ADVANCED OPTICAL MATERIALS
www.advopticalmat.de

RADAR: Raman Spectral Analysis Using Deep Learning for Artifact Removal

*Joel Sjöberg, Nicoleta Siminea, Andrei Păun, Adrian Lita, Mioara Larion, and Ion Petre**

Raman spectroscopy is a non-destructive analytical technique that reveals molecular vibrations, enabling precise identification of chemical compounds and material properties. Its spatial resolution and compatibility with microscopic imaging allow for high-resolution chemical mapping of heterogeneous samples. However, spectral artifacts such as baseline drift, cosmic rays, and instrumental noise complicate data interpretation, necessitating correction. RADAR is introduced, two lightweight deep learning models for artifact removal, capable of simultaneous denoising and correction of Raman spectra, significantly accelerating high-quality data acquisition. The models help reduce the data acquisition time by 90% while preserving signal integrity, as demonstrated on noisy spectra from a diversity of samples, biological and non-biological. These models are versatile and can be readily applied to novel Raman datasets, offering an order-of-magnitude improvement in acquisition efficiency. This work advances Raman spectroscopy as a faster, more reliable tool for chemical analysis, with broad applications in materials science, biomedical research, and beyond.

1. Introduction

Spontaneous Raman spectroscopy is one of the emerging techniques that produces molecular fingerprints of materials that are scanned without the need for sample preparation.[1,2] Due to inelastic scattering being a rare event, low signal-to-noise ratio prevents this technique from widespread applications. To obtain a reasonable signal-to-noise ratio, the acquisition time of a sample must be increased to hours, which impedes the use of this approach in real time. One way to circumvent this is through the use of surface-enhanced Raman spectroscopy, where a gold or other metal nanoparticle acts as a substrate that is capable of localized surface plasmon resonance.[3] Although this method provides an increase in signal intensity by orders of magnitude, it is not reproducible and has other drawbacks such as the matrix effect and cytotoxicity when used with live cells.[3] Another popular method to increase the sensitivity of spontaneous Raman spectra is to use stimulated Raman spectroscopy.[4,5] This technique is more expensive to set up and relies on the use of two incident lasers: a pump laser at one frequency and a Stokes laser at a different frequency. If the difference between these two lasers matches the vibrational frequency of interest, the intensities of the pump and the Stokes laser will be lowered (stimulated Raman loss) and increased (stimulated Raman gain), respectively. This increase in the Raman spectrum leads to very fast imaging of tissue or cells. However, this method produces an image of one frequency at a time, leading to collecting images of different time points, and therefore limiting its applicability for dynamic processes that change over time.

Another way to increase the signal-to-noise ratio of spontaneous Raman spectra is through denoising and artifact removal techniques. High-quality artifact removal techniques are vital for Raman spectroscopy, improving spectral quality for unknown compound identification and ultra-fast imaging.[6] Furthermore, improving the speed of signal processing is highly desirable for spectroscopic techniques because it has the potential to speed up analysis and produce fast responses that are suitable for clinical settings. Such progress is revealed in this study, resulting in the minimization of acquisition time. The artifacts targeted in this work are the baseline shifts, sporadic cosmic rays, and background noise. Their removal reveals the underlying signal peaks, which are highly informative of the chemical fingerprint of the analyzed sample.

J. Sjöberg, I. Petre
Department of Mathematics and Statistics
University Of Turku
Turku 20014, Finland
E-mail: ion.petre@utu.fi

N. Siminea, A. Păun
Faculty of Mathematics and Computer Science
University of Bucharest
Bucharest 010014, Romania

N. Siminea, I. Petre
Departament of Bioinformatics
National Institute for Research and Development in Biological Sciences
Bucharest 060031, Romania

A. Păun
Research Institute for Artificial Intelligence (ICIA)
Romanian Academy
Bucharest 060031, Romania

A. Lita, M. Larion
Neuro-Oncology Branch, National Cancer Institute
National Institutes of Health
Bethesda 20892, USA

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/adom.202500736

[Correction added on July 8, 2025, after first online publication: the third author's affiliation has been updated in this version.]

DOI: 10.1002/adom.202500736

The signal components of Raman spectra are the peaks (P), the baseline (B), the cosmic rays interference (CR) and the noise (N). Areas where the intensities steadily increase and then decrease following a local maximum in intensity are called Raman peaks (P). These information-rich peak areas offer a quantitative insight into the molecular content of the sample.[7] It is vital to extract P with high precision, such that all peaks inside a sample can be reliably compared within the spectrum and across multiple spectra. However, other components of the spectra may obscure P, complicating this analysis. Baselines (B) can shift the relative intensity between Raman peaks as a result of environmental light, sample emission, or even the instrument itself.[8]

Cosmic rays (CR) are sporadic particles emitted from the sun and outside our solar system which cause abrupt high-intensity frequencies or "spikes" in the spectrum.[8] The noise (N) typically originates from sporadic factors related to instrument calibration and the environment.

Preprocessing of Raman spectra is a well-studied topic and several methods exist to implement it. Baseline correction is often approached as an optimization problem where B is an approximately polynomial line, as is done, for example, in the ModPoly and IModPoly algorithms.[9,10] An alternative is the *bubble-fill* algorithm[11] that computes the baseline by expanding ovals (bubbles) until they touch the spectral line and leaves a baseline following the base of the spectrum. Zhang et al.[12] introduced the adaptive iteratively re-weighted penalized least squares (airPLS) algorithm for baseline correction, using an adaptive weight measure to avoid P signals while computing B. Concerning CR removal, the approach in Ref. [13] is to replace CR signals with the mean of the surrounding intensities by applying a modified z-score to the spectrum derivative. Another alternative is to require a set of spectra be used for comparison to remove the CR signals.[14] The Savitzky-Golay filter[15] is a popular choice for denoising and can be applied to identify and remove both N and CR through a sliding window approach.

The latest approaches in this field apply deep learning models to remove the need for user-specified parameters which traditional methods require. Deep convolutional neural networks were used in Ref. [16, 17] to remove all components and leave only the P signals. Furthermore,[16,18] included the ability to remove the baseline of the spectra and therefore designed their models to produce two outputs. A common trend in most of these models is the use of densely connected layers as the last layers where the output is produced. This leads to improved model predictions, possibly because the entire input is processed by all layer neurons. However, the inclusion of dense layers also results in a drastic increase in the number of model parameters. Furthermore, these layers require a fixed input and output size, limiting the use of the model to a certain spectral length defined by the developer. Auto-encoders (AEs) have been used in combination with convolutional layers to achieve localized decision making while reducing the size of the models in terms of their parameters. AEs aim to encode the input signals into a lower-dimensional latent space which can then be used to decode the information into the original input space, preserving as much relevant signal as possible.[14,19] Deep learning requires tremendous amounts of data to properly learn distributions and avoid overfitting. The solution is to use custom synthetic data generators able to supply models with massive amounts of synthetic data,[14,18,20–24] with real data used to validate the models. Generative adversarial networks (GANs) have also been used to synthesize data and remedy data limitations present in the medical field.[23,25] To evaluate the performance of denoising models, the signal-to-noise ratio (SNR) is often used to express the prevalence of signal in spectra. In particular, this is useful for comparing the ratio before and after denoising to quantify the improvement in signal quality relative to the reduced noise intensity.[18,19,26,27]

We developed RADAR, two new lightweight deep learning models capable of simultaneous denoising and correcting Raman spectra, with the goal of increasing the speed of high-quality data acquisition. We used the convolutional neural network architecture to obtain the smallest models currently known for Raman spectra denoising and correction. We designed the models to be able to process variable-length spectra and to disentangle the spectra simultaneously into their four distinct components B, CR, N, P. To train the models, we implemented a synthetic Raman spectra generator that combines the four independently generated components B, CR, N, P into a full Raman-like spectrum (**Figure 1**). The synthetic generator allowed us to have the ground truth values for the four components of Raman spectra, which made it possible to train the RADAR models as supervised machine learning projects, and to compare the performance of our models against several benchmark models in various setups.

We also tested the performance of our models and of the benchmark models on real datasets, biological and non-biological. The data we acquired specifically for this project included glioma samples (both noisy signals from formalin-fixed, paraffin-embedded glioma tumor samples, as well as cleaner signals from live glioma cell samples), samples of single-walled carbon nanotubes and of zinc sulfates. Additionally, we analyzed two Raman datasets acquired by others, using different spectrometers and acquisition protocols, on chondrosarcoma cancer samples, and on albite and augite minerals.

While the ground truth for the four components B, CR, N, P is obviously available for the synthetic datasets, it is not available for the real world datasets, making the evaluation of the models on such data more difficult. Without the ground truth, conventional error metrics such as maximum error and root mean square error could not be applied. Instead, we assessed how different methods estimated the noise level and the information-rich signal within the spectra. To facilitate this analysis, the datasets collected in-house were acquired at varying exposure times. Since longer exposure times are expected to yield higher signal-to-noise (SNR) ratios, this provided an indirect means of validating model predictions. This strategy also enabled us to rank the different models by their estimated SNRs, further demonstrating the effectiveness of our approach. We demonstrated that our models can reduce the exposure time by 90% while improving the SNR.

