석사학위논문

신경망 기반 자동화 접근을 통한 1차원 신호 데이터 처리 시스템

자동 매개변수 최적화 및 트랜스포머 기반 보정을 활용한 1차원 신호 데이터 처리 성능 향상

2026년

한 성 대 학 교 대 학 원

A I 응 용 학 과

A I 응 용 학 전 공

김 예 란

석 사 학 위 논 문
지 도 교 수 이 웅 희

신경망 기반 자동화 접근을 통한 1차원 신호 데이터 처리 시스템

자동 매개변수 최적화 및 트랜스포머 기반 보정을 활용한 1차원 신호 데이터 처리 성능 향상

A Study on a Neural Network-Based Automated System for One-Dimensional Signal Data Processing

2025년 12월 일

한 성 대 학 교 대 학 원

A I 응 용 학 과

A I 응 용 학 전 공

김 예 란

석 사 학 위 논 문
지 도 교 수 이 웅 희

신경망 기반 자동화 접근을 통한 1차원 신호 데이터 처리 시스템

자동 매개변수 최적화 및 트랜스포머 기반 보정을 활용한 1차원 신호 데이터 처리 성능 향상

A Study on a Neural Network-Based Automated System for One-Dimensional Signal Data Processing

위 논문을 공학 석사학위 논문으로 제출함

2025년 12월 일

한 성 대 학 교 대 학 원

A I 응 용 학 과

A I 응 용 학 전 공

김 예 란

김예란의 공학 석사학위 논문을 인준함

2025년 12월 일

심사위원장 공진우 (인)

심 사 위 원 이웅희 (인)

심 사 위 원 강미선 (인)

ABSTRACT

A Study on a Neural Network-Based Automated System for One-Dimensional Signal Data Processing

Kim, Yaeran

Major in Applied Artificial Intelligence

Dept. of Applied Artificial Intelligence

The Graduate School

Hansung University

One-dimensional (1D) signal data constitutes a fundamental data type across numerous scientific and industrial disciplines, including spectroscopy, chemical analysis, and biosignal monitoring. However, regardless of the specific domain, non-stationary baseline drift and background noise inherent to the measurement process significantly impede accurate quantitative interpretation. Existing techniques face limitations such as reliance on manual tuning or insufficient generalization performance. To address these challenges, this study proposes a generalized automated processing system. Furthermore, to empirically validate the proposed system, Raman spectral data, a representative form of 1D signals, is utilized to verify its effectiveness.

This paper presents two complementary approaches. First, Automated Parameter Optimization Network for Baseline Correction (APON-BC), proposed for computational efficiency, is a hybrid system that automatically predicts optimal algorithmic parameters using a neural network, enabling rapid processing in

resource-constrained environments. Experimental results demonstrate that APON-BC achieves exceptional efficiency and stability, reducing FID and ED by 84.71% and 64.65%, respectively, compared to existing techniques, with a remarkably low computational cost of 0.002 GFLOPs. Second, Baseline Correction with Transformer (BC-Former), proposed for precise reconstruction, is a Transformer-based end-to-end deep learning model. It effectively corrects complex nonlinear baselines by leveraging realistic synthetic data and a Dynamic Loss Weighting strategy. BC-Former exhibits unrivaled precision and adaptability to unseen data, achieving CS and PCC scores of 0.998 and minimizing FID and ED to 0.037 and 0.286, respectively.

Consequently, this study not only resolves the inherent trade-off between computational efficiency and reconstruction precision but also offers a comprehensive solution tailored to a wide spectrum of operational environments, ranging from resource-constrained embedded devices to high-performance computing scenarios. By demonstrating the methodological evolution from a hybrid parameter-prediction approach to a fully end-to-end deep learning framework, this research establishes a new standard for the robust automation and reliability of 1D signal data analysis.

[Keywords] One-Dimensional Signal Processing, Baseline Correction, Transformer, Parameter Optimization, Synthetic Data Generation, Dynamic Loss Weighting

Table of Contents

List of Tables

List of Figures

Ⅰ. Introduction

One-dimensional (1D) signal data, characterized by sequential measurements along a single axis (such as time, frequency, or chemical shift), constitutes a fundamental data type across numerous scientific and industrial disciplines[1-3]. This data is widely employed in diverse fields including spectroscopy (e.g., Raman and Fourier Transform Infrared Spectroscopy(FTIR)), chemical analysis (e.g., Chromatography), and biosignal monitoring (e.g., Electrocardiogram(ECG) and Electroencephalogram(EEG)). The reliable processing and interpretation of 1D signal data are crucial for extracting insights into molecular composition, chemical characteristics, or physiological states. However, regardless of the domain, real-world 1D signal data often contain undesired background artifacts and noise, such as baseline drift, which arise from instrumental factors, environmental variability, or sample heterogeneity. These unwanted background signals severely impede accurate quantitative analysis and reliable interpretation, making effective baseline correction a critical prerequisite for the trustworthy analysis of 1D signals across all application fields[4-6]. In the chemical domain, for instance, Raman spectroscopy is a powerful analytical technique that enables non-destructive analysis of molecular structures and chemical properties. It is widely employed in diverse fields such as materials science, life sciences, chemistry, and medical diagnostics[7-9]. Like other 1D signal methods, Raman spectral data are profoundly affected by background signals, primarily fluorescence, making effective baseline correction crucial for the reliable analysis of

Raman spectra [10−13]. While this paper focuses on developing and validating methodologies using Raman spectral data as a primary test case, the proposed techniques are designed to be generalizable and applicable to the broad category of 1D signal processing challenges.

To address this challenge, various approaches have been developed, ranging from traditional signal processing−based techniques to neural network−based models and recent Transformer−based architectures. Signal processing−based approaches, such as Adaptive Iteratively Reweighted Penalized Least Squares (AIRPLS)[14] and an Asymmetrically Reweighted Penalized Least Squares (ARPLS)[15], are widely used due to their stability. However, they heavily rely on manual parameter tuning, making the process labor−intensive and subjective, often leading to inconsistent results across large datasets. Neural network−based approaches, such as Convolutional Autoencoder for baseline correction (CAE+)[16] and Baseline Recognition Networks (BRN)[17], offer automated correction but often face structural limitations. Convolutional architectures with fixed receptive fields struggle to capture the long−range dependencies required to model global baseline trends. Furthermore, existing neural network−based models often suffer from poor generalization on unseen data due to limited training datasets. The Transformer[18]−based approaches, such as the method proposed by Zhao et al.[19], have recently been introduced to capture global context. However, current implementations typically rely on simplistic synthetic datasets, leading to over−correction of genuine peaks or failure to adapt to the complex artifacts found in real−world spectra. Consequently, despite these diverse advancements, a critical gap remains in achieving a fully automated, robust, and generalizable solution. Existing methods either suffer from the inefficiency of

manual tuning or face structural and data-driven limitations that hinder their performance in complex real-world scenarios.

To effectively bridge these gaps, this paper proposes two complementary baseline correction methodologies, each offering distinct advantages to address the trade-off between computational efficiency and correction precision. As the first approach, we address the need for an efficient and automated solution by proposing Automated Parameter Optimization Network for Baseline Correction (APON-BC). This approach aims to combine the stability of signal processing algorithms with the automation capabilities of neural networks, thereby eliminating manual intervention while maintaining computational efficiency. The proposed system systematizes the workflow by integrating three sequential stages, starting with constructing a synthetic dataset with a balanced labeling strategy to mitigate data imbalance, followed by utilizing a Convolutional Neural Network (CNN)[20]-based model to predict optimal parameters, and concluding with applying these predicted values to the target baseline correction algorithm for the final correction. By incorporating the balanced labeling strategy and CNN-based parameter prediction, the proposed system significantly improves baseline correction performance across diverse baseline types. Experimental results show that it outperformed the existing method, achieving significant improvements. Fréchet Inception Distance (FID)[21] and Euclidean Distance (ED)[22] were reduced by 84.71% and 64.65%, respectively, while Cosine Similarity (CS)[23] and Pearson Correlation Coefficient (PCC)[22] increased by 1.94% and 2.46%, respectively. Notably, it also demonstrates stable correction performance even in the presence of complex baseline shapes. Beyond accuracy, the computational efficiency of APON-BC was also

evaluated, demonstrating its suitability for real-time and resource-constrained applications.

As the second approach, we propose Baseline Correction with Transformer (BC-Former) to establish a comprehensive, end-to-end deep learning framework. While APON-BC successfully achieves automation and efficiency, its architecture acts as a parameter optimization system dependent on the target baseline correction algorithm. Structural characteristic of APON-BC suggests a promising avenue for further advancement by transitioning from an algorithm-dependent scheme to a fully end-to-end deep learning framework. By adopting this direct signal reconstruction approach, the system can transcend the fixed mathematical constraints of traditional algorithms and fully leverage the global modeling capabilities of deep learning. Specifically, BC-Former is designed as a Transformer-based model capable of capturing long-range dependencies in spectral data. To ensure robust generalization, we constructed a realistic synthetic dataset incorporating diverse baseline shapes and bottom artifacts, and implemented a Dynamic Loss Weighting (DLW) strategy to optimize multiple loss components during training. Experimental results demonstrate that BC-Former consistently outperforms conventional signal processing and neural network-based methods. Quantitative evaluations confirmed its superior reconstruction capability, with CS and PCC reaching 0.998, while FID and ED were minimized to 0.027 and 0.247, respectively. Notably, the model exhibits robust generalization capabilities, effectively correcting complex nonlinear baselines and subtle artifacts that traditional methods struggle to resolve. Furthermore, the effectiveness of the proposed dynamic adaptive system was also validated, confirming its strong adaptability and robustness when applied to unseen real-world Raman spectral

data.

Based on these two progressive approaches, the primary contributions of this paper are summarized as follows. First, this research resolves the critical trade-off between computational efficiency and correction precision by proposing two complementary methodologies. By developing the lightweight APON-BC system alongside the high-performance BC-Former model, we provide a dual-track solution that caters to a wide spectrum of operational environments, ranging from resource-constrained embedded devices requiring real-time processing to high-performance computing scenarios demanding state-of-the-art accuracy. Second, we present a methodological evolution from a hybrid, parameter-prediction approach to a fully end-to-end deep learning architecture. This progressive study identifies the structural limitations of algorithmic dependency in modular systems and successfully overcomes them through a unified Transformer-based framework, thereby offering a complete roadmap for advancing data-driven signal processing techniques. Third, we enhance the fundamental robustness and automation of spectral analysis by introducing advanced data engineering strategies. Through the implementation of balanced labeling strategies, realistic synthetic data generation, and DLW, both proposed approaches eliminate the subjectivity of manual tuning and ensure stable generalization even against complex, unseen baseline patterns. Finally, the effectiveness of this systematic approach is rigorously validated through comprehensive benchmarks. Extensive experiments demonstrate that both methods not only outperform conventional signal processing and neural network-based techniques in their respective domains but also establish new standards for automated and reliable baseline correction.

The remainder of this paper is organized as follows. Section 2 reviews related works and provides essential background knowledge regarding existing baseline correction techniques. Section 3 details the methodology of the first approach, APON-BC, covering the synthetic dataset generation process, the balanced labeling strategy, and the CNN-based model architecture. This section also presents the experimental results and provides the limitations inherent to this approach. Section 4 introduces the second approach, BC-Former, which is proposed to overcome the limitations identified in the previous section. It describes the design and implementation of the model, including the realistic synthetic data generation scheme and DLW strategy, followed by a comprehensive evaluation of its performance. Finally, Section 5 summarizes the overall research findings and outlines potential directions for future work.

Ⅱ. Related work

This section reviews existing baseline correction methodologies, structured into four main categories: (1) signal processing-based methods, (2) automated parameter determination-based methods, (3) neural network-based methods, and (4) Transformer-based methods. Finally, we analyze the research gaps within these studies to clarify the necessity and positioning of the proposed research.

