PAPER

Baseline correction of Raman spectral data using triangular deep convolutional networks

Tiejun Chen, [ID] [a] YoungJae Son, [ID] [a] Changqing Dong [ID] [b] and Sung-June Baek [ID] *[a]

Raman spectroscopy requires baseline correction to address fluorescence- and instrumentation-related distortions. The existing baseline correction methods can be broadly classified into traditional mathematical approaches and deep learning-based techniques. While traditional methods often require manual parameter tuning for different spectral datasets, deep learning methods offer greater adaptability and enhance automation. Recent research on deep learning-based baseline correction has primarily focused on optimizing existing methods or designing new network architectures to improve correction performance. This study proposes a novel deep learning network architecture to further enhance baseline correction effectiveness, building upon prior research. Experimental results demonstrate that the proposed method outperforms existing approaches by achieving superior correction accuracy, reducing computation time, and more effectively preserving peak intensity and shape.

1. Introduction

Raman spectroscopy technology requires only a small amount of samples to perform rapid, non-destructive, and repeatable qualitative and quantitative analysis. The combination of a diode laser and a charge-coupled device has enabled the miniaturization of Raman spectrometers and improved their sensitivity.[1,2] As a result, Raman spectroscopy is widely applied in various fields, including the determination of drug authenticity,[3] poison detection,[4] medical diagnostics,[5] and food analysis.[6]

Raman spectra typically contain two main types of noise: background noise, also known as baseline, and additional noise caused by external conditions and auto-fluorescence.[7] The presence of these noise affects the accuracy of subsequent studies and analyses. Consequently, pre-processing steps, particularly baseline correction, have become essential in most Raman spectroscopy applications. Over the years, various methods for baseline correction have been developed, garnering increasing attention from researchers.

Baseline correction methods are generally categorized into two main approaches: traditional mathematical methods and deep learning-based methods. Traditional methods include wavelet transform methods,[8–10] polynomial fitting methods,[11,12] and penalized least squares methods.[13–15] In contrast, deep learning methods primarily utilize deep neural networks to perform regression tasks to predict and correct baselines. Research on deep learning-based baseline correction has focused on designing more efficient network structures or improving existing methods to address their limitations.

Among traditional baseline correction methods, the asymmetrically reweighted penalized least squares (arPLS) method[13] and the adaptive smoothness parameter penalized least squares (asPLS) method[14] are both based on penalized least squares. The arPLS method is considered one of the most effective methods for baseline correction, as it prevents the overestimation of the baseline and ensures satisfactory curve fitting in non-peak regions. However, it tends to misinterpret the tails of peaks as non-peak regions due to their relatively low intensity. The asPLS method mitigates this limitation by introducing a coefficient vector, which dynamically adjusts the arPLS smoothness parameter. This adjustment assigns a larger value to peak regions and a smaller value to non-peak regions, thereby improving the accuracy of baseline correction in areas with low intensity.

These traditional baseline correction methods share a common characteristic: they require human discretion to obtain optimal corrected results. In other words, these methods typically rely on manually selecting appropriate wavelet bases, polynomial orders, and balance parameters to achieve ideal correction outcomes. Consequently, practical applications demand significant time and effort to fine-tune these parameters based on prior knowledge to enhance correction accuracy.

Deep learning-based baseline correction methods effectively address the limitations of traditional approaches. Once the deep learning model is trained, it can be applied directly without additional adjustments, significantly reducing the

[a]Department of Intelligent Electronics and Computer Engineering, Chonnam National University, Gwangju 61186, South Korea. E-mail: tozero@jnu.ac.kr; Tel: +82 62-530-1795
[b]Department of Auckland Bioengineering Institute, University of Auckland, 6/70 Symonds street, Grafton, Auckland 1010, New Zealand

time and effort required. Furthermore, deep learning methods generally demonstrate superior baseline correction performance compared to traditional techniques, leading to increased interest in applying these models to Raman spectroscopy.

However, training deep learning models requires large amounts of labeled data, posing challenges for the scarce and often unlabeled Raman spectral data. To address this issue, Liu manually extracted peak and baseline information from actual spectral data and generated training data by randomly combining peaks, baselines, and additive noise.[16] However, this approach cannot always guarantee the quality of the generated data. In response, Chen *et al.* proposed an improved method that uses mathematical models to generate peaks, baselines, and additive noise separately, and constructing training data by randomly combining these components to enhance data quality.[17]

In this study, we introduce a novel triangular deep convolutional network (TDCN) for baseline correction, which is trained using synthetically generated spectral data.[17] Through both qualitative and quantitative analyses of simulated and actual Raman spectra, the proposed model demonstrates superior baseline correction performance when compared to traditional methods, while also significantly reducing inference time, thus enhancing the overall efficiency. Additionally, to evaluate the effectiveness of our model, we compared it with an existing deep learning model, confirming the improvements achieved by the proposed model.