2. Results

2.1. Synthetic Generator of Raman Spectra

We implemented a mathematical model to synthesize Raman spectra of arbitrary length through the combined contribution

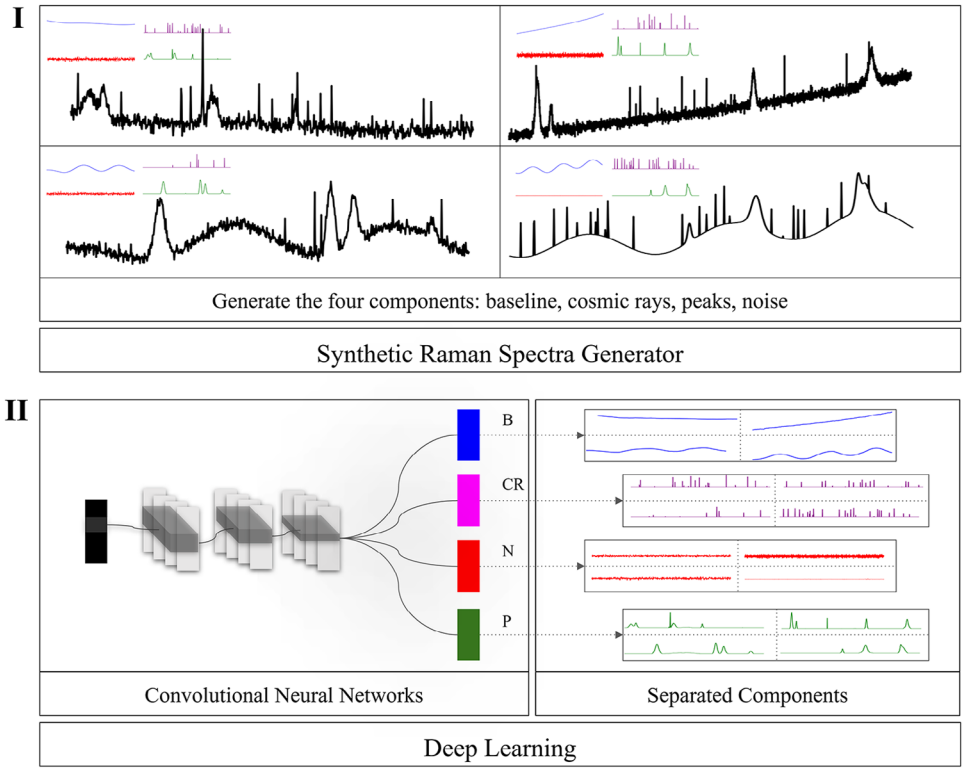

Figure 1. Our algorithmic design. a) We designed a generator to synthesize Raman spectra (in black, four different examples) through synthetic baselines (B, blue in the top-left of each of the four examples of synthetic spectra), cosmic rays (CR, in purple), noise (N, in blue) and peaks (P, in green) components. b) We used the generator to train two deep learning models to extract the B/CR/N/P components of real Raman spectra. We evaluated our models on synthetic data, on biological samples (paraffin-embedded glioma tissue samples, live glioma cells, chondrosarcoma samples) and non-biological samples (albite and augite minerals, single-walled carbon nanotubes, and zinc sulfate).

of the four components B, CR, N, and P. The ability of having four explicit components of a spectrum and of quickly generating large datasets of millions of Raman spectra of arbitrary length was crucial for training the deep learning models for Raman correction and denoising: having the ground truth of the synthetic data allowed us to measure the errors of the machine learning models and to optimize their training. The precise ground truth values of the four components cannot be obtained for experimentally acquired spectra, given their sporadic nature.

The generator creates four synthetic spectral components B, CR, N, and P and combines them into a synthetic spectrum. The generated spectrum is obtained by adding the B, CR, and P components. We applied max-normalization to each spectrum by dividing all its values to the maximum value, a necessary and standard step in preparing data for machine learning. The noise component N is then added to be able to experiment with different signal-to-noise ratios. The length of the synthetic spectra is variable and can be changed by the user from the default value of 1000 in our freely available code (https://seafile.utu.fi/d/80db1f613ef74da0a050/ Seafile and https://github.com/JoelSjoberg/Comprehensive-Raman-Spectral-Analysis-Using-Deep-Learning/tree/main Github). Diverse examples of synthetic spectra are in Figure 1a and in the Figure S1 (Supporting Information).

2.2. Deep Learning Models for Spectral Correction and Denoising

We designed and trained two deep learning models to reliably and accurately decompose Raman spectra into their four components. We aimed to have the models applicable without retraining on spectra of variable length, and to have as few learnable parameters as possible. We tested their performance on several Raman datasets collected from biological and non-biological materials.

We trained our models on synthetic spectra, using our generator to rapidly produce synthetic training data on-the-fly. To prevent overfitting, we trained the models using a learning rate decay for 12 epochs. In each epoch, batches of 100 synthetic spectra are generated 200 000 times, resulting in a total of 20 million spectra generated for each epoch. In total, both models are trained on 240 million synthetic spectra. This number of data points is roughly 34 times larger than the number of parameters in our larger model (585,299).

One of our models used a *standard* convolutional neural network (CNN) architecture with a four-component output, one for each of the spectral components of the input. The output returning N is calculated as a simple subtraction between the original input spectrum and the predicted B, CR, and P-vectors. This means that the N-predictions of the standard model do not have dedicated learnable parameters. The architecture

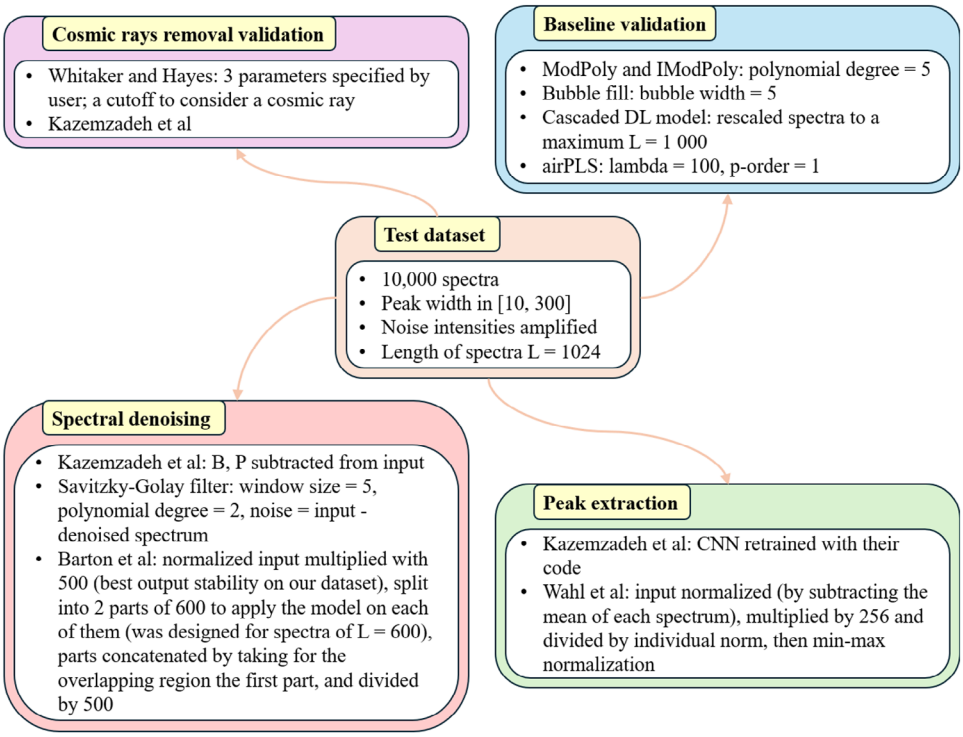

Figure 2. The evaluation protocol of our models against several benchmark literature models. A test dataset with 10,000 synthetic spectra was generated, with their explicit baselines, peaks, cosmic rays, and noise components saved. The extraction of each of the components was evaluated against the ground truth and against the predictions of the relevant benchmark models.

of the standard model is shown in the **Figure 2** (Supporting Information).

Our other model used an *ensemble* architecture, with a series of convolutional neural networks feeding into each other, each focused on identifying specific components of the input spectrum. This model has 280,592 parameters. The architecture of the ensemble model is in the **Figure 3** (Supporting Information).