2.1 Signal processing-based methods

Representative traditional methods for baseline correction rely on mathematical modeling to distinguish the background signal from spectral peaks. Early approaches, such as polynomial fitting[24,25] and Asymmetric Least Squares (ALS)[26], provided fundamental frameworks for baseline estimation but often struggled with complex baseline shapes or required rigid parameter settings. To address these limitations and improve adaptability, various methods have been developed.

Zhang et al.[14] proposed the AIRPLS method, which iteratively updates the weights of the sum of squared errors between the fitted baseline and the original signal. This method offers the advantage of relatively simple parameter tuning and is applicable to a wide range of Raman spectra. Building on this, Ye et al.[27] extended the ARPLS[15] method, which adjusts weights iteratively to distinguish baselines from peaks, and later proposed the improved asymmetrically reweighted penalized least squares (IARPLS), which enhances baseline

estimation accuracy by refining the weight update mechanism. Xu et al.[28] introduced the doubly reweighted penalized least squares (DRPLS) method, which combines two distinct weighting functions to more effectively estimate complex baselines. Other signal processing-based methods include adaptive smoothness parameter penalized least squares (ASPLS)[29], Goldindec[30], and iterative reweighted quantile regression using augmented Lagrangian optimization[31].

These methods are generally easy to implement and interpret. However, their performance tends to degrade significantly when applied to data with complex baseline patterns, and they often require meticulous parameter tuning to achieve optimal correction.

2.2 Automated parameter determination-based methods

To overcome the limitations of manual tuning inherent in traditional signal processing methods, various approaches have been proposed to automate the selection of optimal parameters. These methods typically rely on mathematical decision functions or statistical criteria to identify parameter values that minimize baseline estimation errors without user intervention.

Baek et al.[15] initially proposed the ARPLS method, and later, Park et al.[32] introduced a decision function to automatically determine the optimal smoothing parameter for ARPLS. Similarly, Zhang et al. introduced the Extended Range Penalized Least Squares (ERPLS) method[33], which aims to automatically determine the optimal value of the smoothing parameter lambda within the ASPLS framework[29]. By automating parameter selection without requiring user intervention, these methods improve efficiency in large-scale data processing and enhance consistency in analysis results.

However, both the decision function for ARPLS[32] and ERPLS[33] exhibit inherent limitations in their ability to adapt to diverse data characteristics and complex noise environments. Fixed mathematical models or single-parameter optimization strategies often fail to fully capture the variability and complexity of real-world spectral data, and frequently require additional adjustments when applied to new or unseen data conditions.

2.3 Neural network-based methods

As an alternative to traditional signal processing techniques that rely on explicit mathematical modeling and manual parameter tuning, deep learning-based approaches have emerged as powerful tools for spectral analysis. These data-driven models aim to automatically learn complex spectral features and baseline patterns directly from large datasets, thereby eliminating the need for human intervention.

Han et al.[16] developed a convolutional denoising autoencoder for noise reduction (CDAE) and CAE+, both of which demonstrated improved noise suppression and peak preservation compared to traditional signal processing techniques. Chen et al.[34] proposed a model combining ResNet[35] and UNet[36] architectures to automatically correct baseline drift in spectral data. Their approach was more user-friendly and outperformed conventional methods that require manual parameter tuning. Liu[17] introduced the BRN, which leverage adversarial networks and deep residual learning to achieve accurate baseline correction without the need for manual adjustments. These neural network-based methods are capable of handling complex baseline patterns more effectively than signal processing-based techniques and offer automated correction without human intervention.

However, they tend to overfit to the baseline types present in the

training data, thereby limiting their generalization performance when applied to spectra acquired under different measurement conditions or from unseen datasets. Moreover, most of these models rely on convolutional architectures with fixed receptive fields, which inherently lack the capacity to capture long-range dependencies, an essential capability for modeling complex and distributed baseline patterns in spectral data.

2.4 Transformer-based methods

Recently, the Transformer[18] architecture has gained attention for 1D data, such as spectral signals, due to its ability to capture long-range dependencies through a self-attention mechanism. Self-attention, the core of the Transformer, directly models relationships among all elements in a sequence, making it particularly effective for learning global context in sequential data. Leveraging this capability, several studies have explored Transformer-based models for baseline correction in Raman spectra.

Zhao et al.[19] applied a Transformer model to baseline estimation and reported superior performance compared to CNN-based and ResUNet-based[37] approaches. They constructed a training dataset using manually labeled baselines and applied CubicSpline interpolation[38] along with random noise for data augmentation. Jiao et al.[39] proposed a hybrid model combining CNN and Transformer encoders, incorporating residual dense blocks, multi-self-attention, and a feature fusion layer to capture multi-scale features during correction. Their method outperformed conventional signal processing techniques in both baseline correction and spectral reconstruction.

However, current Transformer-based approaches still face critical challenges. For instance, in studies utilizing semi-synthetic data

derived from real samples, the limited diversity of baselines can constrain the model's generalization performance when applied to spectra with unseen characteristics. Furthermore, these methods often exhibit limited flexibility in handling a wide range of spectral variations, tending to specialize in specific scale ranges or data types rather than offering a robust, universally applicable solution.

2.5 Research gaps and contributions

Despite the extensive development of baseline correction methodologies, critical research gaps remain across existing approaches. Signal processing-based methods, while stable and interpretable, are fundamentally limited by their reliance on manual parameter tuning, which renders them inefficient and inconsistent for large-scale analysis. Although automated parameter determination techniques have been proposed to mitigate this, they are still constrained by rigid mathematical assumptions that fail to adapt to complex or irregular baseline patterns.

Deep learning-based approaches emerged to enable data-driven automation, yet they face distinct structural and practical challenges. Most existing models rely on CNNs, which suffer from fixed receptive fields, limiting their ability to capture the global dependencies essential for accurate baseline modeling across the entire spectrum. Furthermore, while recent Transformer-based architectures offer global context awareness, current studies often rely on simplistic synthetic datasets. This results in poor generalization on real-world data and frequent over-correction of genuine peaks, as the models fail to distinguish between complex artifacts and actual spectral signals.

To bridge these gaps, this paper proposes a progressive, two-step

approach. To address the inefficiency of manual tuning in signal processing and the high computational cost of end-to-end deep learning, the first approach proposes APON-BC. This system addresses the need for a lightweight, automated solution by combining a parameter-predicting CNN with the target baseline correction algorithm. It effectively resolves the trade-off between operational efficiency and automation, providing a practical tool for resource-constrained environments.

While the first approach focused on efficiency, the second approach introduces BC-Former to fundamentally advance robustness and precision, addressing the generalization constraints of prior deep learning models. To achieve this, we incorporate a realistic synthetic data generation scheme and DLW strategy. By doing so, it fills the critical gap in handling complex, nonlinear baselines and subtle artifacts, achieving state-of-the-art precision and generalization capability on unseen real-world spectra.

This research distinguishes itself from prior works by establishing a dual-track framework that systematically resolves the limitations inherent in existing methodologies. Unlike traditional signal processing methods reliant on manual tuning or automated parameter determination techniques constrained by rigid mathematical rules, the proposed APON-BC introduces a hybrid, learning-based adaptation strategy. This design maintains the stability of established algorithms while leveraging the automation capabilities of neural networks. Furthermore, conventional neural network-based models are often limited by fixed receptive fields, whereas existing Transformer-based studies frequently fail to generalize due to simplistic data assumptions. In contrast, the proposed BC-Former establishes a robust end-to-end framework reinforced by realistic spectral

synthesis and DLW. By integrating these advanced data engineering strategies, our approach overcomes the structural and data-driven deficits of prior deep learning approaches, offering a comprehensive solution that ensures both high-precision reconstruction and adaptability to unseen real-world variations.

Ⅲ. APON-BC

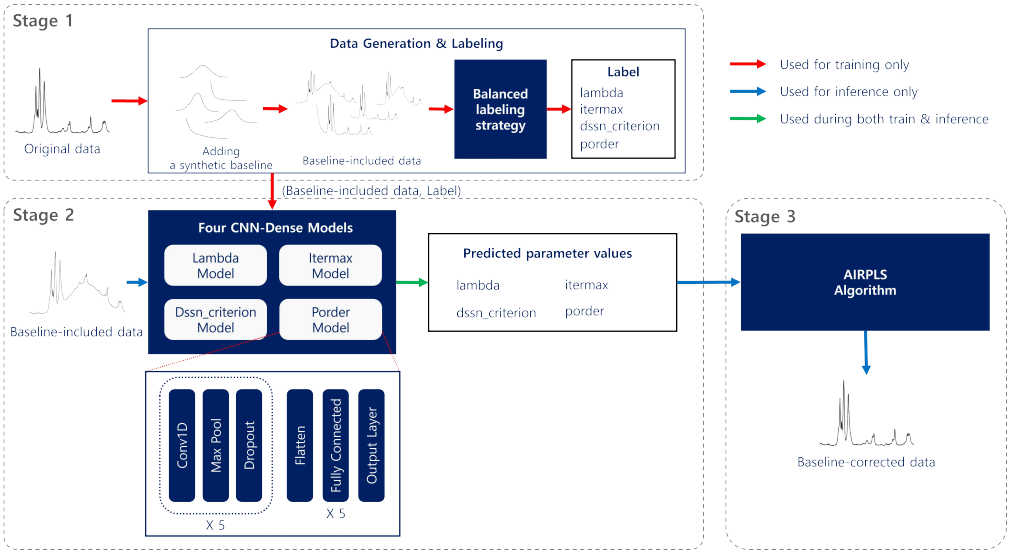
This section presents the first approach of this paper, Automated Parameter Optimization Network for Baseline Correction (APON-BC). APON-BC is a generalized framework designed to automate parameter selection for various mathematical correction algorithms, effectively addressing the inefficiency of manual tuning in traditional methods. In this approach, we implement and validate the system using the AIRPLS algorithm as a representative case study, owing to its widely recognized stability in Raman spectral analysis. While validated with AIRPLS, the framework is inherently adaptable to a broad range of other correction algorithms.

We begin by detailing the core methodology of the system. This includes the construction of the synthetic dataset, the labeling strategy, and the architecture of the parameter prediction model. Finally, the section concludes with a comprehensive evaluation of the system's performance and an analysis of its limitations.

3.1 Methodology

APON-BC is constructed as a sequential pipeline that integrates data engineering with deep learning-based inference. The overall process is divided into three distinct stages, ensuring a seamless transition from synthetic training data generation to the final algorithmic baseline correction.

3.1.1 Overall design of the APON-BC



[Figure 3-1] Overall architecture of the proposed APON-BC system

[Figure 3-1] provides a schematic overview of the proposed APON-BC system. As illustrated in the figure, the system consists of three main stages executed sequentially, progressing from training data generation to inference-based baseline correction. In Stage 1, synthetic baselines are added to original spectral data to create baseline-included samples. A labeling strategy is then applied to assign corresponding parameter values, which are used as ground truth labels for model training. In Stage 2, the baseline-included data serve as input, and each of the four parameter values obtained from Stage 1 is used as the target for training a separate CNN-Dense model. Each model is designed to predict one of the AIRPLS parameters: *lambda, itermax, dssn_criterion, porder*. Once training is complete, these models are applied to unseen baseline-included data to infer the optimal parameter values. In Stage 3, the predicted

values are passed to the AIRPLS algorithm for final baseline correction. This modular system enables automatic and data-adaptive baseline correction. Among the three stages, Stage 1 and Stage 2 are novel contributions of this approach and are described in detail in Sections 3.1.2 and 3.1.3, respectively

3.1.2 Dataset construction and labeling

Stage 1 of the proposed APON-BC system comprises two main components: the construction of baseline-included data and the labeling process. Constructing the baseline-added dataset with corresponding labels is essential, since ground-truth baseline-corrected spectra cannot be directly obtained from real measured data. Accordingly, we generate baseline-included data by adding a synthetic baseline to the original data and devise a labeling strategy for these datasets. By training on a diverse set of labeled baseline-included data, the system achieves robust and accurate baseline correction across various experimental conditions. To assign labels that reflect meaningful parameter configurations, we initially adopted a straightforward labeling strategy that selects the parameter set yielding the best correction performance for each baseline-included instance. However, due to certain limitations discussed in Section 3.1.2.2, this approach was ultimately not used in the final system. Instead, a balanced labeling strategy was developed to address these issues. For completeness, we first describe the initial labeling method in Section 3.1.2.2 (1), followed by the balanced approach in Section 3.1.2.2 (2). The detailed procedures for baseline-included data generation and the labeling strategy are described in the following subsections.