2. Simulated training data

A large amount of actual Raman spectral data is difficult to obtain, while deep learning models typically require a large volume of data for effective training. Simulating training data using mathematical methods has become the key to solving the issue. Therefore, a few studies have begun to train their deep learning models through data simulation.[18,19] In this study, we employed a method capable of generating more natural and high quality simulated spectra for training data.

The simulated spectral data contain a pure spectral signal (peak), baseline, and additive noise, expressed mathematically as follows:

$$s[i] = (1 - \beta)p[i] + \beta b[i] + n[i], \tag{1}$$

where $s[i]$ indicates the raw spectral data, $p[i]$ expresses the peak and $b[i]$ and $n[i]$ are baseline and additive noise, respectively. Here, $s[i]$ can be regarded as the input to the deep learning model, while $b[i]$ serves as the target data, *i.e.*, the output.

The peak $p[i]$ is simulated by using the following Hanning window function:

$$p[i] = 0.5 - 0.5 \cos\left[\frac{2\pi i}{N-1}\right], \quad 0 \le i < N. \tag{2}$$

The value of peak width, N, was set to a range of 5–21, and the minimum height of the peak was set to 0.05 in this experiment. The number of spectral peaks was set between 5 and 15.

The baseline was modeled by cubic spline interpolation using 2–7 anchor points generated randomly. Using these anchor points, various baselines were generated for better prediction performance. The additive noise was represented by Gaussian white noise.

The final simulated spectra were generated by mixing the three elements mentioned above. The baseline mixing ratio, β, ranges from 0.1 to 0.8, while the peak mixing ratio is $1 - \beta$. To ensure compatibility with the deep learning model, the data length was set to 512 and the intensity was normalized to a range of 0–1 using the min–max normalization method.

Fig. 1 shows an example of a simulated spectrum and its baseline. Compared to the spectral data generation method proposed by Liu,[16] this method produces more natural and high-quality spectra. Additionally, the training data generated by this method are more diverse, enhancing the representativeness and applicability of the dataset. In this study, 128 000 training samples and 32 000 validation samples were generated to train the deep learning model.

3. Proposed deep-learning architecture

Deep learning technology has been successfully applied across various research fields, delivering outstanding results. In particular, convolutional neural networks (CNNs) have gained significant attention due to their superior feature extraction capabilities and fewer parameters compared to fully connected neural networks. As a result, many studies have developed custom deep learning models based on CNN architectures.[20,21]

One of the most well-known deep CNN architectures, ResNet, employs a residual learning framework to address the vanishing gradient problem that arises from increasing network depth. This framework enables the extraction of more representative features from the input data.[22] The residual framework primarily includes two types of shortcut connections: identity shortcuts and projection shortcuts. The projection shortcut connection adjusts the output feature size of the
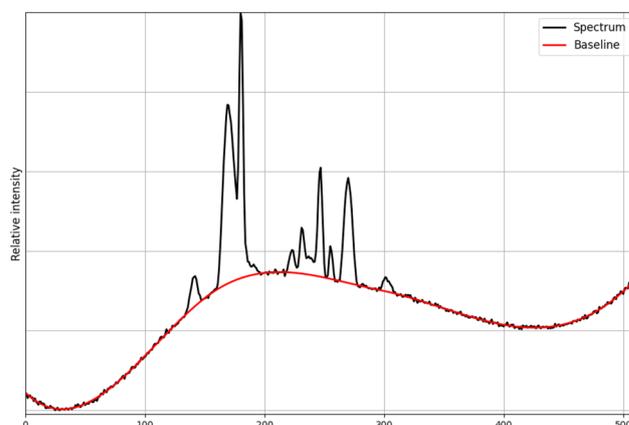

Fig. 1 An example of a simulated spectrum and its baseline.

previous layer through downsampling, allowing it to be added to the output features of the subsequent layer.

The UNet architecture was originally designed for medical image segmentation.[23] It consists of two main components: a contracting path and an expansive path. The contracting path functions similarly to a conventional deep CNN, extracting features from the input data. The expansive path performs upsampling on the feature maps and integrates them with features from the contracting path through skip connections. This skip connection framework allows UNet to recover lost pixel information, enhancing the accuracy of segmentation.