2.3. Model Evaluation on Synthetic Datasets

We evaluated both our models on their ability to extract the components B, CR, N, and P. We compared our models with several literature benchmark models, with the evaluation design indicated in Figure 2. For this, we generated a synthetic test set consisting of 10 000 spectra, with peak widths randomly selected from the interval [10 300]. To add further variance to the synthetic spectra, we instructed the generator to amplify the N intensities of the test set. The synthetic spectra have a length of 1024, which is the spectral length that the recent models of Refs. [16] and [27] used, rendering it directly applicable for their models as well. To evaluate the performance of the model, we used the *maximum error* metric, which captures worst-case error over all frequencies in the spectrum, i.e., the largest absolute difference between a predicted value and its corresponding target value:

$$\text{MaxError} = \max_{1 \le f \le N} \left| y(f) - y'(f) \right| \tag{1}$$

where $y(f)$ denotes the true intensity of the spectrum at frequency f, and $y'(f)$ is the predicted intensity at the same frequency. We also tracked the *root mean square error* (RMSE), a score function indicating an aggregated mean deviation from the ground truth over all frequencies in the spectrum:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{f=1}^{N} \left(y(f) - y'(f) \right)^2} \tag{2}$$

2.3.1. Baseline Correction

We evaluated the extraction capabilities of our models for B and compared them with standard methods designed to detect B, such as airPLS,[28] ModPoly and IModPoly,[9,10] and the bubble fill algorithm.[11] Although these methods are generally applicable to a diversity of spectra, we found that the parameters of these methods require fine-tuning, depending on their input. We tested many setups of different parameters and chose the parameters which gave an overall minimal error for all generated spectra. We selected the standard parameters lambda of 100 and p-order of 1 for airPLS, a polynomial degree of 5 for the ModPoly and IModPoly methods, and a minimum bubble width of 75 for the bubble-fill algorithm.

We also compared our models with the cascaded deep learning model of Kazemzadeh et al.[16] Their model was trained using transfer learning with the U-net model to extract B and P.

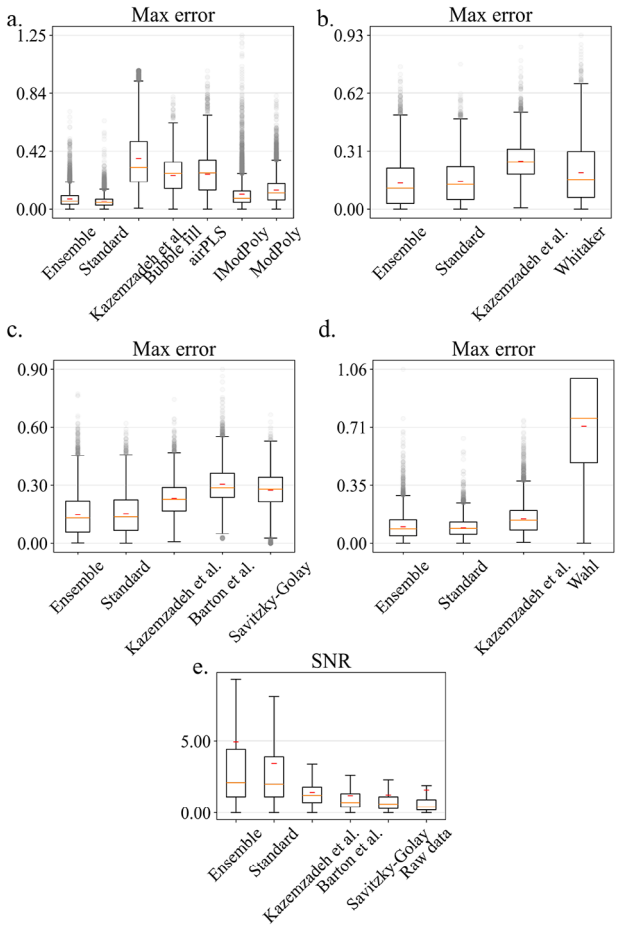

Figure 3. The performance of the models extracting each of the four components of synthetic Raman spectra. The small blue line inside the box displays the mean value of the metric and the orange line displays the median. a) Baseline extraction, b) Cosmic ray extraction, c) Noise extraction, d. Peak extraction, e) Signal-to-noise ration improvement.

The model includes a smaller network at the end to perform denoising. The Kazemzadeh et al. model contains 21,671,810 parameters, a much higher number than our models, and is trained to accept spectra with diverse intensities. The model maintains the scale and ratio between the B and P vectors, which makes it comparable to our models on normalized spectra, with the exception that input spectra need to be scaled up to keep the model predictions stable. We rescaled our generated spectra to a maximum intensity of 1000, to make them compatible with the type of input spectra expected by the Kazemzadeh et al. model.

The performance of the B-extraction in all these benchmark models is shown in Figure 3a and in the Table S1 (Supporting Information). The comparison shows that our models are capable of identifying the baselines with a mean maximum error score of less than 0.1 for both models, which outperforms the competing methods. The variance of the scores shows that our models have highly consistent behavior in this regard, maintaining a low variance for their error scores.

2.3.2. Cosmic Ray Removal

We compared our models with the method presented by Whitaker and Hayes[13] for the removal of CR. They accomplish this by computing the modified z-score of the first derivative of the input spectrum. A cutoff point is then used to determine the location of the CR signal around which the spike is replaced with the mean of the surrounding intensities. Furthermore, we extracted CR-predictions from the Kazemzadeh et al. model by subtracting the ground truth B vector from their predicted B. Their model was not designed to extract CR, but we found that this method yielded good results. The results are shown in Figure 3b and in the Table S2 (Supporting Information). The comparison shows that our models maintain low RMSE and maximum error scores, while the method by Whitaker and Hayes has comparative mean and median with more outlier values and more examples in the fourth quartile. The Kazemzadeh et al. model had a larger error score for both RMSE and maximum error.

We further examined the effect of CR-predictions in terms of the number of CR present in the input spectra. We evaluated model performance on a version of the synthetic data devoid of CR signals. We then iteratively increased the number of spikes present and evaluated again until a total of 100 CR signals were present in the input spectra. The results are available in the Table 3 (Supporting Information). We found that there were only marginal difference between the models. The Whitaker method had the most difficulties in removing them likely due to the reliance on standard deviation from surrounding wavelengths which can be affected when the number of cosmic rays increases. In extremely contaminated acquisitions we recommend, as a practical measure, combining RADAR with hardware-level spike filtering or initial pre-screening.

2.3.3. Peak Extraction

Following the advice of the authors of Ref. [17] (private communication), we retrained their CNN architecture, using their own code, and included it in our comparison. To make the comparison with the Wahl model[17] feasible, we preprocessed the normalized input spectra by subtracting the mean of each spectrum. We then multiplied the spectra by 256 and divided each spectrum by their individual norms. The Wahl model scales the P intensities while attempting to reduce the N intensities, resulting in outputs that do not preserve the original ratio between the spectra components. To enable comparison with our models, we resolved to Min-Max-normalize the outputs of the Wahl model and the target values they are compared to.

Our models outperformed the Kazemzadeh model and the Wahl model with a mean RMSE of 0.02 for the *standard* and the *ensemble* models (Supplementary Material, Table 4). Likewise, the max-error metrics showed that our models have lower max error

than both competing models. The metrics of the Wahl model are weaker in comparison to the other models, possibly because our synthetic data generator is somewhat different from the one for which the Wahl model was designed.

2.3.4. Spectral Denoising

To evaluate the denoising effect of our models, we computed the Signal-to-Noise Ratio (SNR) of the corrected spectra in addition to the RMSE and max error metrics. In cases where both signal and noise are represented by random variables, the ratio can be defined as the mean value of the signal divided by the standard deviation of the noise as utilized by [29]: where μ_P is the mean value of the vector P in all nonzero locations and σ_N is the standard deviation of the noise component of the spectrum. We ignore the regions without peaks in the P-vector to avoid affecting the mean value by zero-intensity regions.

We compared our models with several published methods. We used the Kazemzadeh et al. model to produce P-, and B-predictions on the input spectra and then subtract the predictions from the input to get the noise vector as the remainder. For the Savitzky-Golay filters, we selected a window size of five and a polynomial degree of two for the method and then subtracted the denoised spectrum from the input to extract the noise. We also applied the autoencoder model trained by Barton et al.[14] to denoise Raman spectra. This model was designed for spectra with 600 frequencies and, in our experiments, it provided stable output predictions for spectral intensities between 0 and approximately 1000. To compare this model with the other ones, we resolved to split each spectrum into two parts consisting of 600 frequencies. We then applied their model on both parts to cover all the 1024 frequencies in the test spectra. The predictions were then concatenated to get predictions for the entire spectral validation set, with the first part covering the first 600 frequencies including the overlap region. We multiplied the normalized input spectra by 500 before splitting them and feeding them to the model. We chose 500 since it provided the best output stability for the model when denoising our datasets. The outputs were then divided by 500 to return to the original normalized scope. The results are in Figure 3c and in the Table S5 (Supporting Information).

As shown in the box plots in Figure 3, our models outperform the benchmark models in the maximum error and SNR metrics. The Barton et al. model[14] performed quite well on these data; however, the ends of the corrected spectra were often skewed upward, resulting in a penalty from the metrics. Likewise, the noise it extracts also includes cosmic rays, further penalizing their metric results.

Furthermore, we evaluated the models on datasets with specific SNRs to better compare model performances on specific SNR values. The results of this experiment are shown in the Table S6 (Supporting Information). The results show that all models improve at a seemingly exponential rate as the SNR increases. With this we were able to see that the models performed well on spectra with SNR = 0.5. While improvements to the ratio could be seen in the case of SNR = 0.1 we note that the signal is so sporadic in this case that no useful signal can be extracted in practice. The models maintain competitive performance across all SNRs generated with RADAR significantly outperforming them on higher SNR values as seen in the inter-quartile range between the models' improvements to the SNR.