3.1.2.1 Generation of synthetic baseline-included data

This subsection describes the process of generating synthetic baseline-included data, including the motivation for using synthetic data and the specific functions used to simulate realistic baseline patterns.

Real-world spectral data inherently contains baseline drift. However, the true baseline-free spectra are unknown, making it impossible to obtain perfectly corrected ground-truth data for supervised training. Consequently, evaluating the model's generalization ability on real-world data becomes difficult due to the lack of a reliable reference for comparison. To overcome this limitation, we assume that certain clean spectral data are baseline-free and treat them as ground-truth targets. Synthetic baselines are then generated and added to these clean spectra to create the training inputs. These baselines are designed to mimic diverse patterns commonly observed in practice, such as Gaussian and exponential decay curves[40]. These types of baselines are commonly observed in real-world spectral data, and their abrupt signal variations and nonlinear characteristics present significant challenges for accurate correction. The model is trained to learn a mapping from the baseline-included spectra to their corresponding baseline-free counterparts. This approach allows the system to infer optimal correction parameters from baseline-included spectra, enabling robust correction on real data. The functions used to generate the synthetic baseline shapes are described below.

1) Gaussian function

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \quad (1)$$

A Gaussian baseline is generated using the Gaussian function defined in Equation (1). This function is characterized by the mean μ and variance σ^2, which determine the position and width of the resulting baseline curve.

2) Exponential function

$$f(x) = f_0 \exp(-\lambda x). \quad (2)$$

An exponential baseline is generated using the exponential decay function defined in Equation (2). This function is defined by the initial value f_0 and the decay rate λ, which control the starting intensity and the rate at which the baseline decreases.

Using the functions described above, synthetic baselines are added to the baseline-free spectra, resulting in a diverse dataset that incorporates a variety of baseline shapes.

3.1.2.2 Labeling strategy

To train the proposed neural network for automated parameter prediction, it is essential to assign optimal parameter values as ground truth labels to each synthetic baseline-included spectrum. This subsection details the labeling methodology employed to determine the most effective combination of AIRPLS parameters, including *lambda, itermax, dssn_criterion*, and *porder*. We first describe the initial labeling strategy, which selects parameters based solely on correction performance. Subsequently, we introduce the balanced labeling strategy, which was developed to address the data imbalance issues inherent in the initial approach and to ensure robust model generalization.

1) Initial labeling strategy

Although the performance of baseline correction is highly sensitive to parameter settings, the existing method typically uses a fixed *dssn_criterion* value of 1×10^{-3}, treating it as a constant rather than a tunable parameter. Other parameters such as *lambda, itermax* and *porder* are technically adjustable, but they are not dynamically optimized and

often remain fixed unless manually tuned by the user. This reliance on manual tuning limits the adaptability and flexibility of conventional approaches. In contrast, the proposed APON-BC system treats *dssn_criterion* as a tunable parameter and includes it in the optimization process along with *lambda, itermax* and *porder*. The system performs a systematic exploration of parameter combinations to identify the set that yields the best correction performance for each baseline-included instance. More specifically, to assign labels, AIRPLS is applied across all parameter combinations, and the one yielding the best correction is selected as the label. These labels are then used to train the model to predict optimal parameters for baseline correction.

The ranges of candidate parameter values used for this process are systematically defined to cover a broad spectrum of baseline characteristics. The smoothing factor (*lambda*) was selected from a logarithmic scale 10^i, where i ranges from 0 to 9. The maximum number of iterations (*itermax*) was configured to vary from 5 to 50 in increments of 5. The negative error tolerance threshold (*dssn_criterion*) consisted of five distinct values: 1×10^{-6},1×10^{-5},5×10^{-5},1×10^{-4} and 5×10^{-4}. Finally, the smoothing order (*porder*) was chosen from the integer set {1, 2, 3}.

All possible parameter combinations within the defined ranges are generated, and baseline correction is performed for each combination using the AIRPLS algorithm. For every corrected output, the ED[22] between the corrected result and the baseline-free original spectral data is computed. The parameter combination that yields the smallest ED is selected as the optimal set for the corresponding baseline-included instance. This optimal parameter set is then assigned as the label for that data instance. For example, if the combination *lambda* = 10^5, *itermax* = 30, *dssn_criterion* = 1×10^{-4}, and *porder* = 2 produces the best correction performance, this set is recorded as the label for the corresponding

baseline-included data instance, represented as the tuple (10^5, 30, 1×10^{-4}, 2). Through this process, each baseline-included input is paired with a unique set of four optimal parameter values, which serve as its training label.

2) Balanced labeling strategy

While the initial labeling strategy assigns optimal parameter values based solely on correction performance, it often results in data imbalance due to the overconcentration of specific parameter values. To address this issue, we introduce a balanced labeling strategy designed to balance the distribution of parameter values across the dataset and enhance the model's robustness. This strategy retains the core procedure of the initial labeling strategy, such as predefined ranges of parameter values and ED-based selection, while modifying the parameter selection loop to enforce balanced usage of each candidate value.

The initial labeling method, which selects parameters based on optimal correction performance, tends to produce imbalanced distributions of labeled parameter values. This imbalance arises because experimentally determined optimal parameters are frequently concentrated within a narrow range, while other values appear infrequently or not at all. For example, in the dataset generated by the initial labeling strategy, values such as 10^8 and 10^9 for *lambda* may be overrepresented, whereas other values are underrepresented. This skewed distribution negatively impacts the model during training and inference. Specifically, the model may overfit to the dominant parameter ranges, impairing its ability to generalize to unseen data. As a result, the imbalance undermines the system's ability to maintain consistent performance across diverse baseline correction scenarios.

Algorithm 1 Balanced labeling strategy

Initialization:

Define parameter name: *param*,
i.e. {*lambda, itermax, dssn_criterion, porder*}.

Define candidate values for other parameters (*Parameter Values*):
$values^{param}$

$values^{lambda} \leftarrow [10^i \ for \ i \in \{0,1,...,9\}]$

$values^{itermax} \leftarrow [5i \ for \ i \in \{1,2,...,10\}]$

$values^{dssn\_criterion} \leftarrow [1\times10^{-6},\ 1\times10^{-5},\ 5\times10^{-5},\ 1\times10^{-4},\ 5\times10^{-4}]$

$values^{porder} \leftarrow [1,2,3]$

Set per-parameter maximum label count:

$MAX_c^{param} \leftarrow total\_samples // length(values^{param})$

Initialize label counts: for each parameter, set
$label\_count^{param} \leftarrow [0,0,...,0]$
(where the length of the list is the length of the $values^{param}$)

Generation of baseline-included data:

Load the predefined baseline-free Raman spectrum: *original*

Select a random baseline type: Gaussian or Exponential

Apply selected baseline to the original Raman spectrum

Iteration over *Parameter Values*:

$L \leftarrow lambda$

$L\_values \leftarrow values^{lambda}$

$MAX_c \leftarrow MAX_c^{lambda}$

while *True* **do**

if *for every parameter type param and every index k*,
$label\_count^{param}[k] \geq MAX_c^{param}$ **then**

break

for *each value* $v \in L\_values$ **do**

for *each combination of* $values^{param}$ *in the Cartesian product of the remaining parameter sets* **do**

$correction \leftarrow$ Apply AIRPLS correction using the full set of parameters

$dist \leftarrow ||original - correction||$ // Euclidean distance

Store optimal parameter set(selected_combination) minimizing *dist*

$\forall\ (param, v) \in$ selected_combination: $label\_count^{param}[index\_of(v)] \mathrel{+}= 1$

Check limits:

for each value v in *L_values* **do**

if $label\_count[index\_of(v)] \geq MAX_c$ **then**

Remove v from *L_values*

if *L_values* is empty **then**

Update L to next priority parameter in order:

34: $lambda \rightarrow itermax \rightarrow dssn\_criterion \rightarrow porder$
35: Re-assign corresponding L_values
36: Also update $MAX_c \leftarrow MAX_c^L$
37: **Saving and exporting results:**
38: Save optimal parameter combinations
39: Export results

[Table 3-1] Algorithm of balanced labeling strategy

To address the issue of parameter imbalance, APON-BC proposes an effective mitigation approach using a balanced labeling strategy, detailed in [Table 3-1]. In lines 1-10, we first define the full set of parameters *param*, and, for each *param*, enumerate its candidate values $values^{param}$, which serve as the target for exploration. We then compute a per-parameter maximum label count MAX_c^{param}, which is defined as the integer division of the total number of samples by the number of candidate values for that parameter. For example, when generating samples for a total of 1,000 spectra and a given parameter has 10 candidates ($|values^{param}| = 10$), then MAX_c^{param} is 100. Then, the system initializes a zero-vector $label\_count^{param}$ of length $values^{param}$, which records how many times each specific value has been used.

After that, in lines 11-14, following the procedure introduced in Section 3.1.2.1, the system loads the baseline-free *original* data, randomly selects a synthetic baseline type, and generates the corresponding baseline-included spectral data. The procedure in lines 15-27 performs baseline correction using AIRPLS across various parameter combinations. First, we define a leading parameter L (initially *lambda*), and then assign its candidate values L_values and corresponding maximum label count MAX_c. The iteration continues until every count of candidate values for each parameter reaches its MAX_c^{param}, according to lines 20-21. As described in Section 3.1.2.2 (1), the candidate combination that yields the most similar correction result to the original spectrum

based on ED is selected as the optimal parameter set.

After selecting the best-performing combination, its corresponding $label\_count^{param}$ is updated by incrementing each chosen candidate value's label count. For example, if in iteration 5 the parameter *porder*=3 is selected, the corresponding entry in $label\_count^{porder}$ is incremented from [200, 105, 284] to [200, 105, 285]. Lines 28-36 manage the re-balancing logic based on $label\_count^{param}$. If the count for a value v in the L_values exceeds MAX_c, v is removed from L_values. If L_values becomes empty, the exploration target L is shifted to the next priority parameter (e.g., from *lambda* to *itermax*) and reset both L_values and MAX_c accordingly. Finally, as described in lines 37-39, the selected optimal parameter combinations are stored, and the results are saved as output files, completing the labeling process.

By ensuring a more balanced distribution of labeled parameter values across the dataset, this strategy mitigates bias toward overrepresented values. As a result, the trained model in the APON-BC system can more effectively predict appropriate parameter values across diverse baseline-included data, leading to improved and consistent baseline correction performance.

3.1.3 Design of the parameter prediction model

Stage 2 of the proposed APON-BC system involves designing and training a parameter prediction model that directly estimates optimal parameter values from input spectral data. This model enables automatic parameter tuning tailored to different baseline characteristics, thereby facilitating effective and adaptive baseline correction. A key advantage of APON-BC is its architectural flexibility; the output layer can be dynamically configured to match the specific parameter requirements of the target algorithm. Further details of the parameter prediction model are

presented in the following subsections.