In this study, we propose a deep learning network architecture called the Triangular Deep Convolutional Network (TDCN), which integrates design concepts from both ResNet and UNet. As illustrated in Fig. 2(a), the network features a triangular-shaped structure composed of multiple convolutional cells. Each cell can be constructed in different ways, as depicted in Fig. 2(b), (c) and (d). To distinguish and describe their characteristics, we refer to these cells as simpleCell, simpleResCell, and ResCell, respectively. A common feature of these cells is the fusion of outputs from surrounding cells through addition, while their primary difference lies in the implementation of the residual block.

Let $x^{i,j}$ represent the output from the convolutional cell $C^{i,j}$ in a deep learning network, where i denotes the row index and j denotes the column index. The output of each cell $x^{i,j}$ can be computed using the following formula:

$$x^{i,j} = \begin{cases} \mathbf{C}(\mathbf{D}(x^{i-1,j})), & i>0, \quad j=0 \\ \mathbf{C}(x^{i,j-1}, \mathbf{U}(x^{i+1,j-1})), & i=0, \quad j>0 \\ \mathbf{C}(\mathbf{D}(x^{i-1,j}), x^{i,j-1}, \mathbf{U}(x^{i+1,j-1})), & i>0, \quad j>0. \end{cases} \quad (3)$$

Here, the cell, $\mathbf{C}(\cdot)$, is an integrated convolution block applied to the fused features from the preceding cell's output. The functions $\mathbf{D}(\cdot)$ and $\mathbf{U}(\cdot)$ correspond to down-sampling and up-sampling operations, which are implemented using a convolutional layer (Conv1d) and a transposed convolutional layer (ConvTranspose1d), respectively. It is important to note that Fig. 2 only illustrates the structure of the cell outlined by the red dashed line, while the structures of other cells are adapted based on their inputs. For instance, the cell, $C^{0,0}$, receives the raw spectrum as input, making its structure relatively simple and consisting of a single-input convolution.

The key distinction between SimpleCell and SimpleResCell lies in the use of residual connections. SimpleResCell employs a residual structure, whereas SimpleCell does not. Meanwhile, the difference between SimpleResCell and ResCell lies in the scope of the residual structure: SimpleResCell applies the residual structure only after fusing outputs from different cells, whereas ResCell applies it throughout the entire cell.

We selected ResUNet[17] as the deep learning method for comparison. This model combines the advantages of ResNet and UNet. While its overall structure resembles UNet, ResUNet incorporates residual connections, which enhance the performance compared to both ResNet and UNet. To comprehensively evaluate the effectiveness of the proposed approach, we used ResUNet as a benchmark model, enabling a direct comparison with our method and an indirect comparison with ResNet and UNet.

A distinctive feature of the proposed model is its simultaneous execution of contraction and expansion within each cell, unlike UNet, which separates the contracting and expanding paths. This architecture allows the model to receive and
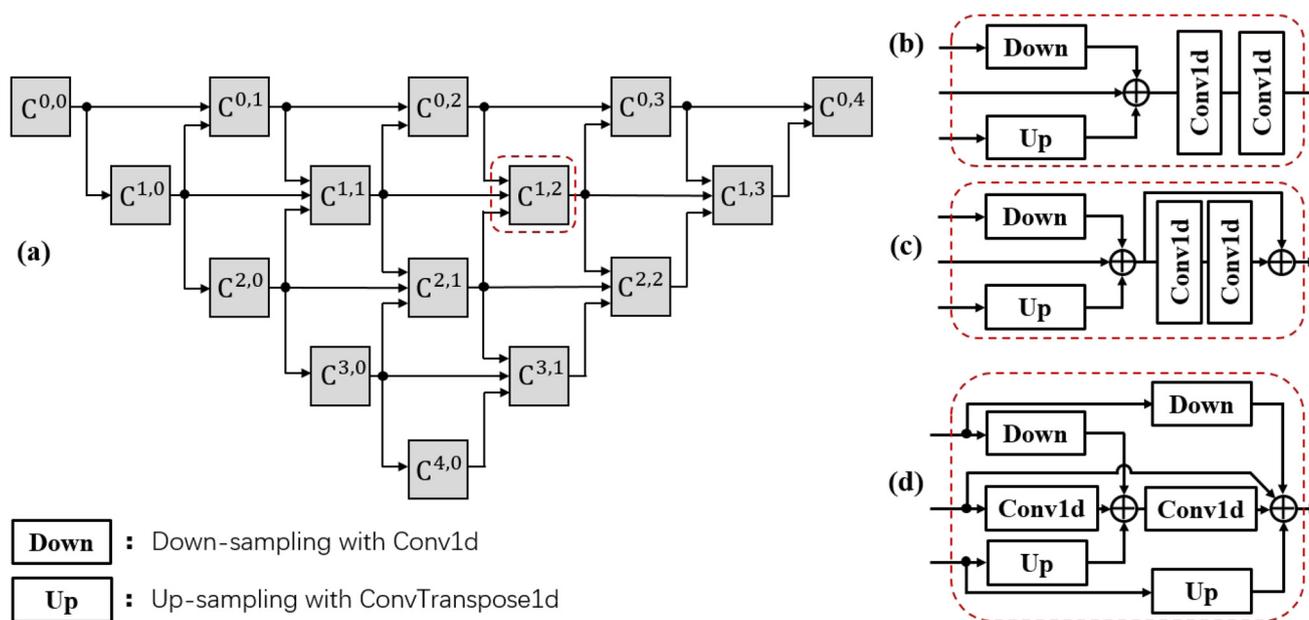