2.4. Model Evaluation on Experimentally Acquired Spectra

We further evaluated the models for their ability to extract the different components of Raman spectra acquired experimentally from diverse sources using various spectrometers and acquisition protocols. Specifically, we used data from chondrosarcoma samples[30] and a subset of Raman spectra of albite and augite minerals.[31,32] We compared the RADAR models against those by Kazemzadeh et al. and Wahl. The results are shown in the Figure S4 (Supporting Information). For compatibility with the benchmark models, all spectra were truncated or interpolated to a fixed length of 1024 frequencies. In all cases, we found that Kazemzadeh et al.'s model achieved performance comparable to RADAR in predicting P-components. The B-predictions were less straightforward, as Kazemzadeh et al.'s method removed both B and CR signals simultaneously. Wahl's model could not be directly compared to the others due to its peak-extraction-based design. Although the peak locations it identified were generally consistent, the model frequently introduced small peak-like artifacts, resulting in false positives in the output.

2.5. Model Evaluation on Data Acquired at Different Exposure Times: Reducing the Acquisition Time

The exposure time when the material is scanned determines the quality of the resulting Raman spectrum: the higher the exposure time, the better the signal. However, longer exposure times also require longer acquisition times for spectral data. Improved denoising of the signal can reduce the required exposure time for high-quality spectra, allowing for rapid sample scanning.[33] This can be a critical factor and a bottleneck in the application of Raman spectroscopy in clinical settings, especially in the operating room.[34] We set out to test our models on spectra with varying exposure times to evaluate their potential to reduce required acquisition times. We acquired four separate samples, biological (FFPE glioma tumor samples and live glioma cells) and non-biological (carbon nanotubes and zinc sulfate), each scanned at several different exposure times. To apply the Kazemzadeh et al. and Barton et al. models to these spectra containing 1738 frequencies, we split the spectra into two pieces containing 1024 frequencies for the Kazemzadeh et al. model and three pieces containing 600 frequencies for the Barton et al. model. We then followed the same procedure as in the spectral denoising step, resulting in N-predictions for the entire spectral dataset for both models. Savitzky-Golay filters were also used to compare to a standard method for spectral denoising on real data. For the Barton et al. model and for the Savitzky-Golay method, we removed the baseline predicted by the Kazemzadeh et al. model to show better contrast between our models and the competing methods. We manually located the signal and noise regions in the mean spectrum of each sample to compute the SNR for the raw data and for each denoising model. The design of the evaluation is summarized in **Figure 4**. The results show that the RADAR models and the Kazemzadeh et al. model are capable of reducing the

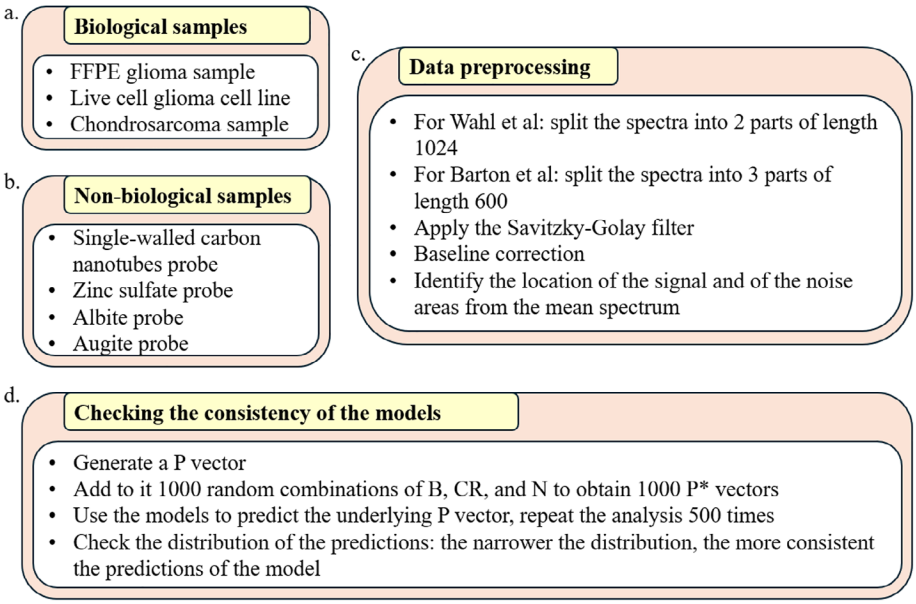

Figure 4. Evaluating RADAR's ability to reduce Raman spectra acquisition time. a) Raman spectra were acquired from two glioma tumour datasets, with each sample scanned at multiple exposure times ranging from 5 s to 0.0025 s. b) Two non-biological samples (single-walled carbon nanotubes and zinc sulfate) were also scanned at exposure times ranging from 5 s to 0.00167 s. c) Data were pre-processed according to the specific requirements of each method. Following baseline correction, the signal and noise regions were identified using the mean spectrum of each sample. d) We evaluated whether the models consistently extracted the peak vector even when the other three components exhibited variability.

exposure time in each case by approximately 90%. The metrics are shown in the **Figure** 5 (Supporting Information).

2.5.1. FFPE Glioma Sample

We applied the models to Raman spectra of an FFPE glioma tumor sample, scanned at different exposure times from 5s to 0.005s. To identify the tumor regions in the sample, we followed the approach of Ref. [35]: we performed baseline correction with our *standard* model and applied the DBSCAN algorithm to the Raman spectra acquired with 5 s of exposure (the highest-quality signal) to remove the non-tumor regions from the data. In Figure 5, we display the distribution of the raw data for each available exposure time of the glioma samples. We also show the P-predictions for each of the models applied to this dataset. The blue and green boxes overlaying the distributions represent areas of noise and signals respectively, displaying the regions used to compute the SNR for each exposure time. The results of the SNR calculations are in the Figure S6 (Supporting Information). Our models, along with the Kazemzadeh et al. model, were able to maintain a high SNR until the exposure time of 0.25 seconds. We concluded that all models were able to extract good quality P-predictions when evaluating spectra with exposure times as low as 0.5s, a 90% reduction from the total 5s of exposure time.

2.5.2. Live Glioma Cells Sample

We applied the same computational protocol to the Raman spectra extracted from live glioma cells, with exposure times ranging from 2 s to 0.0025 seconds. The resulting SNR boxplots are in the Figure S7 (Supporting Information). All models were able to reliably extract the signal while reducing the exposure time from 2 to 0.25s, an 87.5% decrease in exposure time.

2.5.3. Two Non-Biological Datasets

To further quantify the improvement in reducing the required exposure time we evaluated the models on two additional non-biological datasets: a single-walled carbon nanotube probe and a zinc sulfate probe, with each sample scanned at various exposure times ranging from 5 to 0.00167s. The single-walled carbon nanotubes were randomly dispersed into a glass matrix. This sample is a good source of carbon and we selected it because carbon has a big Raman cross section corresponding to a sharp peak at around 1600 cm^{-1}. The zinc sulfate probe contains intrinsically a lot of water, leading to many peaks at frequencies between 200 and 1600 cm^{-1}. The boxplots for denoising the carbon nanotubes and zinc sulfate are presented in Figures S8 and S9 (Supporting Information). We applied RADAR and all benchmark denoising methods to these samples and computed the improvement in the SNR across different exposure times. The results are in the Figures S8 and S9 (Supporting Information). Our analysis shows that RADAR achieves the highest SNRs. However, at lower exposure times (where the noise level is higher), all methods fail to improve the SNR compared to the raw data.

2.5.4. Saliency Maps

Saliency maps are used in machine learning, especially in computer vision, to identify and visualize which parts of an input are