Layer or Operation	In Channel	Out Channel	Kernel Size	Stride	Padding
1. Conv1D	1[1)]/64	64	8	1	same
2. MaxPooling1D	64	64	2	2	same
3. Dropout[2)]	64	64	-	-	-
Repeating layers 1 to 3 five times	-	-	-	-	-
Flatten	64	-	-	-	-
4. Fully Connected	Flattened	64	-	-	-
5. Fully Connected	64	32	-	-	-
6. Fully Connected	32	16	-	-	-
7. Fully Connected	16	8	-	-	-
8. Fully Connected	8	4	-	-	-
9. Fully Connected	4	1	-	-	-

[Table 3-2] Architecture of the parameter prediction model in the APON-BC system

3.1.3.1 Network architecture

CNNs are particularly effective for learning patterns in spectral data, as they can extract important features while preserving positional relationships within the signal. In this approach, the spectral data are 1D, representing intensity variations along the spectral axis. Accordingly, a 1D CNN architecture is adopted to predict optimal parameter values from the input spectra. The input data are provided as 1D arrays, with lengths that vary depending on the specific dataset. For network input, the data are reshaped into a three-dimensional tensor of shape (batch size, 1, data length). Each of the AIRPLS parameters, *lambda, itermax, dssn_criterion* and *porder*, has distinct characteristics and interacts differently with the spectral data. Due to these parameter-specific properties, the APON-BC

1) Only the first Conv1D layer

2) dropout rate = 0.3

system employs a separate CNN-based model for each parameter. While all models use the same network architecture, training them independently for each parameter enables the networks to learn parameter-specific features without interference from other prediction targets, thereby improving prediction accuracy.

The architecture of the parameter prediction network is summarized in [Table 3-2]. It consists of five 1D convolutional layers followed by fully connected layers. Each convolutional layer has 64 filters, a kernel size of 8, and a stride of 1. To prevent overfitting, each convolutional layer is followed by a Max Pooling layer (pool size = 2) and a Dropout layer (dropout rate = 0.3). The features extracted from the convolutional layers are flattened and passed through fully connected layers with 64, 32, 16, 8, and 4 nodes, sequentially, before producing the final parameter prediction. This architecture allows the model to effectively learn and predict optimal parameter values for robust baseline correction.

3.1.3.2 Loss function

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2. \quad (3)$$

Since the proposed model is designed for optimal parameter prediction, a regression-based loss function is adopted. Mean Squared Error (MSE) is used as the training loss to penalize large deviations between predicted and true parameter values. MSE is calculated as the average of the squared differences between predicted values $\hat{y_i}$ and actual values y_i over n data points, as shown in Equation (3). In the context of baseline correction parameter estimation, it is important to penalize larger prediction errors more heavily, as even small inaccuracies in parameter tuning can lead to significant distortions in the corrected spectrum. Because MSE amplifies larger errors through squaring, it is well-suited as a loss function for this task.

3.2 Performance evaluation

In this section, we evaluate the performance of the proposed APON-BC system. To validate the effectiveness of the proposed system, we implemented the system using AIRPLS as a representative instance and compared this automated system against existing methods. We first present the experimental environment and dataset configuration, followed by the evaluation methodology based on similarity metrics. Next, we investigate the impact of data imbalance on correction performance and compare the proposed system with conventional approaches to validate its improvements in baseline correction. Finally, we assess the computational efficiency of the proposed system relative to other neural network-based baseline correction models, confirming its suitability for resource-constrained environments.

3.2.1 Implementation details

The experiments were conducted on a server equipped with four RTX A6000 GPUs, an AMD Ryzen CPU (7.01 GHz), and 46.07 GB of RAM, running Ubuntu 20.04 LTS with Linux kernel version 5.15.

The software environment was based on Python 3.8.18, and the neural network models were implemented using TensorFlow[41] 2.12.0. Data processing for model training and inference was performed using Python's standard libraries, along with NumPy[42] and Pandas[43].

The original spectral data were obtained from simulated Raman measurements of the chemical compound 2-Chloroethyl ethyl sulfide, using a 532 nm excitation laser. These spectra served as clean, baseline-free references to which synthetic baselines were added for dataset construction. The resulting dataset was synthetically constructed to support effective model training and evaluation. It was divided into a

training set of 1,076 samples, a validation set of 358 samples, and a test set of 350 samples, enabling a structured and rigorous evaluation of model performance. To ensure numerical stability and consistent scaling during training, the input spectral data were min−max normalized to the range [0, 1]. Likewise, the label data were each normalized using min−max scaling based on their respective predefined value ranges. After training, the predicted labels were inverse−transformed back to their original scales and used as the final parameter values for baseline correction. The training settings were as follows: the learning rate was set to 0.0001, and the batch size to 16. Training was performed for up to 10,000 epochs using the Adam[44] optimizer. To stabilize the training process, an exponential moving average (EMA) with a momentum of 0.99 was applied.

We employed four metrics for performance evaluation: FID[21], PCC[22], ED[22], and CS[23]. In this experiments, FID was used to assess the correction performance by measuring the similarity between the distribution of the original spectral data and the corrected results. Lower FID values indicate greater similarity between the two distributions. PCC measures the linear correlation between two variables. Values close to 1 indicate a strong positive correlation, while values near −1 indicate a strong negative correlation[22]. ED quantifies the absolute distance between two vectors, with smaller values indicating higher similarity. Similarly, CS evaluates the directional similarity between two vectors, ranging from −1 to 1, where values approaching 1 represent a high degree of alignment in vector direction[23]. These four metrics are jointly used to comprehensively evaluate the performance of the APON−BC system from multiple perspectives.

3.2.2 Performance evaluation results

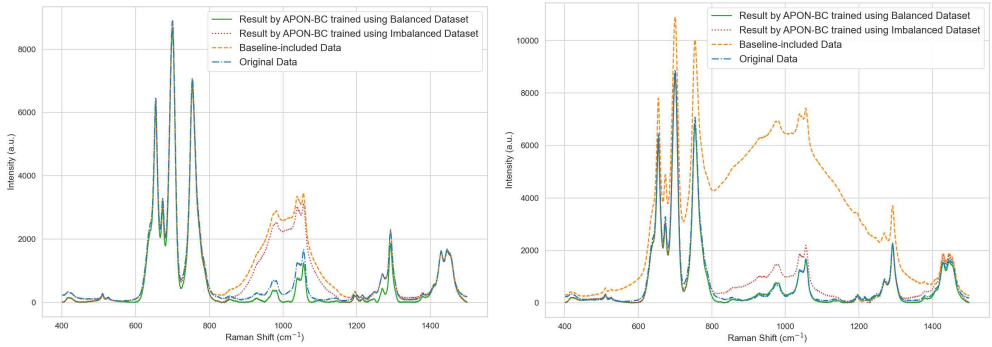
We conducted three comparative experiments to evaluate the effectiveness and practicality of the proposed system. The first experiment investigates the impact of data imbalance on correction performance by comparing models trained on imbalanced versus balanced datasets. The second experiment compares the correction performance of the proposed APON-BC system with that of an existing baseline correction method. Finally, the third experiment analyzes the computational requirements of the proposed system in comparison with other neural network-based baseline correction models.

3.2.2.1 Effect of data imbalance on correction performance

As discussed in Section 3.1.2.2, the initial labeling strategy resulted in a disproportionate representation of certain parameter values. To address this issue, a balanced labeling strategy was introduced to construct a more balanced dataset. The effectiveness of this approach was evaluated by comparing models trained on balanced versus imbalanced datasets.

[Figure 3-2] and [Figure 3-3] show the correction results for baseline-included data generated using Gaussian and exponential distributions, respectively. In [Figure 3-2] (a), which illustrates data with a Gaussian baseline, the APON-BC model trained on the balanced dataset produced more accurate baseline estimates, particularly in the 800-1200 cm^{-1} region where strong spectral peaks were present. In contrast, the model trained on the imbalanced dataset tended to overestimate the baseline in this region. Additionally, as shown in [Figure 3-2] (b), the model trained on the imbalanced dataset demonstrated generally acceptable baseline removal across the broad 400-1400 cm^{-1} range but failed to properly correct the baseline in the critical 800-1200 cm^{-1}

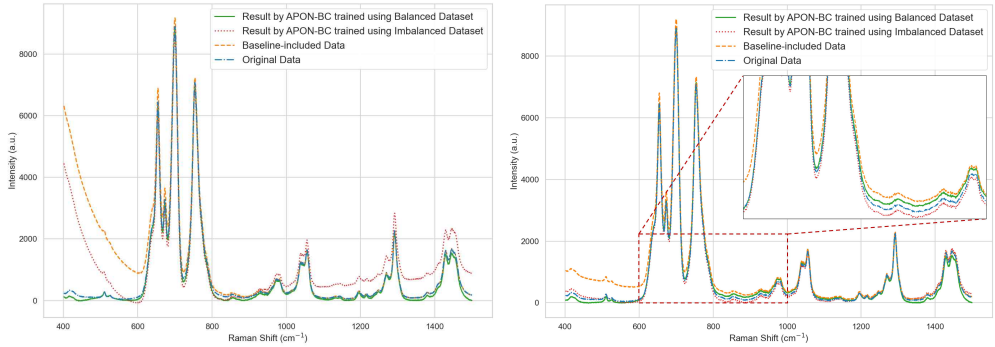
region. In comparison, the model trained on the balanced dataset achieved more accurate correction in this region and exhibited more consistent overall performance.



(a) Gaussian-distributed baseline-included data 1

(b) Gaussian-distributed baseline-included data 2

[Figure 3-2] Comparison of correction performance between APON-BC models trained on balanced and imbalanced datasets using Gaussian-distributed baseline-included data



(a) Exponential-distributed baseline-included data 1

(b) Exponential-distributed baseline-included data 1

[Figure 3-3] Comparison of correction performance between APON-BC models trained on balanced and imbalanced datasets using exponential-distributed baseline-included data

Similar results were observed in [Figure 3-3], which presents baseline-included data with an exponential distribution. In the 400-600 cm^{-1} range of [Figure 3-3] (a), where baseline variations are more abrupt, the model trained on the balanced dataset exhibited more stable correction performance. In contrast, the model trained on the imbalanced dataset tended to under-correct the baseline in this region. In [Figure 3-3] (b), both models demonstrated similarly strong correction performance across most of the spectrum. However, as observed in the magnified region, the model trained on the imbalanced dataset showed a slight reduction in peak intensity in the 600-1000 cm^{-1} region, whereas the model trained on the balanced dataset maintained a more stable and accurate correction. These results suggest that training with a balanced dataset helps mitigate localized degradation in correction performance and contributes to more reliable baseline estimation across various spectral regions.