Fig. 2 Architecture of the proposed TDCN: (a) overall network structure, (b) SimpleCell unit, (c) SimpleResCell unit, and (d) ResCell unit.

integrate multi-dimensional outputs from adjacent cells, effectively capturing spatial relationships within the data.

4. Experiments and results

All experiments in this study were conducted using Python version 3.11.10 and the deep learning model was trained on a computer equipped with GeForce RTX 3080 GPU using the PyTorch deep learning framework.

The experimental results are presented in two parts. The first part utilizes simulated spectra with standard baselines to assess the efficacy of the proposed method. Specific types of simulated spectral signals are employed for qualitative analysis, while a generated test dataset is used for quantitative analysis. The second part is based on actual Raman spectra to demonstrate the baseline correction effectiveness of the proposed method, while its results are compared to those of traditional methods to highlight the differences between their approaches.

To comprehensively evaluate the performance of the proposed method, we chose the existing traditional mathematical method and the deep learning method with proven efficacy for comparative experiments. In the analysis of results, due to the absence of significant qualitative differences between deep learning methods, our focus is on visually illustrating the differences between deep learning and traditional methods through qualitative analysis. Additionally, we conducted a quantitative comparison of the baseline correction performance of each method to show the superiority of the proposed approach.

4.1 Data preparation and training strategy

In the experiments, 128 000 training samples were generated to train the deep learning model, with an additional 32 000 and 20 000 samples generated for validation and quantitative analysis, respectively. To qualitatively evaluate the feasibility and superiority of the proposed method for actual Raman spectra, experiments were conducted using spectra from 10 different substances recorded using two distinct Raman spectroscopy systems. The first system was a Renishaw 2000 Raman microscope equipped with a 514.5 nm argon ion laser and the second was an inVia Inspector portable Raman system utilizing a 632.8 nm He–Ne laser.

In the experiments, the optimization algorithm for the deep learning model was the adaptive moment estimation (Adam) algorithm. The learning rate was set to 5×10^{-4} and the batch size was set to 500. The root mean square error (RMSE) was used as the training loss function, while the mean absolute error (MAE) served as the validation loss function. The RMSE and MAE are defined as follows:

$$\text{RMSE} = \left[\sum_{i=0}^{N-1} (y_i - z_i)^2 / N \right]^{\frac{1}{2}} \quad (4)$$

$$\text{MAE} = \sum_{i=0}^{N-1} |y_i - z_i| / N \quad (5)$$

The deep learning model was trained for a total of 1000 epochs. If the validation loss decreased, the current model weights were saved. The training strategy included a learning rate decay, where the learning rate was reduced to 4/5 of its current value if the validation loss value did not change over a span of 75 epochs. Training was terminated if the learning rate was reduced to 1/8 of the original value.

4.2 Baseline correction with the synthesized spectra

The baseline correction of Raman spectral data using the deep learning model involves two steps. The first step is the training process, in which the deep learning model is trained using simulated spectral data. The input to the model consists of simulated spectral data and the output is the target data representing the baseline. The second step is the inference process, where the trained model is used to predict the baseline of the actual Raman spectral data. The desired corrected spectral data were then obtained by subtracting the predicted baseline from the measured spectral data.