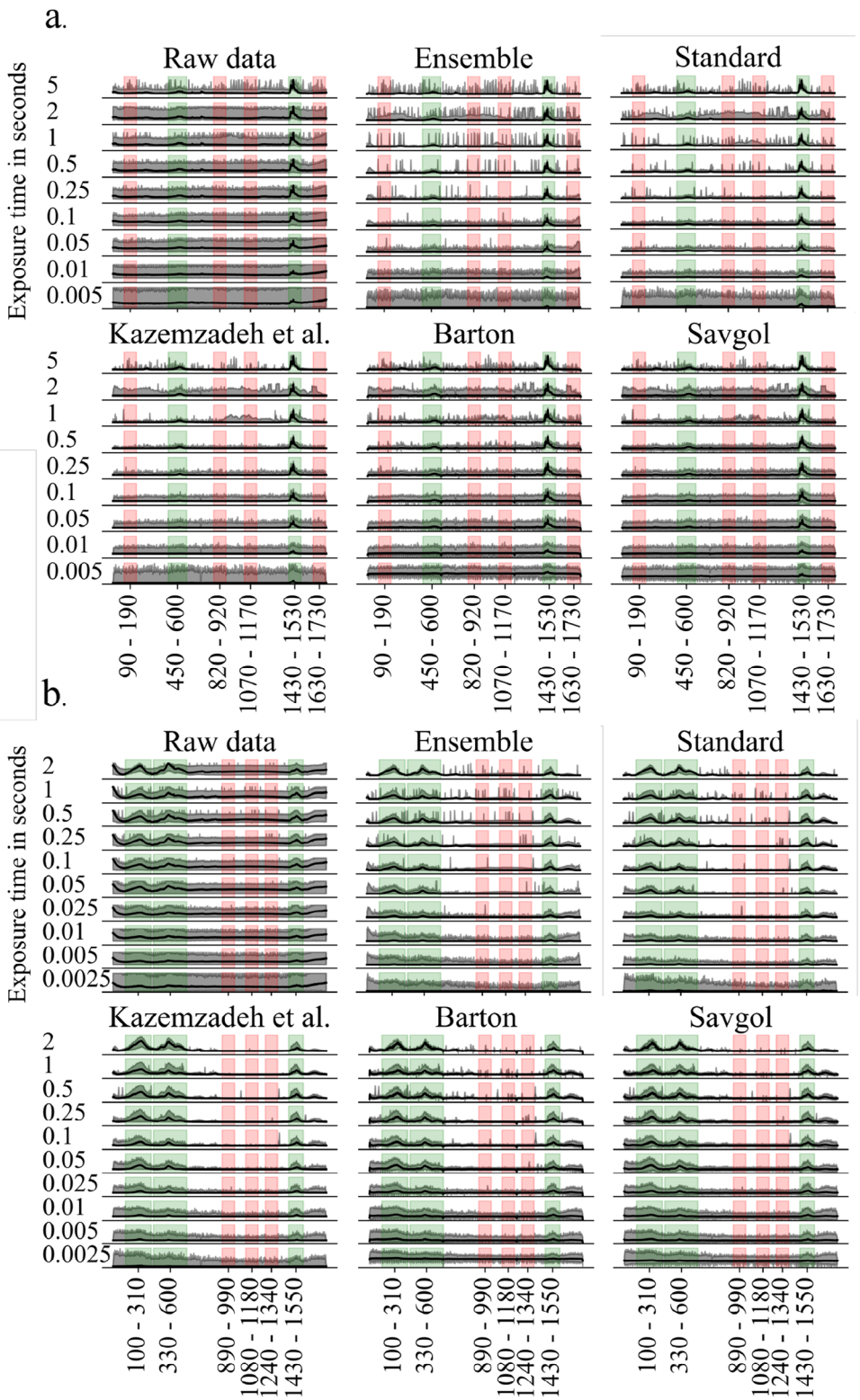

Figure 5. The distributions of spectra extracted with different exposure times. The transparent gray area in each plot is the distribution of the spectra, while the thicker black line is the mean of the distribution. The red and green boxes overlaying the distributions represent areas of noise and signals respectively, displaying the regions used to compute the SNR for each exposure time. a) The spectra and P-predictions of the FFPE glioma sample. b) The spectra and P-predictions of the live cells sample.

most influential for a model's prediction. They help interpret and explain deep neural networks, particularly convolutional neural networks (CNNs). They are simple to compute, being based on gradient calculations, but they are sometimes noisy and hard to interpret.[36] They are most often used for classification problems, but can in principle be used, as we do here, also for regression problems.[37] This method can strengthen the explainability in image classification models.[38,39] Recently, this method was utilized by Ref. [40] to compute saliency maps for Raman spectra classification models, promoting their use on 1-dimensional spectral datasets. We computed the saliency maps of the four raw materials available in this study for the ensemble and standard RADAR models to examine the explainability of their P-predictions. For each sample, we extracted 100 spectra and computed the saliency map with respect to the center of the manually identified peak areas. We then normalized each of the saliency scores and took the median of the resulting score vector. These median saliency vectors showed good alignment with peak positions and consistently low saliency in non-P regions, indicating that both models successfully focus on the relevant input features when predicting P. The saliency maps are shown in Figure 10 (Supporting Information).

3. Discussion

There is a fast growing literature on new machine learning models for baseline correction and spectral denoising. The models we propose in this study are a contribution to this effort and, to our knowledge, the smallest state-of-the-art models currently available. High-quality preprocessing of Raman spectra is essential to their broad applicability. This involves accurate extraction of the four components (B, CR, P, and N) and the ability to process spectra of variable length. Having access explicitly to the four components (rather than just to the peaks, the information-rich component) is important in calibrating and validating the experimental assay. Dealing with spectra of variable length between different assays implies either changing the preprocessing setup, a computationally expensive and time-consuming step, or having the computational tools agnostic to the spectral length. In this study, we proposed a solution that addresses these problems, in the form of RADAR, two deep learning models for artifact removal in Raman spectra.

Through our models, users can easily process spectra of any length, gaining the ability to extract each component from the spectrum to analyze them, allowing for a transparent method capable of showing exactly how each component contributes to the input spectrum.

Our models are trained on millions of synthetic spectra offered by our synthetic generator. The trained models are accompanied by our open-source code to make it possible for users to refine them for wildly different types of peak widths, originating from different application domains than the medical setup we were motivated by. By providing an open source generator alongside the RADAR models, we support continued efforts in this field. This includes addressing specific constraints of the spectrometer, of the acquisition conditions, and of the materials being scanned, that may require model fine-tuning for domain specific tasks not accounted for in our model training. To the best of our knowledge, ours are the smallest models currently known in the literature for simultaneous denoising and spectral correction, making them less prone to overfitting and easier to use while maintaining competitive performance to proven state-of-the-art models.

In virtually all synthetic Raman generators, Gaussian and Lorentzian curves are commonly used to generate both B and P. This raises a question about the ambiguity between P and B during the signal extraction phase. This means that one important aspect of these models is whether B and P are consistently extracted by the models. In cases of extreme overlap or low signal-to-noise ratio, some ambiguity will certainly remain, an inherent limitation in both model-based and traditional approaches. We tested however the results of our models under less extreme conditions. We designed an experiment intended to measure how consistent the model results in P were when applied to spectra skewed by different components B, CR and N. The experiment was designed as follows: generate one P vector, then generate 1000 unique combinations of random B, CR and N vectors to be added to a copy of P to offer us a test set of 1000 uniquely disturbed P vectors denoted P*. We then used the models tested in this work to extract P from each P*. The purpose of this experiment was to check how each model extracts P and whether or not the predictions remain consistent compared to P*. To measure the consistency of the model predictions, we used the standard deviation, the absolute value of the z scores and variance measured for each predicted frequency P. The mean values of all metrics were then calculated and used to denote how much the predictions vary from each other. The z score was suitable for this task as it denotes the distance from the average measured in units of the mean. A value of 1 or larger is considered an outlier, whereas a value less than one denotes normality and is therefore consistent with the average prediction P. Variance also describes the tendency of predictions to differ between the different predictions. We measured variance in two different areas for each model's predictions on the 1000 disturbed spectra: one where the generated P had non-zero intensities, and the other where P had zero intensities. Consistency can then be measured as the tendency of each model to vary in both signal and non-signal regions, combined with the z-score and standard deviation. We performed this experiment 500 times, each iteration generating novel P and its corresponding 1000 P*-vectors. The main steps of the experimental design are shown in Figure 4D). The performance of our models compared to the Kazemzadeh et al. and Wahl models in this setup for the first generated P-vector is shown in **Figure 6**. Also displayed in Figure 6 are the boxplots for z-score, standard deviation, and variance throughout all 500 trials. Based on this experiment and the boxplots in Figure 6 we can see that the z-scores varied drastically for all models throughout all 500 trials. The Kazemzadeh et al. model managed to achieve low mean z-scores for most trials but has a comparable second and third quadrant to our *ensemble* model, which has the lowest maximum z-score. The variance (measured at locations where the generated P-vectors contained signals) also showed the struggle that the Kazemzadeh et al. and Wahl models have in producing consistent P-predictions. The standard deviation (taken over all frequencies of the predicted spectra) mimics this trend, showing a slight advantage for our *standard* architecture. Overall, all models performed well with the exception of the Wahl model, which struggled to provide consistent outputs for the P* vectors (Table 7, Supporting Information).