Dataset types	PCC(↑)	CS(↑)	FID(↓)	ED(↓)
Imbalanced	0.940	0.952	1.35×10^{8}	2.22×10^{4}
Balanced	0.998	0.998	1.04×10^{7}	6.01×10^{3}

[Table 3-3] Comparison of experimental results between models trained on imbalanced and balanced datasets

In both cases, the APON-BC model trained on a balanced dataset preserved the characteristics of the original spectral data more effectively while achieving more accurate baseline removal. This observation is further supported by the quantitative results presented in [Table 3-3]. The table shows that the model trained on the balanced dataset achieved superior correction performance compared to the model trained on the imbalanced dataset. Specifically, PCC and CS values increased, indicating stronger linear correlation and greater directional similarity between the

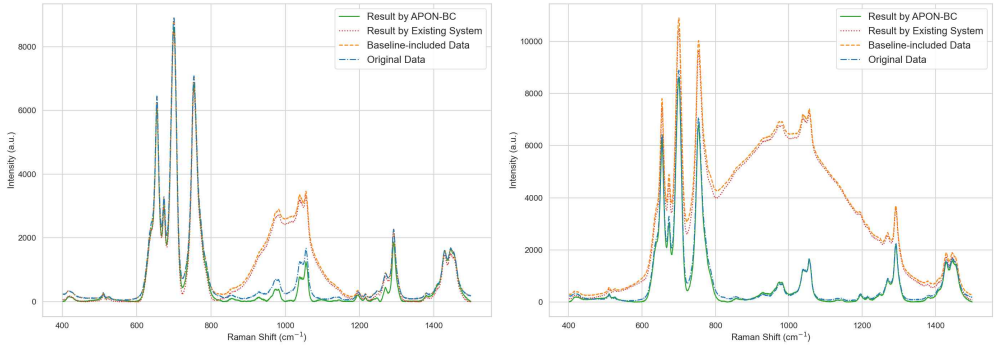
corrected and original data. Additionally, both FID and ED values decreased, reflecting improved similarity and reduced deviation from the baseline-free spectra. These results suggest that imbalanced datasets can lead the model to overfit specific parameter ranges, thereby degrading correction performance. In contrast, training on a balanced dataset mitigates overfitting and enables more consistent and reliable correction across various baseline types.

3.2.2.2 Performance comparison between the proposed and existing systems

We evaluated the baseline correction performance of both the existing and proposed systems. [Figure 3-4] and [Figure 3-5] show the correction results for baseline-included data generated using Gaussian and exponential distributions, respectively. In [Figure 3-4] (a), for the Gaussian-distributed baseline, the proposed system effectively removed the baseline while preserving the shape of the original signal. As a result, the corrected spectrum more closely resembled the original data, particularly in the 800-1200 cm^{-1} region. Similarly, in [Figure 3-4] (b), the proposed system achieved precise baseline estimation in the 600-1400 cm^{-1} range, successfully preserving complex peak structures. In contrast, the existing system exhibited insufficient baseline correction in both cases, causing signal distortion in [Figure 3-4] (a) and inaccurate baseline estimation in the complex peak region shown in [Figure 3-4] (b).

Similarly, in [Figure 3-5] (a), which presents baseline-included data following an exponential distribution, the proposed system achieved stable and accurate baseline correction, particularly in the 400-600 cm^{-1} region where sharp baseline variations occur. In [Figure 3-5] (b), the proposed system consistently preserved the characteristics of the original signal

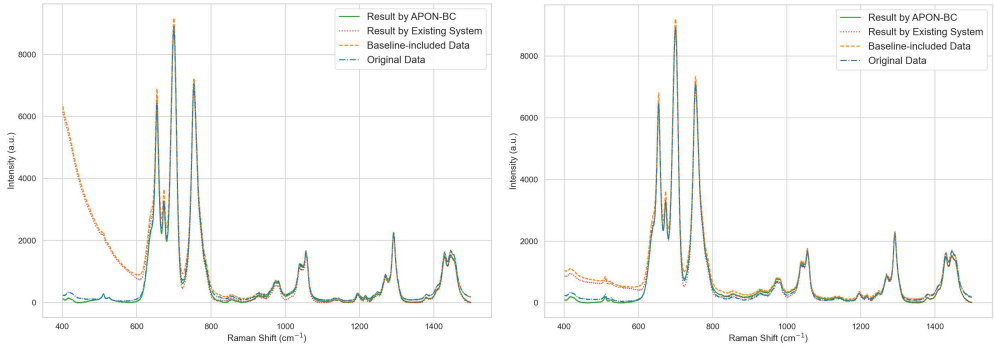
while effectively removing the baseline, demonstrating superior correction performance. In contrast, the existing system exhibited unstable baseline estimation in [Figure 3-5] (a) and showed a significant drop in correction accuracy in [Figure 3-5] (b), especially in regions with abrupt baseline changes. These results confirm that the proposed system consistently outperforms the existing system in baseline correction across various baseline types.



(a) Gaussian-distributed baseline-included data 1

(b) Gaussian-distributed baseline-included data 2

[Figure 3-4] Comparison of correction performance between existing and proposed systems for Gaussian-distributed baseline-included data



(a) Exponential-distributed baseline-included data 1

(b) Exponential-distributed baseline-included data 2

[Figure 3-5] Comparison of correction performance between existing and proposed systems for exponential-distributed baseline-included data

Method	PCC(↑)	CS(↑)	FID(↓)	ED(↓)
Existing system	0.974	0.979	6.80×10^{7}	1.70×10^{4}
APON-BC	0.998	0.998	1.04×10^{7}	6.01×10^{3}

[Table 3-4] Comparison of experimental results between the proposed and existing systems

[Table 3-4] presents a quantitative comparison of the correction performance between the existing and proposed systems. As shown in the table, the proposed system achieved improved results across all evaluation metrics. Specifically, it produced significantly lower FID and ED values, and higher PCC and CS values, indicating better similarity to the original signal. These results provide strong quantitative evidence that the proposed system more effectively removes the baseline while preserving the characteristics of the original spectral data.

3.2.2.3 Comparison of computational requirements between the proposed method and other neural network-based baseline correction models

In this section, we quantitatively compare the computational requirements of the proposed APON-BC system with those of representative neural network-based baseline correction models. Specifically, the comparison is based on three metrics: floating point operations (FLOPs), the total number of parameters (Params), and model size. The compared models include BRN[17], CAE+[16], one-dimensional Transformer (1dTrans)[19], and the proposed APON-BC. For APON-BC, we report two configurations: one using a single parameter prediction network, and another using the full set of four independent networks.

Model	FLOPs(GFLOPs)	Params	Model Size(MB)
APON-BC (1 model)	0.002	1,162,337	4.43
APON-BC (4 models)	0.008	4,649,348	17.72
BRN	0.057	1,797,050	6.86
CAE+	27.563	2,842,449	10.84
1dTrans	56.664	127,009,793	484.56

[Table 3-5] Comparison of computational requirements between APON-BC and other neural network-based baseline correction models

As shown in [Table 3-5], the BRN and CAE+ models require relatively high FLOPs and parameter counts, while the 1dTrans model exceeds 100 million parameters and occupies more than 480 MB of storage. Although BRN has a relatively small number of parameters and a compact model size, it processes the input spectrum by splitting it into windows of 100 data points and passing each through the network multiple times. Additionally, ten generator networks are trained and used independently, substantially increasing the total FLOPs despite the lightweight architecture of each individual network. These characteristics make such models less suitable for real-time applications or deployment on resource-constrained devices. In contrast, the APON-BC achieves the lowest FLOPs and model size in its single-network configuration, and maintains high computational efficiency even when all four networks are used concurrently.

In conclusion, APON-BC delivers competitive baseline correction performance while offering clear advantages in model compactness and computational efficiency. These properties make it particularly suitable for industrial environments with limited resources, portable analytical devices, and embedded systems. Its ability to combine both accuracy and efficiency distinguishes APON-BC from existing neural network-based approaches.

Ⅳ. BC-Former

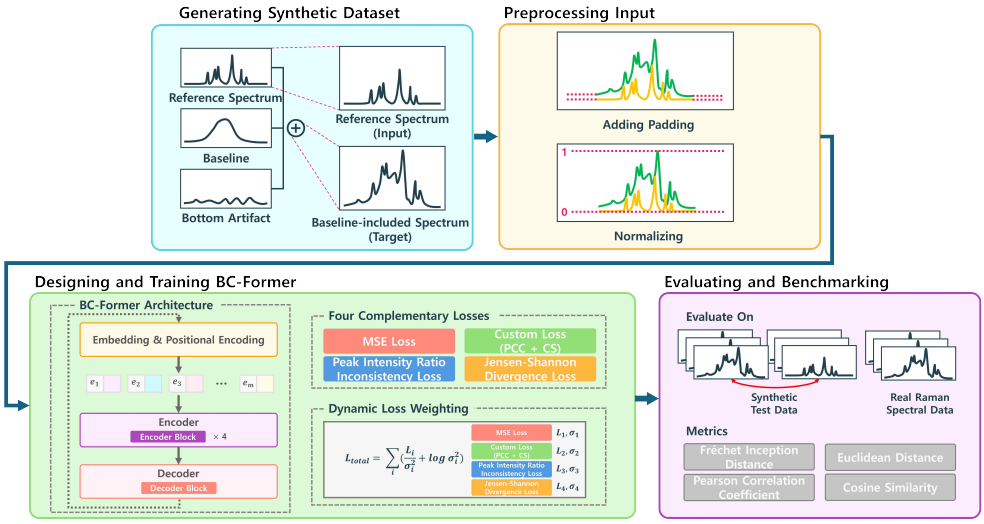
This section introduces the second approach of this paper, Baseline Correction with Transformer (BC-Former). While the previously proposed APON-BC successfully established an efficient automated system through parameter prediction, BC-Former aims to further advance the baseline correction capabilities by transitioning to a fully end-to-end deep learning framework. This structural evolution allows the system to transcend the fixed mathematical constraints of traditional algorithms and directly learn the optimal signal reconstruction mapping from data. By adopting this direct approach, BC-Former is specifically designed to capture long-range dependencies in Raman spectra. This design capability allows it to comprehensively address the generalization challenges faced by existing deep learning models, thereby enabling precise correction of complex nonlinear baselines and subtle artifacts. The following subsections will describe the methodology in detail, including the realistic synthetic dataset construction, the model architecture, and the DLW strategy, followed by a comprehensive evaluation demonstrating its superior performance and adaptability compared to conventional methods.

4.1 Methodology

In this section, we present the overall architecture and training procedure of BC-Former. We describe the end-to-end processing

pipeline and the generation of synthetic Raman data for robust training, followed by the preprocessing steps for training stability. We then introduce the model architecture and training strategy, which incorporate multiple loss functions with dynamic loss optimization.

4.1.1 Overall design of the BC-Former

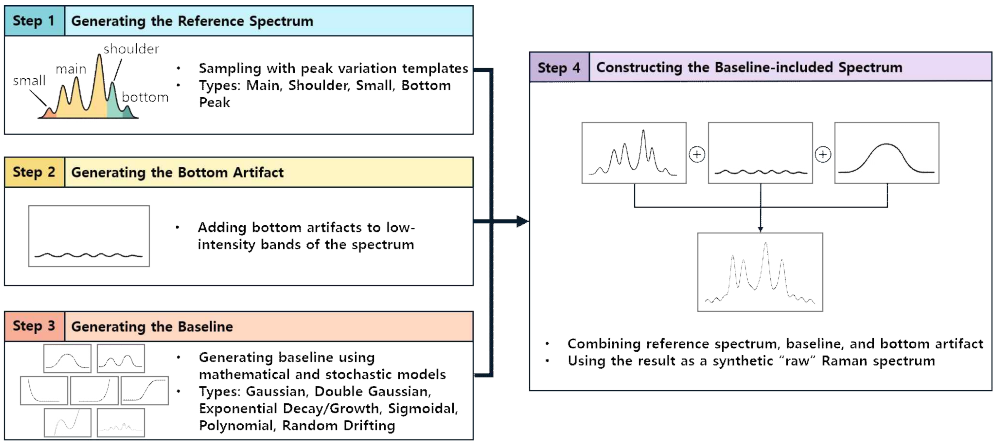


[Figure 4-1] Overall pipeline for developing and evaluating the BC-Former model

[Figure 4-1] provides a schematic overview of the procedure for building and evaluating the BC-Former model. To train the model, we first generate synthetic spectra that include a variety of baselines and bottom artifacts, effectively simulating the characteristics of real Raman data. These spectra are then normalized and padded to ensure input consistency. The preprocessed inputs are passed through BC-Former, which employs a Transformer-based encoder and decoder to generate baseline-corrected outputs. To enhance generalization across diverse baseline types, the model is trained

using multiple loss functions with complementary characteristics, whose contributions are dynamically adjusted through a dedicated weighting strategy. Finally, the trained model is evaluated on not only synthetic spectra but also real Raman data, with performance assessed using various quantitative metrics. Each of these aspects is described in detail in the following sections.