The effectiveness and superiority of the proposed method were verified using simulated data. To facilitate qualitative comparison of the performance of the proposed method and traditional methods under varying peak and baseline conditions, we generated simulated peaks using a Gaussian function and constructed baselines using quadratic and cubic polynomials. The simulated pure signals, which consist of three Gaussian peaks, were constructed as follows:

$$p[i] = 0.3\text{e}^{-\left(\frac{i-100}{5}\right)^2} + 1.0\text{e}^{-\left(\frac{i-350}{10}\right)^2} + 0.5\text{e}^{-\left(\frac{i-370}{5}\right)^2}. \quad (6)$$

where $i = 0, 1, ..., 511$. The intensities of the three peaks are 0.3, 1.0, and 0.5, respectively. Simulated spectra with varying signal-to-noise ratios (SNRs) were generated by adding Gaussian white noise. The SNR can be obtained according to the following equation:

$$\text{SNR}_{\text{dB}} = 10 \log_{10}\left(\frac{P_s}{P_n}\right). \quad (7)$$

where P_s and P_n are the power of the peaks and the noise. The SNRs for the low-noise and high-noise signals were 30.7 dB and 18.7 dB, respectively. The simulated pure signal, low-noise signal, and high-noise signal are shown in Fig. 3(a). To better approximate real spectra, we generated pure signals containing both single and overlapping peak types. From the figure, it is evident that the simulated signal consists of a small single peak on the left and a larger, overlapping peak on the right.

Fig. 3(b) displays the generated simulated spectral signal alongside the baseline correction results obtained using both the traditional method and the proposed method. To comprehensively evaluate the performance of the proposed method, we introduced additive noise and baselines of varying intensities. The peaks and baselines were combined at different ratios. As shown in the figure, the height of the peaks varies
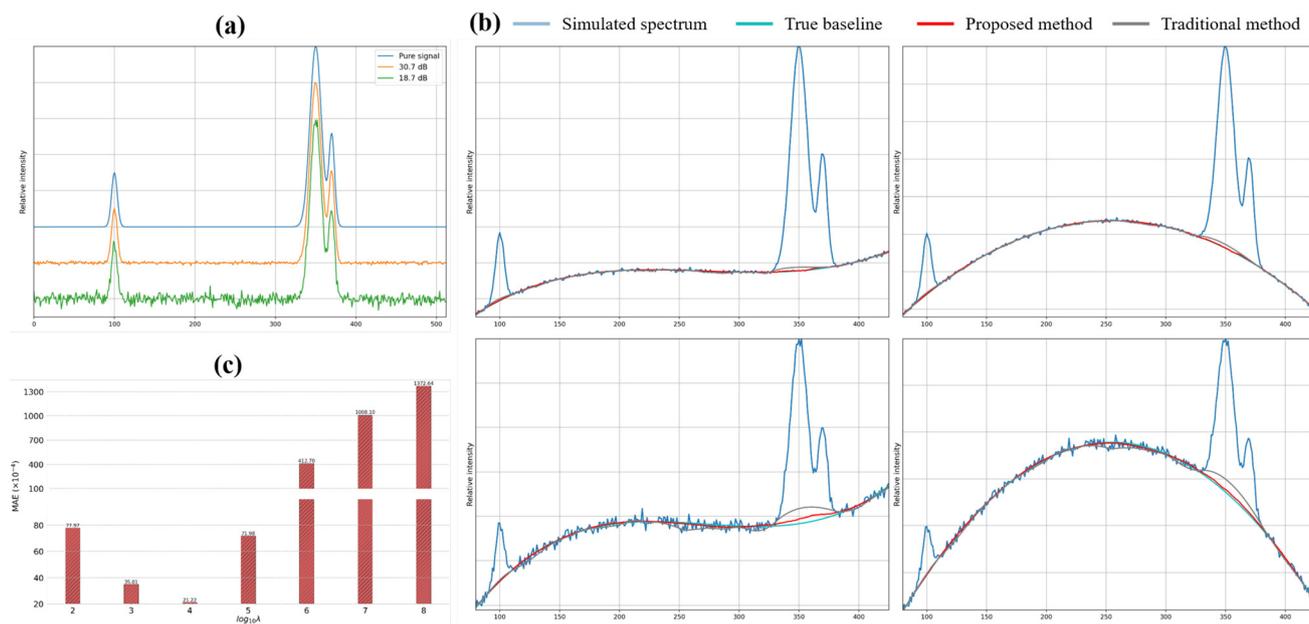
Fig. 3 Examples of simulated spectra. (a) Simulated pure signals with no noise, low noise, and high noise. (b) Baseline correction results for the proposed method and the traditional method applied to the simulated spectra. (c) MAE with varying λ values for the asPLS method.

with the combination ratio, even when the same baseline is used.