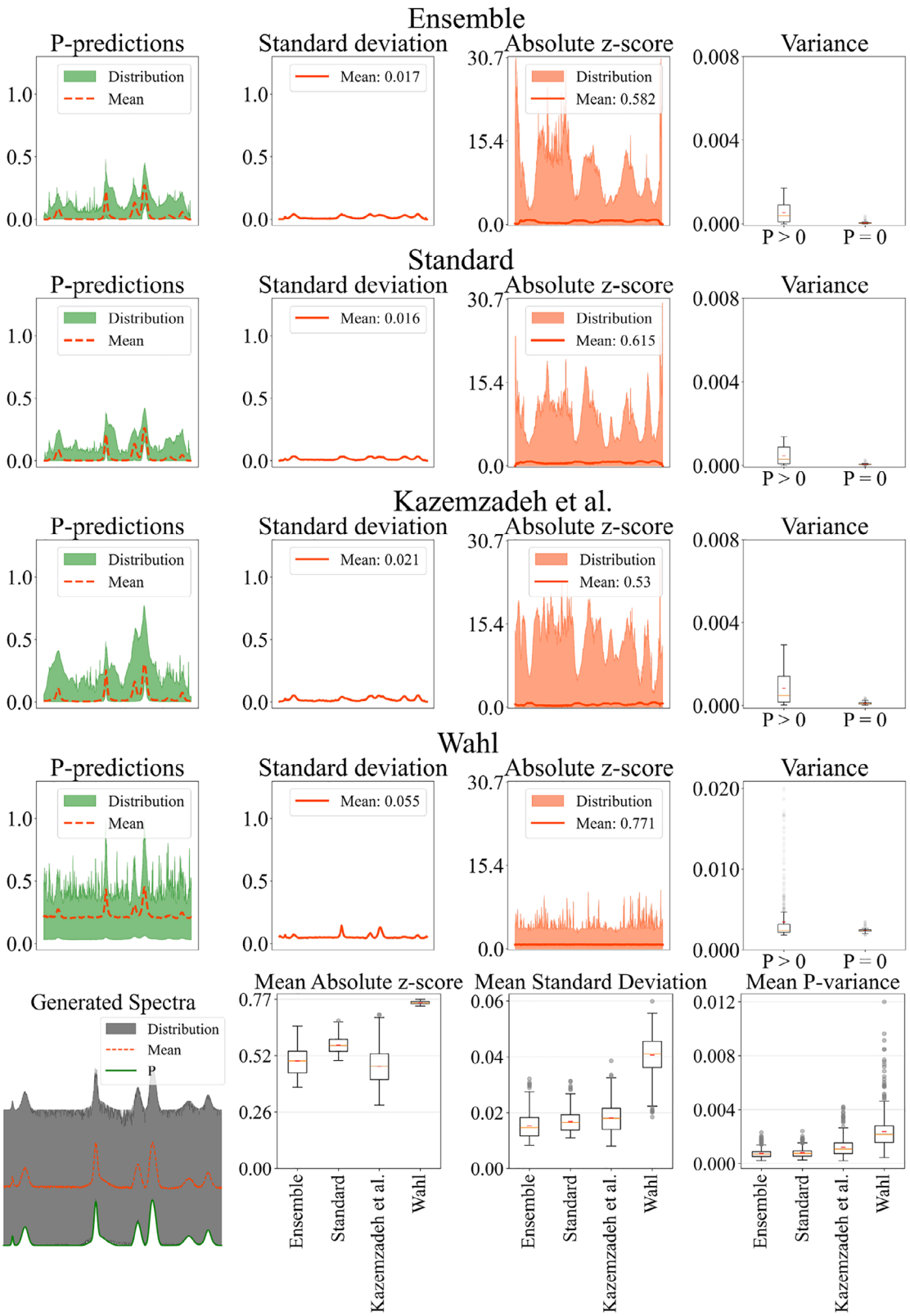

Figure 6. The consistency of the *ensemble*, *standard*, Kazemzadeh et al. and Wahl models, measured through the standard deviation, absolute z-score and P-variance for 1000 randomly generated P*-vectors. The generated spectra are shown in the lower left corner with the gray area displaying the distribution of all generated 1000 spectra for one trial. The last row contains the mean metrics taken across the 500 trials.

RADAR managed to outperform established methods and other machine learning methods for baseline correction. Both model maintained low mean max error (standard: 0.06, ensemble: 0.08) compared to Kazemzadeh et al. (0.37) on our synthetic data. The metrics for cosmic ray removal were similarly improved (standard: 0.15, ensemble: 0.14) over the competing Whitaker method (0.19). For denoising, we identified an improvement in max error (standard: 0.15, ensemble: 0.15) and SNR in RADAR (standard: 3.45, ensemble: 4.94) over the max error of Savitzky-Golay filters (max error: 0.28, SNR: 1.22) and the Barton et al. denoising CNN model (max error: 0.31, SNR: 1.18). For synthetic spectra with a SNR of 0.5, our models significantly increased SNR to 3.11 and 2.73 for the standard and ensemble models respectively. For identifying peak intensities, our models achieved lower mean max errors (standard: 0.09, ensemble: 0.10) than Kazemzadeh et al. (0.15) and Wahl (0.71). The variance when measuring consistency between predictions on a P vector manipulated by different components was low for RADAR (standard: 0.0008, ensemble: 0.0008) relative to Kazemzadeh et al. (0.0012) and Wahl (0.0024). On Raman spectra extracted from real materials, we identified peaks and noise regions manually to estimate the true N and P vectors which we then used to compute max error metrics and SNRs. When predicting P both RADAR models and Kazemzadeh et al. maintained competitive performance. For SNR on the raw spectra, our models managed to improve the SNR better than all competing methdos in three out of the four materials analyzed. The carbon nanotubes sample was problematic for all machine learning methods to improve in all cases except for the maximum exposure time of 1 second, where the ensemble model produced the greatest increase in SNR. In cases where the exposure time was decreased, none of the available methods managed to improve the SNR beyond the SNR of the raw data.

One of the key strengths of RADAR is its ability to output interpretable components, including the baseline, noise, cosmic ray artifacts, separately from the purified signal. In practice, these components can indeed be visualized alongside the raw input spectrum, enabling users to directly assess what the model has subtracted or preserved. This visual transparency is especially valuable in clinical and industrial settings, where interpretability and trust in automated processing are essential. An example of such a visualization is in the Figure S4 (Supporting Information).

4. Experimental Section

Synthetic Data Generator: The data consisted of spectra made out of four different generated components. The generator synthesized baselines, cosmic rays, noise, and peaks independently, and then adds their frequencies together to generate the full spectrum. It was designed to work with spectra of arbitrary lengths. A diagram of the generator is in **Figure 7**.

Baselines: The baseline component B was modeled based on five basic types of baseline (linear, sinusoidal, polynomial, Gaussian, and Lorentzian), following.[41] Each baseline we generated was the sum of a variable number (one to five) of baselines, each of a different type. This variable combination of functions were chosen to synthesize a more diverse set of baselines. The resulting spectrum was then max-normalized to obtain a final baseline spectrum with a maximum intensity of one. The method of Ref. [41] focused on generating five different baselines based on the following mathematical functions:

1. Linear: $f(x) = ax + b$;
2. Sinusoidal: $f(x) = sin(ax + b)$;
3. Polynomial: $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0$;
4. Gaussian $f(x) = ae^{\frac{-(x-b)^2}{2c^2}}$;
5. Lorentzian $f(x) = \frac{1}{\pi} \frac{a}{(x-b)^2+a^2}$.

This approach was expanded by defining parameters to generate diverse and stable versions of the listed baseline functions 1–5 for variable lengths without dependence on preexisting data. The functions defined on the interval [0, 1] were considered. For the functions 2–5, the variety was further increased by normalizing their output to be within the [0, 1] interval. A scale variable and an offset variable were then defined to scale down the values of the baseline and impose an offset from a minimum value of zero. Furthermore, to expand the complexity of the functions obtained, a baseline was considered as the sum of five random functions 1–5. This aims to avoid eventual overfitting to baselines which strictly follow the patterns of 1–5 and enables the identification of more complex baselines.

Cosmic Rays and Noise: Cosmic rays were generated as a list with 0-entries, with a length defined by the parameter L set to 1000 by default. The parameter m was used to denote the maximum number of cosmic rays in the spectrum, with a default value of 30. Cosmic rays were inserted into the list at random indices. The generator first generated a random number of cosmic rays $\mathcal{U}(0, m)$. Then it generated a random index for each cosmic ray of $\mathcal{U}(0, L)$. Finally, the intensities of the spikes were generated randomly from $\mathcal{U}(0, 1)$. The random nature of this algorithm indicated that the number of cosmic rays, their positions in the spectrum, and their intensities were all diverse from output to output. This component was max-normalized so that its values were in the interval [0, 1].

The noise was generated as a line with L points, where each point was drawn from a normal distribution $\mathcal{N}(0, sd)$ where sd was a small random floating-point number $\mathcal{U}(0, 0.1)$. The points on the left and right extremities of the noise vector were set to 0. This eliminated the ambiguity of where the baseline correction should start and end on the spectrum.

Peaks: The peak component (P) consisted of a random number of peaks, each of random width between 16 and 256. Each peak was inserted into a random place in the spectrum. The final P component was obtained as the sum of the generated peaks, followed by a max normalization step. Note that several peaks, possibly generated using different mathematical functions, could get combined in some parts of the spectrum, offering a more diverse set of functions for this component.

The peaks were generated by randomly choosing Gaussian, Lorentzian, and pseudo-Voigt curves with randomly adjustable widths. These functions were often used to simulate spectral peaks.[20,21] A maximum number of peaks m_p with a default value of *ten* was given as a parameter to the generator function, after which a random number is drawn $\mathcal{U}(0, m_p)$ to determine the final number of peaks. Every peak was inserted into a list of zeros in a similar fashion to cosmic rays and noise.

