significantly larger than the resonant term it can become difficult to detect the latter, particularly in the presence of the second noise source. The additive noise in the simulations is modeled by a normal distribution, see Equation 16; an additive Gaussian noise can be used to account for the various sources of experimental noise, including read noise, dark current, and shot noise [43]. The latter two sources of noise are ideally modeled by a Poisson distribution but for practical mean values, both are approximated by a single additive Gaussian distribution with a high level of accuracy.

In order to investigate the capacity of VECTOR to recover the Raman spectrum from the bCARS spectrum in the presence of these noise sources, and also to investigate if this capacity could be improved by better training VECTOR with more noisy signals, the following approach was taken: three different VECTOR 16 (Dataset ix) were trained and then tested on the same spectrum under a variety of different noise conditions. These three networks are defined in Table B. In short, the first case is the VECTOR 16 Dataset ix network presented in the main body of the paper, for which the NRB and the resonant terms have approximately equivalent amplitudes, and there is a low level of additive noise. The second case is the same except that the amplitude of the noise term is increased tenfold. The third case is the same as the second except that the ratio of the two terms can vary by up to 100.

These three networks were then tested on the same dataset of 15 spectra, which correspond to one Raman spectrum (of type Dataset ix) under 15 different noise conditions: three additive noise amplitudes with standard deviation of 0.001, 0.01, and 0.1, and five different relative strengths of the NRB : 100, 10, 1, 0.1, and 0.01. For the first network the results are shown in Fig. 1. Also shown in the figure is the signal-to-noise ratio (SNR), defined as the maximum value in the Raman spectrum divided by the mean of the absolute value of the predicted spectrum subtracted from the true spectrum.

The performance of the network decreases as the both the NRB and additive noise terms increase. Interestingly, the performance of the network for the highest level of additive noise improves as the NRB increases for the first three levels, which relates to the amplification of the spectrum as discussed earlier. The same set of results are shown for the second network in Fig. 2 in which it can be seen that the network performs significantly better than the first network for high levels of noise. The third network has similar results, as shown in Fig. 3.

Table 9: Details of Specnet Architecture

Layer type	Operation (K, C_{in}, C_{out})	Activation	Output size ($T \times C$)	Parameter
Input	-	-	1000×1	-
Batch Normalization	-	-	1000×1	4
Activation	-	Relu	1000×1	0
Conv-1D	32, 1, 128	Relu	969×128	4224
Conv-1D	16, 128, 64	Relu	954×64	131136
Conv-1D	8, 64, 16	Relu	947×16	8208
Conv-1D	8, 16, 16	Relu	940×16	2064
Conv-1D	8, 16, 16	Relu	933×16	2064
Dense	- , 16 , 32	Relu	933×32	544
Dense	- , 32 , 16	Relu	933×16	528
Flatten	-	-	14928×1	0
Dropout(0.25)	-	-	14928×1	0
Dense	- , 1 , 1000	Relu	1×1000	14929000
Total Number of Paramters			15,077,772	

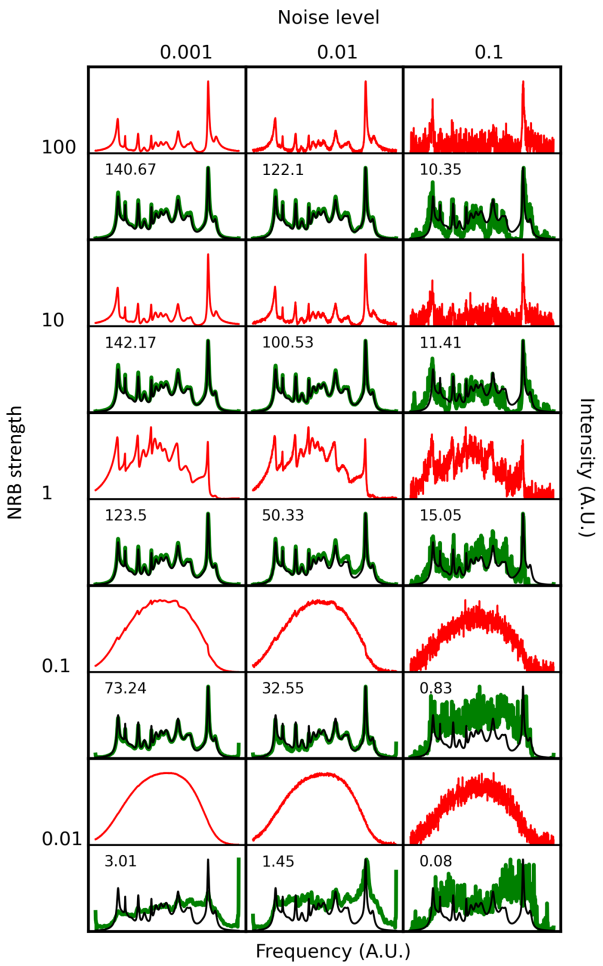

Figure 1: Results for VECTOR16 - Dataset ix - Case 1, as defined in Table B. This is the same network used in the main body of the paper for which the NRB and the resonant terms in the training set have approximately equivalent amplitudes, and there is a low level of additive noise. Fifteen instances of the same Raman spectrum are tested under different noise conditions: the additive noise term increases from right to left and the relative strength of the NRB increases from top to bottom. The input bCARS spectrum is shown in red, the predicted spectrum is shown in green, and the true spectrum is in black. The SNR for the predicted spectra are also shown in the figures.