The proposed method was compared with the asPLS method for baseline correction, which has demonstrated excellent performance in its respective domain. Prior to the experiments, optimal parameters for asPLS were selected through preliminary testing to ensure the best results using the 20 000 test data. The results from all experiments with simulated data were evaluated using the RMSE and the MAE.

The smoothness parameter λ of asPLS significantly influences the baseline correction results. If λ is set too large or too small, it will adversely affect the correction performance. It is recommended that λ varies on a logarithmic scale[24] with values ranging from 10^2 to 10^8 for optimal parameter selection. As shown in Fig. 3(c), the MAE value is minimized when λ is set to 10^4 for baseline correction.

From the four baseline correction results, it is evident that the traditional method generally overestimates the baseline compared to the proposed method, particularly in regions with wider peaks. Moreover, for data with high baseline ratios and high-intensity additive noise, the overestimation of the baseline by the traditional method is more pronounced. In the spectra with high-intensity additive noise, the presence of noise significantly impacts the baseline correction performance of the traditional method, particularly in non-peak regions. Overall, qualitative analysis indicates that the proposed method offers significant advantages over the traditional method, particularly under complex conditions involving both additive noise and varying baseline characteristics.

To further evaluate the performance, quantitative comparative experiments were conducted between the proposed method, traditional methods, and other deep learning

Table 1 The results of quantitative analysis for various methods

Methods	MAE ($\times 10^{-4}$)	RMSE ($\times 10^{-4}$)	Variance ($\times 10^{-7}$)	Param.
SimpleCell	5.145	6.719	1.004	0.877 M
SimpleResCell	4.876	6.342	1.029	0.877 M
ResCell	5.347	7.033	1.196	1.274 M
ResUNet	5.062	6.609	1.097	0.898 M
asPLS	21.22	63.86	22.542	N/A

approaches using a test dataset of 20 000 samples. Table 1 presents the quantitative analysis results for various methods, including the MAE, RMSE, and MAE variance for the test data. It is clear from the table that the deep learning-based baseline correction method outperforms the traditional method in terms of overall performance. Among the proposed network architectures, the SimpleCell model demonstrates strong performance in terms of MAE variance. However, the SimpleResCell model achieves the lowest MAE and RMSE values, indicating superior baseline correction accuracy. While ResUNet also performs well overall, the SimpleResCell model outperforms it across all evaluation metrics.