The versatility of this synthetic generator was in the ability to generate a diverse set of spectra that have their peaks obscured by a wide variety of artifacts. Each component was separated from the others, which means that wdata points and labels were generated for supervised learning tasks. Data were synthesized with a diversity of signal-to-noise ratio values by multiplying the noise and the signal components by independent random values. As a final step, the generator scales every element by measuring the maximum value of the summed synthetic spectrum. Then it divided the synthetic spectrum and each component by the measured value. This preserves the relative scale of every component with respect to the sum of all components. This means that any model trained using this generator could handle spectra that were Min–Max normalized. Min–Max normalization was a common preprocessing technique that avoids exploding signals in neural networks by scaling the input values to the closed interval [0, 1]. Using this method, the problem of modeling was simplified for

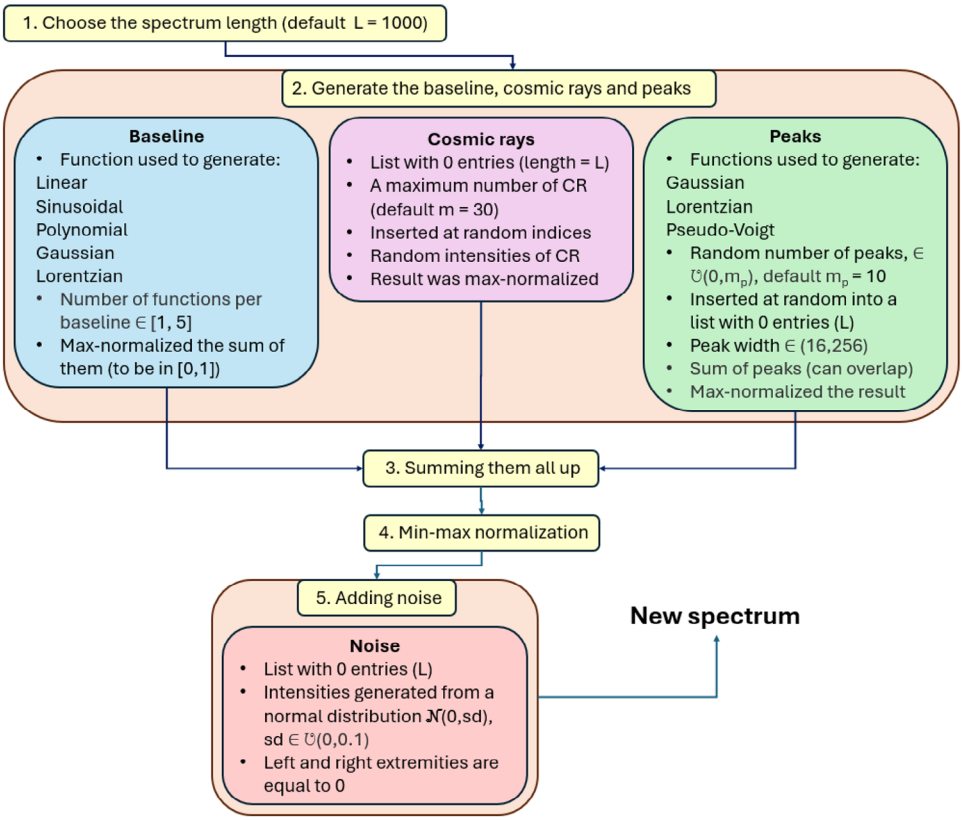

Figure 7. Overview of the synthetic data generator for Raman spectra. Baseline, cosmic rays, peaks, and noise are generated independently. 1) Spectral length can be freely chosen, with a default length of 1000 data points. 2) The baseline is created as a random sum of 1 to 5 curves, each selected from linear, sinusoidal, polynomial, Gaussian, or Lorentzian functions. The resulting baseline is max-normalized (i.e., scaled so the maximum is one and the minimum is 0). A random number of cosmic rays (default: 30) are added, each with a random intensity and placed at a random index. The result is max-normalized. A random number of peaks (default: 10) are generated. Each peak has a randomly chosen width (between 16 and 256 frequency units) and a shape selected from Gaussian, Lorentzian, or pseudo-Voigt functions. Peaks are inserted at random positions and may overlap. The result is max-normalized. 3-4) The baseline, cosmic rays, and peaks are summed together and the combined signal is again max-normalized. 5) Gaussian noise (mean 0) is added to the spectrum.

the generated spectra and guarantee stable model performance on spectra processed by Min–Max normalization.

Two Deep Learning Architectures: *Core structure*: Two different deep learning architectures were used to extract the B, CR, N, and P components from the input spectra with length *L*. Both models used convolutional layers from input to output layers, with the aim of keeping the total number of parameters as small as possible. The schematic differences between them is shown in **Figure 8**. The lack of dense layers allowed the networks to work with variable input lengths, a novelty and a useful feature for Raman denoising models. The benefit of dense layers was that information from every frequency in the input spectrum could be used to devised the output. The downside of using dense layers was that the number of parameters easily explodes, leading to models that could easily overfit to training data. To allow for information from the whole input layer while avoiding dense layers, each model employed two sequential convolutional layers meant to reduce the size of the input spectrum. The result of these layers was a latent representation of shape (*L*, 256) which was truncated via a global maximum pooling layer to a vector of shape (1, 256). The dot product between this and the input of shape (*L*, 1) was then calculated, leading to a latent representation of shape (*L*, 256). By expanding the dimensions of this representation to (*L*, 256, 1), 2D convolutions were employed on this representation. By using two of these 2D convolutions, while keeping a large stride on the middle dimension, the model was allowed to use multiple representations of the input at once. A latent representation of (*L*, 1, *F*) was obtained, where *F* was the number of filters computed by the models (20 for the *standard* model and 10 for the *ensemble* model). The shape of the latent representation is (256, 1) for the *standard* model and (144, 1) for the *ensemble* model. For all 1D convolutions, a padding technique was used that extended the input to the left and to the right with flat lines, i.e., for a spectrum with signal [*i*, ..., *j*] the left end was padded with *i*-values and the right end with *j*-values. This was done to avoid cases where kernel activations were disturbed due to sharp shifts at the extremities of the input spectrum. Each 1D convolutional layer was succeeded by a batch normalization layer followed by a *leakyReLU* activation layer, the output layers being the only exception to this pattern. Each model had four output layers, one for each component. The B and P outputs used average pooling layers to smooth the output before returning it. The B, CR and P outputs used *ReLU* activation, and the N output used the *tanh* activation function to preserve negative noise intensities.

Standard Architecture: The standard architecture resembled a standard multi-output CNN. After using the dot product mentioned above, the representation was reshaped to (length, 20) which was then processed by two 1D convolutional layers, leading to a latent representation with a maintained shape (length, 20). Three independent 1D convolutional layers were then used to compute the output for the B, CR, and P components. The N component was then computed by subtracting each predicted component from the original input. Then each component was returned in the following order: B, CR, N, and P. This model was the larger variant of the two, containing a total of 585,779 parameters. In each layer following the dot product layer, the convolutional layers

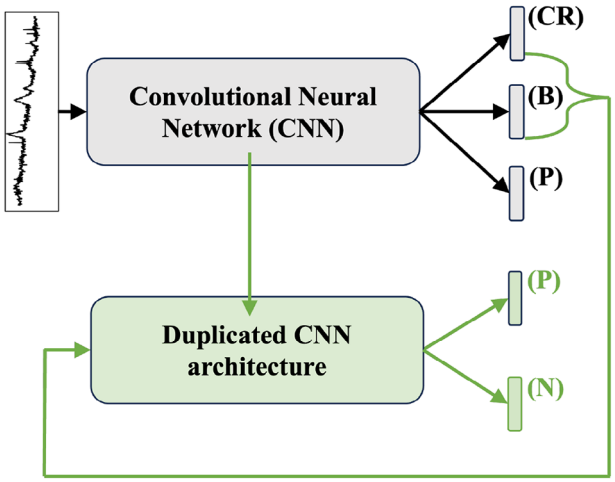

Figure 8. Simplified diagram of the standard and the ensemble model architectures. Both architectures share a common structure (in gray) composed of convolutional layers, batch normalization, and LeakyReLU activation. In the standard model, the noise component N is not predicted directly; instead, it is computed by subtracting the predicted cosmic rays (CR), baseline (B), and peak (P) from the input. In contrast, the ensemble model includes an additional CNN pipeline (in green). This second pipeline takes the CR and B predictions from the first pass as input and processes them further (together with a specific internal representations of the first pipeline, indicated with the green arrow) to generate refined predictions of P and N. The detailed architectures are in the Supporting Information (Figures 2 and 3).

all compute 20 filters. The output of B was smoothed using an average pooling layer of size 33. The peaks were also smoothed with an average pooling layer using a smaller pool size of four. A diagram showing the architecture of this model is in the Figure 2 (Supporting Information). The code for the trained model, including the numerical setup of all layers, is available on https://seafile.utu.fi/d/80db1f613ef74da0a050/ Seafile and https://github.com/JoelSjoberg/Comprehensive-Raman-Spectral-Analysis-Using-Deep-Learning/tree/main Github.

Ensemble Architecture: The ensemble architecture is based on the sequential subtraction of each component from the input, using the residue to predict the next component. The model begins by using the structure described above, the dot product strategy, before estimating B and CR. This model used an average pooling layer in order to provide a smooth representation of B. The predicted B and CR were then subtracted from the input. This residue vector was then fed through another pipeline, again using the dot product and 2D convolutions to be processed by two 1D convolutions before two independent 1D convolutional layers each process the latent representation to predict P and N. In this model, all components were predicted using learnable parameters to stabilize model training. In addition, a residual connection was made between the parts of the model predicting B and CR and the other part predicting P and N. This link helps to avoid the vanishing gradient problem while slightly accelerating the training process. The model was smaller than the standard model, containing 280,592 parameters. Each component output used an *ReLU* activation function, in place of the *LeakyReLU* activations in the preceding layers, to guarantee a minimum output signal of 0 (N being the only exception to this, using the *tanh* function). A diagram showing the composition of the ensemble model is in the Figure 3 (Supporting Information). The code for the trained model, including the numerical setup of all layers, is available on https://seafile.utu.fi/d/80db1f613ef74da0a050/ Seafile and https://github.com/JoelSjoberg/Comprehensive-Raman-Spectral-Analysis-Using-Deep-Learning/tree/main Github.