4.1.2 Synthetic Dataset Generation



[Figure 4-2] Overview of synthetic dataset generation

This section describes the overall structure and generation procedure of the synthetic dataset used to train the BC-Former model. [Figure 4-2] illustrates the overall data generation process, which consists of four main steps. First, a reference spectrum simulating the shape of an ideal Raman spectrum is generated (Step 1), followed by the creation of bottom artifacts that reflect low-intensity noise patterns observed in real data (Step 2). Next, various types of baseline patterns are produced (Step 3), and the three components are then combined to construct a baseline-included spectrum (Step 4). The resulting synthetic spectra are designed to

cover a wide range of peak shapes and baseline conditions, thereby enhancing the generalization capability of BC-Former. Section 4.1.2.1 explains how the composition of our dataset differs from those used in related studies. Also, Section 4.1.2.2 provides a detailed description of the generation methods for each component and the overall synthesis process.

4.1.2.1 Limitations of existing synthetic datasets and our improvements

In related studies on baseline correction in Raman spectroscopy, synthetic datasets were often constructed by using reference spectra with monotonous peak structures or by incorporating only limited types of baseline distortions. For instance, Xu et al.[45] and Gebrekidan et al.[46] generated reference spectra by linearly combining randomly placed Gaussian peaks. While this approach offers simplicity and controllability, it fails to capture the complex and non-ideal peak patterns commonly observed in real Raman spectra. Additionally, some studies applied only restricted forms of baseline distortions[16], limiting the datasets' ability to represent the diversity of baseline variations found in practice.

In contrast, this approach constructs a more generalized synthetic dataset by addressing these limitations. We generate reference spectra with peak profiles that more closely resemble those in real Raman data, including varied peak positions and intensities. Furthermore, our dataset incorporates diverse combinations of baseline patterns to reflect the variability observed across real-world spectra. We also introduce bottom artifacts commonly found in raw measured spectra to further enhance realism. This approach not only alleviates the challenges of collecting large-scale real Raman data but also enables robust training on a broad and diverse dataset, allowing BC-Former to effectively learn the wide range of spectral variations encountered in real-world applications.

4.1.2.2 Synthetic Dataset Components and Generation Procedures

The synthetic dataset constructed in this approach consists of three main components: (1) a reference spectrum, (2) a baseline, and (3) a bottom artifact. The remainder of this section describes the procedures used to generate each component.

1) Design and Construction of Synthetic Reference Spectrum

The reference spectrum represents an ideal Raman spectrum without any baseline and serves as the ground truth for baseline correction during model training. To reflect the complexity observed in real Raman spectra, the reference spectrum is constructed to include four types of peaks: main peaks, shoulder peaks, small peaks, and bottom peaks. This categorization is established based on experimental observations of typical Raman spectral characteristics. The goal is to systematically incorporate the intensity and structural features observed in real spectra into the reference spectrum. Each peak is modeled using a Gaussian function, and detailed descriptions along with corresponding equations for each type are provided below.

Main peaks constitute the primary spectral signals and represent the dominant vibrational modes commonly observed in Raman spectra[47]. These peaks are modeled as shown in Equation (4).

$$S_{main}(x) = \sum_{i=1}^{N_{main}} I_{main,i} \cdot \exp\left(-\frac{(x-\mu_{main,i})^2}{2\sigma_{main}^2}\right). \quad (4)$$

In this equation, $S_{main}(x)$ denotes the signal contribution of each main peak at position x, and the overall spectrum intensity is obtained by summing these contributions. N_{main} represents the number of main peaks. $I_{main,i}$ and $\mu_{main,i}$ denote the intensity and position of the i-th main peak, respectively, and σ_{main} controls the peak width.

Shoulder peaks are located near the main peaks and exhibit relatively lower intensity. In Raman spectroscopy, these peaks are not merely random measurement noise but often arise from the structural characteristics of the material[48]. Equation (5) models the shoulder peaks as $S_{sh}(x)$, which represents the signal contribution of each shoulder peak at a given position x, and the overall spectrum intensity is obtained by summing these contributions.

$$S_{sh}(x) = \sum_{i=1}^{N_{sh}} I_{sh,i} \cdot \exp\left(-\frac{(x-\mu_{sh,i})^2}{2\sigma_{sh}^2}\right). \quad (5)$$

N_{sh} denotes the number of shoulder peaks, and $I_{sh,i}$ and $\mu_{sh,i}$ represent the intensity and center position of the i-th shoulder peak, respectively. The parameter σ_{sh} controls the width of the shoulder peaks. Each $I_{sh,i}$ and $\mu_{sh,i}$ is randomly sampled from a predefined range near $I_{main,i}$ and $\mu_{main,i}$, respectively, reflecting the assumption that shoulder peaks occur adjacent to main peaks with lower intensity.

Small peaks are minor spectral features with varying widths and intensities, added to increase spectral diversity. As described in Equation (6), $S_{small}(x)$ denotes the signal contribution of each small peak at a specific position x, and the total spectrum intensity is obtained by summing these contributions.

$$S_{small}(x) = \sum_{i=1}^{N_{small}} I_{small,i} \cdot \exp\left(-\frac{(x-\mu_{small,i})^2}{2\sigma_{small}^2}\right). \quad (6)$$

The number of small peaks is determined by N_{small}, and the intensity of each peak $I_{small,i}$ is constrained by the maximum value of the main peaks to minimize its influence on the overall spectral shape. The position and width of each small peak are defined by $\mu_{small,i}$ and $\sigma_{small,i}$, respectively.

Bottom peaks are low-intensity spectral features that occur in regions of low overall signal and are designed to capture the fine complexity of

the spectral profile. In this approach, they are generated at randomly selected positions with low amplitudes. As described in Equation (7), $S_{bottom}(x)$ denotes the signal contribution of each bottom peak at position x, and the overall spectrum intensity is obtained by summing these values.

$$S_{bottom}(x) = \sum_{i=1}^{N_{bottom}} I_{bottom,i} \cdot \exp\left(-\frac{(x-\mu_{bottom,i})^2}{2\sigma_{bottom}^2}\right). \quad (7)$$

The number of bottom peaks is specified by N_{bottom}, and the intensity $I_{bottom,i}$ is set to reproduce the subtle fluctuations typically observed in spectral baselines. The position of each bottom peak is determined by $\mu_{bottom,i}$, and generation is restricted to regions where the spectral intensity $S(x)$ is less than 1. The width of the bottom peaks is controlled by σ_{bottom}.

Peak Type	Number N	Intensity I	Width σ	Position μ
Main	10	$\sim u(0,10000)$	$0.002 \bullet len_{data}$	$\sim u(0,len_{data})$
Shoulder	10	$0.6 \bullet I_{main,i}$	$0.006 \bullet len_{data}$	$\mu_{main,i}+\delta_i$, $(\delta_i \sim u(-0.02 \bullet len_{data},0.02 \bullet len_{data}))$
Small	$\sim u(10,20)$	$\sim u(0.1 \bullet I_{\max},0.15 \bullet I_{\max})$	$\sim u(0.001 \bullet len_{data}, 0.15 \bullet len_{data})$	$\sim u(0,len_{data})$
Bottom	5	$\sim u(0.01 \bullet I_{\max},0.02 \bullet I_{\max})$	$0.001 \bullet len_{data}$	$\sim u(indices\ where\ S(x)<1)$

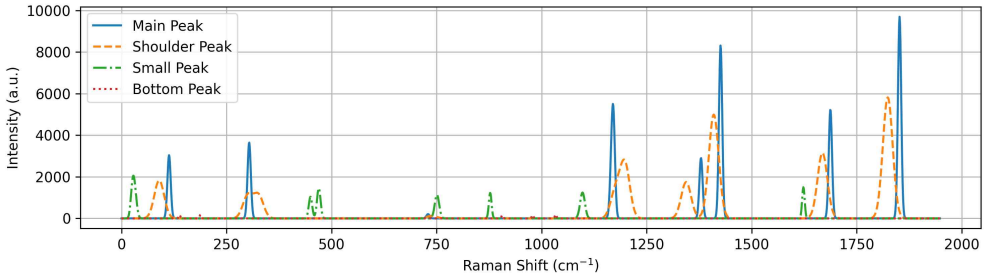
[Table 4-1] Parameter configurations used in synthetic spectrum generation

[Table 4-1] summarizes the types of peaks used to construct the synthetic reference spectrum, along with their generation parameters, including the peak parameters (number N, intensity I, width σ, and position μ). Since peaks are sequentially added to the spectrum in the order of main, shoulder, small, and bottom peaks, the characteristics of each peak type are determined based on the previously accumulated spectral data. Here, $I_{\max}$ denotes the maximum intensity of the spectrum constructed up to the current step, and it serves as the criterion for defining the intensity ranges of small and bottom peaks. In addition, δ_i is an offset used to position each shoulder peak relative to its corresponding main peak, and is sampled uniformly within a predefined range. For bottom peaks, positions are selected from regions where the spectrum intensity is below 1, thereby simulating characteristics near the baseline. len_{data} denotes the total length of the spectrum.

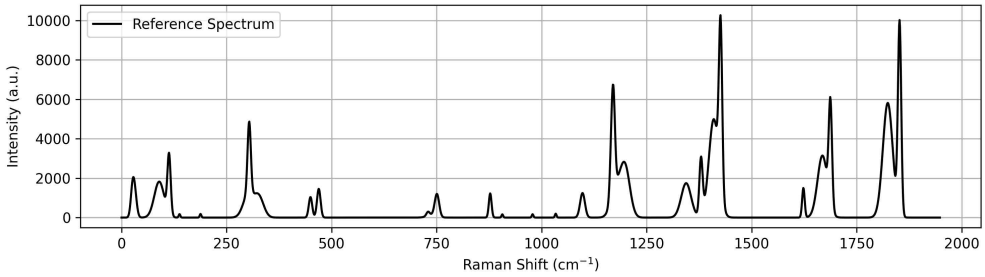
In conclusion, the final reference spectrum $S_{ref}(x)$, obtained by summing the four types of peaks described above, is defined as shown in Equation (8).

$$S_{ref}(x) = S_{main}(x) + S_{sh}(x) + S_{small}(x) + S_{bottom}(x), \quad \forall x \in domain. \tag{8}$$

[Figure 4-3] illustrates four peak types, including main, shoulder, small, and bottom, as presented in [Figure 4-3] (a), and the resulting reference spectrum generated by combining these peaks, as presented in [Figure 4-3] (b). As shown, the systematic design and integration of these diverse peak types enable the construction of a synthetic reference spectrum that effectively captures the complexity of real Raman spectra.



(a) Individual peak components



(b) Final reference spectrum

[Figure 4-3] Representative examples of synthetic reference spectra

2) Design and Construction of Synthetic Baselines

To capture the diverse baseline variations observed in real Raman spectral data, we design multiple types of baselines. This section describes the baseline types and the methods used to generate them.

A Gaussian distribution-based baseline is modeled using a single Gaussian function, which represents a smoothly varying background across the entire spectral range. The corresponding equation is provided in Equation (9).

$$B_{gau}(x) = A_{gau} \cdot \exp\left(-\frac{(x-\mu_{gau})^2}{2\sigma_{gau}^2}\right), \tag{9}$$

where A_{gau} is the amplitude, μ_{gau} is the center position, and σ_{gau} determines the width of the Gaussian curve.