Deep learning methods are slightly faster than the traditional method, while the proposed model requires marginally more time than ResUNet due to the integration of outputs from surrounding cells through element-wise addition. However, it is important to note that the inference time of the traditional method can vary depending on the maximum number of iterations and the adjustment of termination conditions. Similarly, the inference time of the deep learning methods may vary slightly based on the implementation of the output stage.
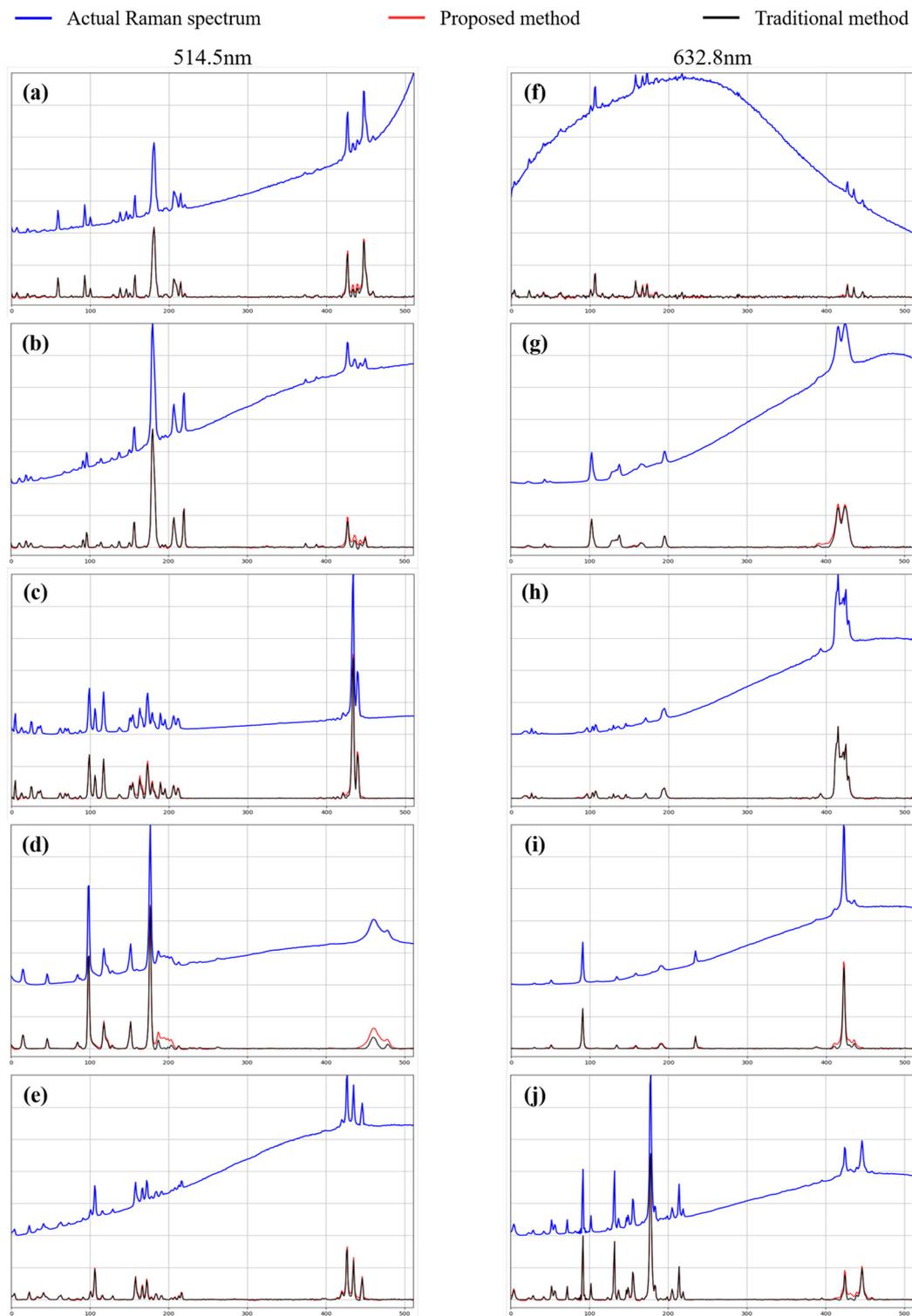
Fig. 4 The results of baseline correction for the proposed method and the traditional method using actual Raman spectra recorded using different instruments at 514.5 nm and 632.8 nm. (a–e): 2,6-dinitrotoluene, 2-methyl-1,3,5-trinitrobenzene, octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine, ammonium dinitramide, and 1,3,5-trinitroperhydro-1,3,5-triazine. (f–j): composition C-4, ethylene glycol, hexane, acetone, and 2-nitrotoluene.

4.3 Actual Raman spectra

The spectra of a set of 5 substances (2,6-dinitrotoluene, 2-methyl-1,3,5-trinitrobenzene, octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine, ammonium dinitramide, and 1,3,5-trinitroperhydro-1,3,5-triazine) and another set of 5 substances (composition C-4, ethylene glycol, hexane, acetone, and 2-nitrotoluene) recorded using different instruments were used for qualitative analysis of the proposed method. The Raman spectra of these materials contain diverse baselines and additive noise, making them suitable for validating the proposed method.

The baseline correction results are presented in Fig. 4. To highlight the differences between the deep learning method and traditional method for baseline correction of real Raman spectra, we also included the correction results from the traditional method. Consistent with the analysis of simulated data, the traditional method tends to overestimate the baseline, leading to inaccurate preservation of the corrected peak intensities. In contrast, the corrected spectra obtained using the proposed method appear more natural and desirable.

The real Raman spectra used in this experiment exhibit various baselines. The experimental results demonstrate that the proposed method effectively removes these baselines across different spectra. Since the proposed method performs well not only with synthetic spectra but also with real spectra, it is expected to be a valuable preprocessing technique for a range of spectra, including FT-IR, FT-NIR, and others, in addition to Raman spectra.

5. Conclusions

This study presents a novel deep learning network architecture for baseline correction of Raman spectra. We designed three distinct cell structures and developed the corresponding deep learning models. Additionally, to ensure the quality of the training data, we implemented a data generation method that produces more natural and high-quality simulated spectra, thereby enhancing the model's generalization capability.

Through both qualitative and quantitative analyses of all methods on simulated spectra, we have verified the overall superiority of the proposed method. The experimental results demonstrate that this method accurately estimates the baseline of spectra with varying baselines and additive noise, particularly in peak areas with wider amplitudes. Moreover, compared to traditional methods, the inference time of deep learning methods is significantly reduced. The proposed method also shows robust performance in the presence of higher levels of additive noise, with minimal impact on baseline correction in non-peak regions.