Model Training and Validation: The models on synthetic spectra were trained with 750 wavelengths and a maximum of five peaks. These values were chosen to accelerate the generator since a shorter spectral length could be generated faster. The shorter length also allowed for larger baseline slopes, which helps the model to adapt to more drastic baseline changes. The widths of each generated peak were set as a random value between 5 and 256 pixels wide. These values were chosen to allow for as much diversity in the spectra as possible. The models were validated on synthetic spectra with 2000 wavelengths and with peak widths between 5 and 300 pixels wide. This was intended to evaluate the model's performance on spectra of a different type than those used in training: longer and with potentially wider and narrower peaks.

Code Availability

The code, data, and models developed in this project are available on https://seafile.utu.fi/d/80db1f613ef74da0a050/ Seafile and https://github.com/JoelSjoberg/Comprehensive-Raman-Spectral-Analysis-Using-Deep-Learning/tree/main Github.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This project was partially supported by the Foundation of the University of Turku (the Niilo and Helmi Näsänen Fund and the Pentti and Tyyni Ekbom Fund), by the Swedish Cultural Foundation, and by the Core Program within the Romanian National Research, Development and Innovation Plan 2022-2027, carried out with the support of MRID (project no. 23020101(SIA-PRO), contract no. 7N/2022 and project PNRR-I8, contract CF 68), and by the Digital Europe Programme project LLMs4EU with ID. 101198470. [Correction added on July 8, 2025, after online publication: funding information in the "Acknowledgement" section has been added in this version.]

Open access publishing facilitated by Turun yliopisto, as part of the Wiley - FinELib agreement.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in Seafile at https://seafile.utu.fi/d/80db1f613ef74da0a050/ , reference number 1.

Keywords

deep learning, denoising, peak detection, raman spectroscopy

Received: March 6, 2025
Revised: May 21, 2025
Published online:

[1] P. R. Griffiths, in *Handbook of Vibrational Spectroscopy* (Eds: P. Griffiths, J. M. Chalmers), John Wiley & Sons, Ltd, New York **2006**, pp. 1–50.

[2] G. Keresztury, in *Handbook of Vibrational Spectroscopy* (Eds: P. Griffiths, J. M. Chalmers), John Wiley & Sons, Ltd, New York **2006**, pp. 1–50.
[3] X. X. Han, C. L. Haynes, Y. Ozaki, B. Zhao, *Nat. Rev. Methods Primers* **2021**, *1*.
[4] E. Ploetz, S. Laimgruber, S. Berner, W. Zinth, P. Gilch, *Appl. Phys. B* **2007**, *87*, 389.
[5] F. Hu, L. Shi, W. Min, *Nat. Methods* **2019**, *16*, 830.
[6] S. Fang, S. Wu, Z. Chen, C. He, L. L. Lin, J. Ye, *TrAC, Trends Anal. Chem.* **2024**, *172*, 117578.
[7] E. Smith, G. Dent, *Modern Raman spectroscopy: a practical approach*, John Wiley & Sons, New York **2019**.
[8] B. Barton, J. Thomson, E. L. Diz, R. Portela, *Appl. Spectrosc.* **2022**, *76*, 1021.
[9] C. A. Lieber, A. Mahadevan-Jansen, *Appl. Spectrosc.* **2003**, *57*, 1363.
[10] J. Zhao, H. Lui, D. I. McLean, H. Zeng, *Appl. Spectrosc.* **2007**, *61*, 1225.
[11] G. Sheehy, F. Picot, F. Dallaire, K. Ember, T. Nguyen, K. Petrecca, D. Trudel, F. Leblond, *J. Biomed. Opt.* **2023**, *28*, 025002.
[12] Z.-M. Zhang, S. Chen, Y.-Z. Liang, *Analyst* **2010**, *135*, 1138.
[13] D. A. Whitaker, K. Hayes, *Chemom. Intell. Lab. Syst.* **2018**, *179*, 82.
[14] S. Barton, S. Alakkari, K. O'Dwyer, T. Ward, B. Hennelly, *Sensors* **2021**, *21*.
[15] A. Savitzky, M. J. Golay, *Anal. Chem.* **1964**, *36*, 1627.
[16] M. Kazemzadeh, M. Martinez-Calderon, W. Xu, L. W. Chamley, C. L. Hisey, N. G. Broderick, *Anal. Chem.* **2022**, *94*, 12907.
[17] J. Wahl, *Multimodal applications in medical technology that utilize Raman spectroscopy*, Ph.D. thesis, Luleå University of Technology **2022**.
[18] Q. Jiao, X. Guo, M. Liu, L. Kong, M. Hui, L. Dong, Y. Zhao, *Chemom. Intell. Lab. Syst.* **2023**, *235*, 104779.
[19] M. Han, Y. Dang, J. Han, *Sensors* **2024**, *24*, 3161.
[20] M. N. Schmidt, T. S. Alstrøm, M. Svendstorp, J. Larsen, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Piscataway, NJ **2019**, pp. 2757–2761.
[21] J. Wahl, M. Sjödahl, K. Ramser, *Appl. Spectrosc.* **2020**, *74*, 427.
[22] T. Chen, Y. Son, A. Park, S.-J. Baek, *Analyst* **2022**, *147*, 4285.
[23] Y. Liu, *Chemom. Intell. Lab. Syst.* **2021**, *213*, 104317.
[24] H. He, M. Cao, Y. Gao, P. Zheng, S. Yan, J.-H. Zhong, L. Wang, D. Jin, B. Ren, *Nat. Commun.* **2024**, *15*, 754.
[25] J. Hu, G. J. Chen, C. Xue, P. Liang, Y. Xiang, C. Zhang, X. Chi, G. Liu, Y. Ye, D. Cui, D. Zhang, X. yu, H. Dang, W. Zhang, J. Chen, Q. Tang, P. Guo, H.-P. Ho, Y. Li, L. Cong, P. P. Shum, *Light: Sci. Appl.* **2024**, *13*, 52.
[26] B. Manifold, E. Thomas, A. T. Francis, A. H. Hill, D. Fu, *Biomed. Opt. Express* **2019**, *10*, 3860.
[27] J. Wahl, E. Klint, M. Hallbeck, J. Hillman, K. Wårdell, K. Ramser, *Biomed. Opt. Express* **2022**, *13*, 6763.
[28] Z.-M. Zhang, S. Chen, Y.-Z. Liang, *Analyst* **2010**, *135*, 1138.
[29] M. Han, Y. Dang, J. Han, *Sensors* **2024**, *24*, 3161.
[30] F. Conti, M. D'Acunto, C. Caudai, S. Colantonio, R. Gaeta, D. Moroni, M. A. Pascali, *Sci. Rep.* **2023**, *13*, 7282.
[31] G. Berlanga, Q. Williams, N. Temiquel, *Machine Learning Raman Open Dataset (MLROD)*, **2022**.
[32] G. Berlanga, Q. Williams, N. Temiquel, *Earth Space Sci.* **2022**, *9*, 2021EA002125.
[33] S. Fang, S. Wu, Z. Chen, C. He, L. L. Lin, J. Ye, *TrAC, Trends Anal. Chem.* **2024**, *172*, 117578.
[34] D. DePaoli, É. Lemoine, K. Ember, M. Parent, M. Prud'homme, L. Cantin, K. Petrecca, F. Leblond, D. C. Côté, *J. Biomed. Opt.* **2020**, *25*, 050901.
[35] A. Lita, J. Sjöberg, D. Păcioianu, N. Siminea, O. Celiku, T. Dowdy, A. Păun, M. R. Gilbert, H. Noushmehr, I. Petre, M. Larion, *Neuro-Oncol.* **2024**, *26*, 1994.
[36] K. Szczepankiewicz, A. Popowicz, K. Charkiewicz, K. Nałęcz-Charkiewicz, M. Szczepankiewicz, S. Lasota, P. Zawistowski, K. Radlak, *Sci. Rep.* **2023**, *13*, 16887.
[37] Z. J. Wang, *Sci. Rep.* **2021**, *11*, 10610.
[38] K. Simonyan, A. Vedaldi, A. Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps **2014**, arXiv:1312.6034 [cs].
[39] M. D. Zeiler, R. Fergus, in *Computer Vision – ECCV 2014*, (Eds: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars), Springer International Publishing, Cham **2014**, pp. 818–833.
[40] C.-L. Chin, C.-E. Chang, L. Chao, *ACS Sens.* **2025**, *10*, 2652.
[41] Y. Liu, J. Wu, Y. Wang, S. Dong, *AIP Adv.* **2022**, *12*, 08521.