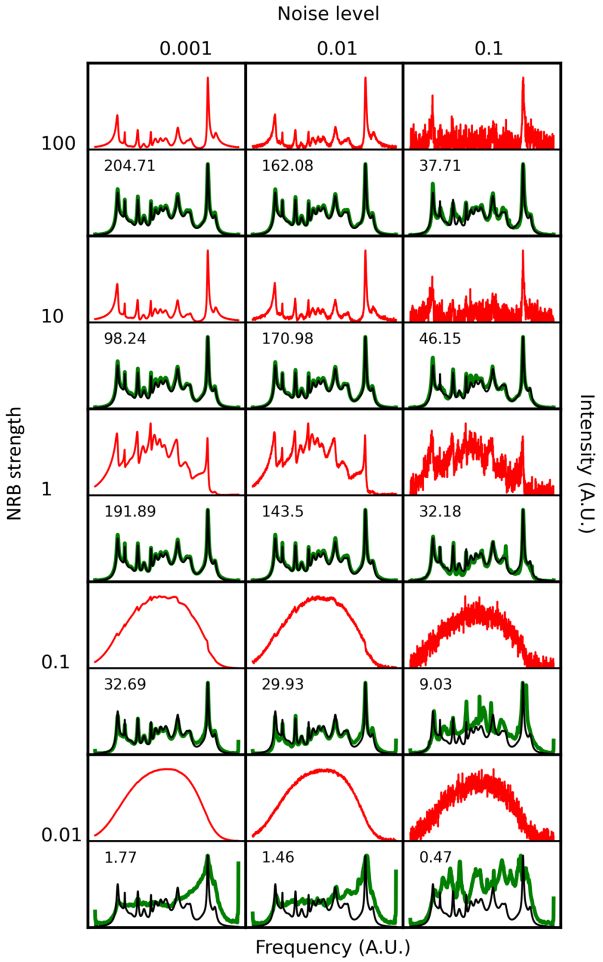

Figure 2: Results for VECTOR16 - Dataset ix - Case 2, as defined in Table B. The second case is the same as the first case except that the amplitude of the noise term is increased tenfold in the training set. Fifteen instances of the same Raman spectrum are tested under different noise conditions: the additive noise term increases from right to left and the relative strength of the NRB increases from top to bottom.The input bCARS spectrum is shown in red, the predicted spectrum is shown in green, and the true spectrum is in black. The SNR for the predicted spectra are also shown in the figures. This network performs better than the first network for high levels of noise.

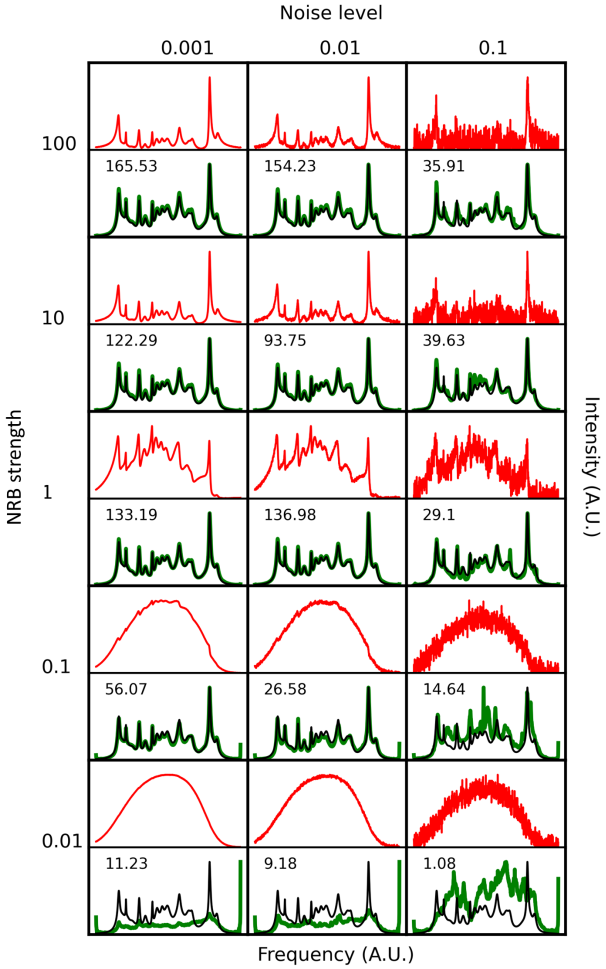

Figure 3: Results for VECTOR16 - Dataset ix - Case 3, as defined in Table B. The third case is the same as the second case except that ratio of the amplitudes of the susceptibilities can vary by up to 100. Fifteen instances of the same Raman spectrum are tested under different noise conditions: the additive noise term increases from right to left and the relative strength of the NRB increases from top to bottom.The input bCARS spectrum is shown in red, the predicted spectrum is shown in green, and the true spectrum is in black. The SNR for the predicted spectra are also shown in the figures.