A double Gaussian distribution-based baseline is constructed by

combining two Gaussian functions to model more complex background variations. By adjusting the center positions and widths of the two components, a wide range of baseline shapes can be represented. The corresponding equation is provided in Equation (10).

$$B_{dgau}(x) = A_{1,dgau} \cdot \exp\left(-\frac{(x-\mu_{1,dgau})^2}{2\sigma_{1,dgau}^2}\right) + A_{2,dgau} \cdot \exp\left(-\frac{(x-\mu_{2,dgau})^2}{2\sigma_{2,dgau}^2}\right), \quad (10)$$

where $A_{1,dgau}$ and $A_{2,dgau}$ are the amplitudes, $\mu_{1,dgau}$ and $\mu_{2,dgau}$ are the center positions, and $\sigma_{1,dgau}$ and $\sigma_{2,dgau}$ determine the widths of the respective Gaussian components.

Exponential decay and exponential growth distribution-based baselines utilize exponential functions to model asymmetric baseline variations. These functions represent patterns that gradually decrease or increase, depending on the value of the exponential rate. The decaying exponential baseline is defined in Equation (11).

$$B_{dexp}(x) = A_{dexp} \cdot e^{-\lambda_{dexp}x}, \quad (11)$$

where A_{dexp} is the amplitude, and λ_{dexp} is the decay rate controlling the decrease of the baseline. Similarly, the growing exponential baseline is defined in Equation (12).

$$B_{gexp}(x) = A_{gexp} \cdot e^{\lambda_{gexp}(x-N)}, \quad (12)$$

where A_{gexp} is the amplitude, λ_{gexp} is the growth rate, and N is the length of the data used to shift the function's offset.

A sigmoidal distribution-based baseline utilizes a sigmoid function to model a smoothly increasing or decreasing background trend. By adjusting the center position and slope parameter, the location and steepness of the transition can be effectively controlled. The corresponding equation is given in Equation (13).

$$B_{sig}(x) = A_{sig} \cdot \left(\frac{1}{1+e^{-k(x-x_0)}}\right), \quad (13)$$

where A_{sig} is the amplitude, k is the slope parameter, and x_0

determines the center position of the transition.

A polynomial distribution-based baseline is modeled using polynomials of degree 2 to 5 to simulate various curved baseline shapes. By adjusting the polynomial coefficients, the curvature and complexity of the baseline can be flexibly controlled. The corresponding equation is given in Equation (14).

$$B_{poly}(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n, \tag{14}$$

where $c_{0,}c_{1,}\cdots,c_n$ are the coefficients that define the curve, and n represents the polynomial degree. These coefficients are determined by performing polynomial fitting (polyfit) on the reference spectrum.

The random drifting baseline is introduced to model irregular and unpredictable variations that cannot be captured by the mathematically defined baseline functions described above. In real-world Raman spectra, nonlinear or abrupt fluctuations frequently occur, which are difficult to describe using a single function or polynomial expression. To address this, we constructed the random drifting baseline by incorporating randomly generated rising and falling segments, along with their respective durations, amplitudes, and directions. To ensure both smoothness and continuity while maintaining the stochastic nature of the baseline, a Gamma distribution and low-pass filtering were applied. The corresponding formulation is given in Equation (15), where *b[i]* represents the intensity value of the initial random drifting baseline at global position index *i*.

$$b[i+j] = b[i-1] + \frac{h \cdot j}{d} \cdot d_{dir}, \tag{15}$$

where j is the local index within each rising or falling segment, and the parameters h, d, and d_{dir} control the height, duration, and direction of each segment, respectively.

The curve generated using Equation (15) is then smoothed using a low-pass filter to yield the final random drifting baseline, denoted as

$B_{random}[i]$. The filtering process is defined in Equation (16).

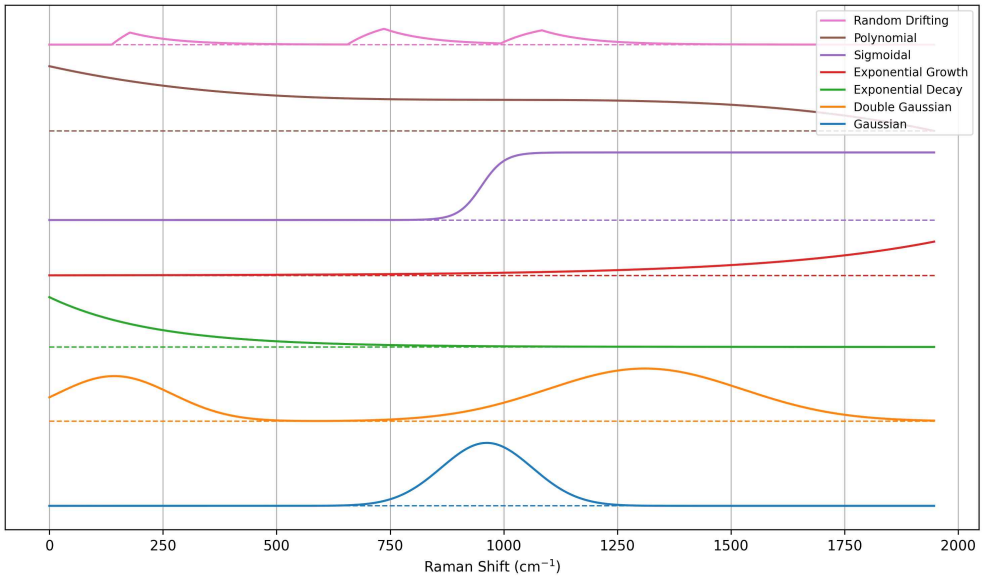
$$B_{random}[i] = \alpha B_{random}[i-1] + (1-\alpha)b[i], \tag{16}$$

where α is the smoothing coefficient, fixed at 0.99.

Baseline Type	Amplitude A Height h	Width σ Duration d Exponential rate λ Slope Parameter k	Center Position (μ or x_0) Degree n Direction d_{dir}
Gaussian	$A_{gau} \sim U(\min(S_{ref}(x)) + 500, \max(S_{ref}(x)))$	$\sigma_{gau} \sim U\left(100, \frac{1}{3} \cdot len(S_{ref}(x))\right)$	$\mu_{gau} \sim U(1000, len(S_{ref}(x)) - 1000)$
Double Gaussian	$A_{1,dgau}, A_{2,dgau} \sim U(\min(S_{ref}(x)) + 300, \max(S_{ref}(x)))$	$\sigma_{1,dgau}, \sigma_{2,dgau} \sim U\left(100, \frac{1}{5} \cdot len(S_{ref}(x))\right)$	$\mu_{1,dgau} \sim U(0, len(S_{ref}(x))/2)$, $\mu_{2,dgau} \sim U(len(S_{ref}(x))/2, len(S_{ref}(x)))$
Exponential Decay	$A_{dexp} \sim U(\min(S_{ref}(x)) + 500, \max(S_{ref}(x)))$	$\lambda_{dexp} \sim U(0.001, 0.009)$	–
Exponential Growth	$A_{gexp} \sim U(\min(S_{ref}(x)) + 500, \max(S_{ref}(x)))$	$\lambda_{gexp} \sim U(0.001, 0.009)$	–
Sigmoidal	$A_{sig} \sim U(\min(S_{ref}(x)) + 500, \max(S_{ref}(x)))$	$k \sim u(0.001, 0.003)$	$x_0 \sim U(1000, len(S_{ref}(x)) - 1000)$
Polynomial	–	–	$n \sim U\{2,5\}$
Random Drifting	$h \sim Gamma(3,300)$	$d = \min\begin{pmatrix} len(S_{ref}(x)) - i, \\ Gamma(1.5,4) \end{pmatrix}$	d_{dir} determined by current value and $Bernoulli(0.8)$

[Table 4-2] Parameter configurations used in synthetic baseline generation

[Table 4-2] summarizes the parameter settings used to construct the synthetic baselines. The amplitude is defined with respect to the reference spectrum $S_{ref}(x)$, and $len(S_{ref}(x))$ denotes its total length. Each baseline type is generated by randomly sampling its parameters within the ranges specified in the table, and a random combination of baseline types, up to a predefined maximum, is applied to each spectrum. This configuration enables the creation of both individual and composite baseline shapes, realistically capturing the variability and complexity of actual Raman spectra and thereby enhancing the model's generalization capability. Representative examples of the constructed baseline types are shown in [Figure 4-4]. In practice, these baseline components are combined in various ways to simulate complex baselines encountered in real Raman spectra.



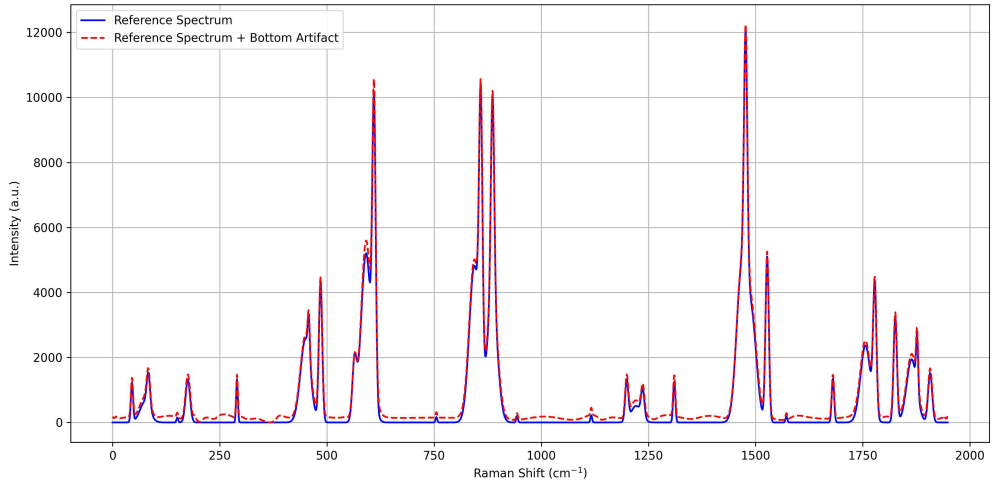
[Figure 4-4] Representative examples of synthetic baseline types

3) Design and Construction of Synthetic Bottom Artifacts

To construct a more realistic dataset, we additionally incorporate bottom artifacts. These artifacts represent subtle background fluctuations frequently observed in Raman spectra, which may resemble baseline distortions and lead to misinterpretations during analysis. Coca-Lopez et al.[49] highlighted that irregular fine structures, such as low-intensity spikes or noise-induced features often caused by cosmic rays or background noise, can result in significant errors during spectral processing. They emphasized the importance of identifying and removing such artifacts to ensure accurate analysis. Since these fluctuations vary depending on experimental environments and sample conditions, they must be explicitly considered during training to enable BC-Former to effectively learn the distinction between the baseline and the reference spectrum. In particular, this data generation strategy encourages the model to accurately distinguish and suppress such complex background patterns.