To further validate the effectiveness of the proposed method in practical applications, we conducted experiments using various Raman spectra recorded using different instruments. The results show that the baseline correction achieved by the proposed method is natural and reasonable for real Raman spectra. The experimental results align with the analysis of the simulated spectra, further demonstrating the superiority and broad applicability of the proposed method in baseline correction tasks for Raman spectra. We anticipate that the proposed method can be applied not only to Raman spectra but also to various IR spectra.

However, the proposed model exhibits a higher parameter growth rate as the network depth increases compared to the other models. Additionally, the integration of outputs from surrounding cells through element-wise addition increases the inference time. In the future, we plan to conduct further research to alleviate this computational requirement.

Data availability

Data for this article are available at **https://github.com/magicalBook/TriangularNetwork**.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

References

1 Q. Xu, H. Chen, S. Ye, Y. Zeng, H. Lu and Z. Zhang, *Anal. Methods*, 2021, **13**, 3414–3423.

2 Y. Cai, C. Yang, D. Xu and W. Gui, *Anal. Methods*, 2018, **10**, 3525–3533.

3 L. S. Lawson and J. D. Rodriguez, *Anal. Chem.*, 2016, **88**, 4706–4713.

4 E. L. Izake, *Forensic Sci. Int.*, 2010, **202**, 1–8.

5 K. Kong, C. Kendall, N. Stone and I. Notingher, *Adv. Drug Delivery Rev.*, 2015, **89**, 121–134.

6 M. Chai, W. Hasi, X. Ming, S. Han, G. Fang and Y. Bu, *J. Food Compos. Anal.*, 2024, **136**, 106793.

7 J. Hwang, N. Choi, A. Park, J.-Q. Park, J. H. Chung, S. Baek, S. G. Cho, S.-J. Baek and J. Choo, *J. Mol. Struct.*, 2013, **1039**, 130–136.

8 F. Qian, Y. Wu and P. Hao, *Opt. Laser Technol.*, 2017, **96**, 202–207.

9 C. G. Bertinetto and T. Vuorinen, *Appl. Spectrosc.*, 2014, **68**, 155–164.

10 Y. Hu, T. Jiang, A. Shen, W. Li, X. Wang and J. Hu, *Chemom. Intell. Lab. Syst.*, 2007, **85**, 94–101.

11 S.-J. Baek, A. Park, A. Shen and J. Hu, *J. Raman Spectrosc.*, 2011, **42**, 1987–1993.

12 F. Gan, G. Ruan and J. Mo, *Chemom. Intell. Lab. Syst.*, 2006, **82**, 59–65.

13 S.-J. Baek, A. Park, Y.-J. Ahn and J. Choo, *Analyst*, 2015, **140**, 250–257.

14 F. Zhang, X. Tang, A. Tong, B. Wang, J. Wang, Y. Lv, C. Tang and J. Wang, *Spectrosc. Lett.*, 2020, **53**, 222–233.

15 Z.-M. Zhang, S. Chen and Y.-Z. Liang, *Analyst*, 2010, **135**, 1138–1146.

16 Y. Liu, *Chemom. Intell. Lab. Syst.*, 2021, **213**, 104317.

17 T. Chen, Y. Son, A. Park and S.-J. Baek, *Analyst*, 2022, **147**, 4285–4292.

18 M. Han, Y. Dang and J. Han, *Sensors*, 2024, **24**, 3161.

19 M. Kazemzadeh, M. Martinez-Calderon, W. Xu, L. W. Chamley, C. L. Hisey and N. G. Broderick, *Anal. Chem.*, 2022, **94**, 12907–12918.

20 T. Chen and S.-J. Baek, *ACS Omega*, 2023, **8**, 37482–37489.

21 T. Chen, Y. Son and S.-J. Baek, *International Conference on Cloud Computing, Performance Computing, and Deep Learning (CCPCDL 2024)*, 2024, pp. 187–192.

22 K. He, X. Zhang, S. Ren and J. Sun, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 770–778.

23 O. Ronneberger, P. Fischer and T. Brox, *Medical Image Computing and Computer-Assisted Intervention–MICCAI* 2015: *18th International Conference, Munich, Germany, October* 5–9, 2015, *Proceedings, Part* III 18, 2015, 234–241.

24 P. H. Eilers, *Anal. Chem.*, 2003, **75**, 3631–3636.