Parameter	Setting Value
Number of Points	100
Position	$F = \{0,1,...,9\} \cup \{len(S_{ref}(x)) - 10,...,len(S_{ref}(x)) - 1\}$ $R \subset U(\{10,11,...,len(S_{ref}(x)) - 11\}),\ \|R\| = 100 - \|F\|$
Amplitude	$\sim U(max(S_{ref}(x)) \cdot 0.381,\ max(S_{ref}(x)) \cdot 0.384)$
Scaling Factor	$\sim u(1.5,2.8)$

[Table 4-3] Parameter configurations used in bottom artifact generation



[Figure 4-5] Representative example of synthetic bottom artifact

In this approach, bottom artifacts are generated based on multiple selected points within the spectral region. First, a boundary set F is defined by including points near the beginning and end of the spectrum to ensure the presence of bottom artifacts near the spectral boundaries. Then, an interior set R is constructed by randomly selecting positions across the entire spectral domain, excluding the boundary regions already assigned to F. Finally, F and R are merged to construct the final bottom artifact set comprising 100 points, $F \cup R$. At each selected position, an amplitude value is randomly assigned within a range defined relative to the maximum intensity of the reference spectrum, $S_{ref}(x)$. These irregularly distributed points are then connected into a continuous and smooth curve using cubic interpolation. To further reflect realistic intensity variations, a randomly selected scaling factor is applied to the entire curve. All parameter values used to generate the bottom artifacts are empirically determined based on actual Raman spectra, aiming to replicate the subtle amplitude fluctuations and irregular background patterns commonly observed in real measurements. The specific

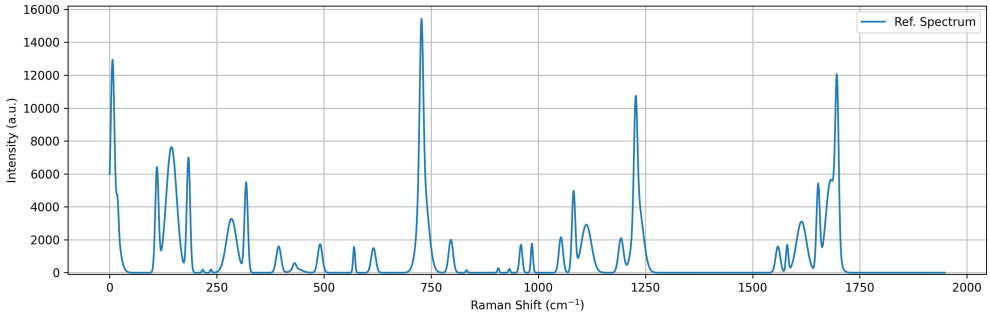
parameter settings are summarized in [Table 4-3]. The effect of bottom artifact addition is further illustrated in [Figure 4-5], which shows how their inclusion leads to more realistic baseline scenarios and emphasizes the additional difficulty they pose in baseline correction tasks.

4) Construction of the Baseline-Included Spectrum

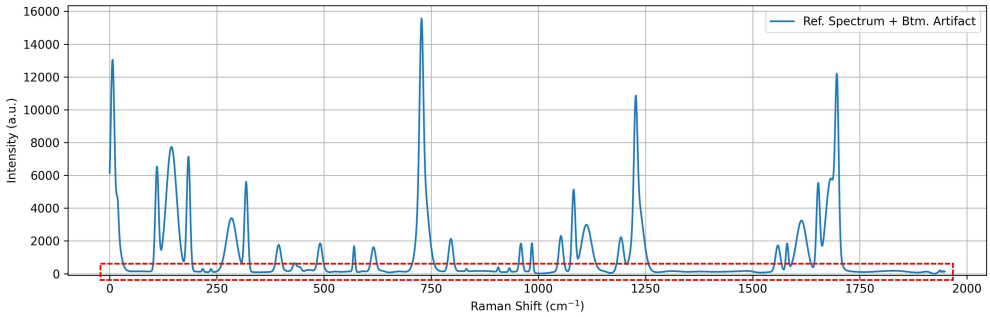
This section describes the procedure for constructing the final baseline-included spectrum by combining three previously defined components. The baseline-included spectrum is generated in three main steps, as shown in Equation (17). Starting from the reference spectrum $S_{ref}(x)$, the bottom artifact $A(x)$ is first added, followed by the addition of the synthetic baseline $B(x)$ composed of randomly selected baseline types.

$$S_{bi}(x) = S_{ref}(x) + A(x) + B(x). \tag{17}$$

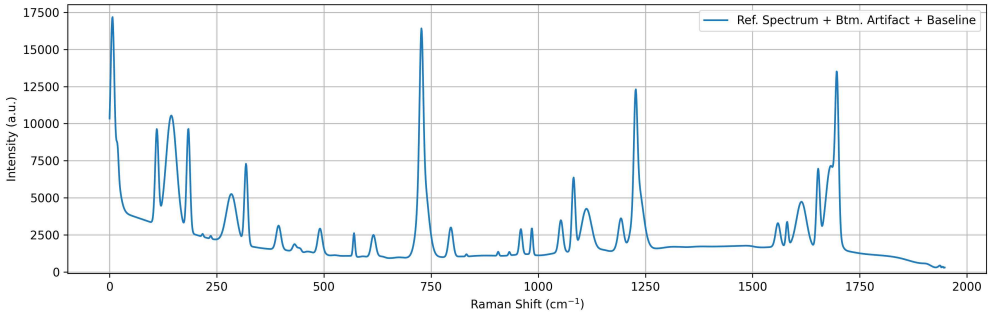
By incorporating diverse baseline patterns and subtle background fluctuations, the resulting baseline-included spectrum $S_{bi}(x)$ provides a realistic representation of actual Raman spectra. This spectrum serves as the input for training the BC-Former model, while the corresponding reference spectrum $S_{ref}(x)$ is used as the ground truth. [Figure 4-6] illustrates the synthetic data generation process, showing how each component is sequentially integrated to construct synthetic spectra. The parameter settings in [Table 4-1], [Table 4-2], and [Table 4-3] were partially derived from existing literature[50], particularly for certain types of baselines, while others were manually adjusted according to recurring patterns and characteristics observed in real Raman spectra. Through this design, the constructed synthetic dataset captures the inherent complexity and variability of real-world Raman spectra, enabling BC-Former to effectively learn and generalize diverse baseline structures and artifact-induced distortions.



(a) Reference spectrum



(b) Reference spectrum with bottom artifact



(c) Reference spectrum with bottom artifact and additional baseline

[Figure 4-6] Representative examples of the synthetic data generation process

4.1.3 Preprocessing for Model Training

The input data for BC-Former undergoes preprocessing to enhance the effectiveness of model training. This step plays a crucial role in

ensuring data consistency and training stability. The following key preprocessing techniques were applied.

4.1.3.1 Normalizing

To ensure consistent training behavior and enable BC-Former to focus on relative spectral patterns, all input spectra are normalized to the range of [0, 1] using min-max normalization. Instead of applying normalization based on the global minimum and maximum across the entire dataset, each sample is normalized individually using its own minimum and maximum values. This approach preserves the unique characteristics of each spectrum without introducing inter-sample distortion. To maintain a one-to-one correspondence between each baseline-included spectrum and its reference counterpart, a tailored normalization scheme is applied. For each pair, the maximum value of the baseline-included spectrum is used as the upper bound, and the minimum value, defined as zero in the reference spectrum, is used as the lower bound. This design reflects the increase in overall spectral intensity caused by the added baseline. This sample-wise normalization ensures consistency in input representation, supports stable model convergence, and enables BC-Former to effectively remove baseline influence and reconstruct the original spectral signal.

4.1.3.2 Padding

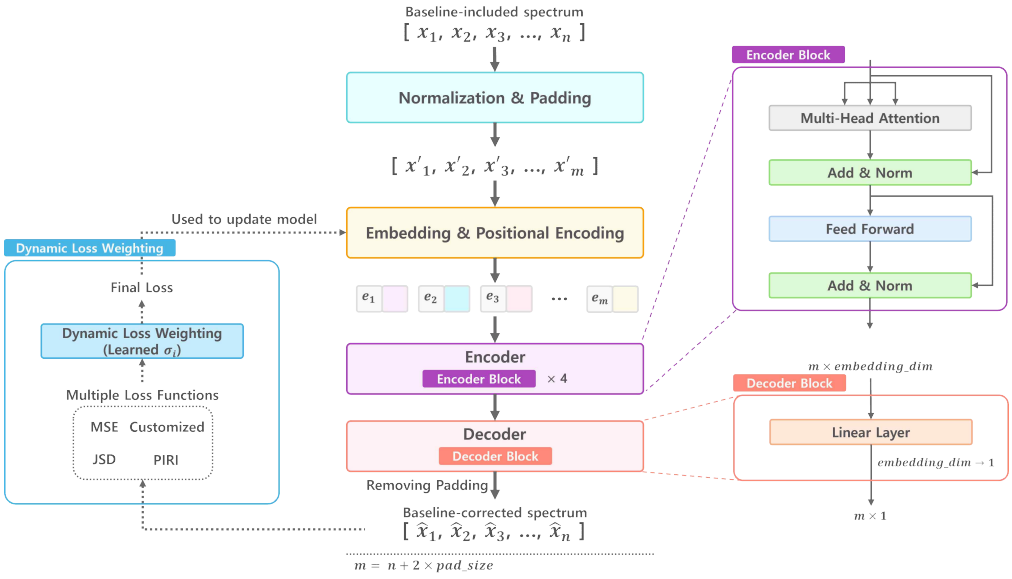
After normalization, 150 padding values are appended to both ends of each input sequence to prevent potential information loss at the boundaries. Rather than applying simple zero padding, the padding is performed by repeating the first and last values of the sequence. This approach helps preserve the natural flow and trend of the 1D data. Since BC-Former is designed to learn continuous spectral patterns,

discontinuities at the sequence boundaries can negatively impact prediction performance. To address this, the padding strategy smooths boundary conditions and allows the model to more stably learn the global structure of the spectrum.

4.1.4 Configuration of BC-Former

This section describes the architecture and training strategy of BC-Former. Rather than simply adopting a Transformer encoder, BC-Former integrates its architecture with preprocessing and carefully designed loss functions, enabling the model to address the nonlinear and diverse baseline patterns observed in spectroscopic data. The following subsections provide an overview of the model architecture, the loss functions employed, and the training strategy used to optimize performance.

4.1.4.1 Architecture



[Figure 4-7] Overview of the BC-Former model architecture

The architecture and core operations of BC-Former are illustrated in [Figure 4-7]. First, the input to the model is a baseline-included spectrum $[x_1, x_2, x_3, \cdots, x_n]$, which undergoes normalization and padding, resulting in an extended input sequence of length m, $[x'_1, x'_2, x'_3, \cdots, x'_m]$. Each scalar value is transformed into a vector representation $[e_1, e_2, e_3, \cdots, e_m]$ through an embedding layer that projects the data into a vector space of predefined dimension. This transformation enables the input data to be represented in a high-dimensional format that is suitable for processing by the Transformer model. Since the Transformer architecture does not inherently preserve sequential order, positional encoding is applied to incorporate positional information. We adopted sinusoidal positional encoding using sine and cosine functions, which are added to each embedding vector to help the model recognize the order and relative distance of data points within the sequence. Next, BC-Former employs a Transformer Encoder composed of multiple stacked encoder blocks to process the embedded sequence. Each encoder block consists of a multi-head self-attention mechanism and a position-wise feed-forward network. These components enable the model to capture complex dependencies across sequence positions and perform nonlinear transformations at each position. In particular, the attention mechanism is effective in modeling long-range interactions and global spectral patterns. To further enhance training stability and improve generalization, layer normalization and dropout are applied within each layer. Finally, a decoder block composed of a single linear layer is used to reconstruct the output as a 1D spectral sequence with the same feature dimension as the original input. After reconstruction, the padded regions are removed, thereby restoring the sequence to its original length n and producing the final baseline-corrected spectrum $[\widehat{x_1}, \widehat{x_2}, \widehat{x_3}, \cdots, \widehat{x_n}]$. In addition to this architectural design, BC-Former employs multiple loss functions

during training and incorporates the DLW strategy to automatically adjust the relative importance of each loss component, thereby achieving high-performance baseline correction. Through this combination, BC-Former is able to reliably correct the complex baseline patterns encountered in practical Raman spectra.

4.1.4.2 Loss functions and training strategies

To optimize the training of BC-Former, a combination of multiple loss functions is employed. The specific loss functions used are described below.

1) Mean Squared Error Loss

MSE Loss measures the average squared difference between the model's output $\hat{y_i}$ and the ground truth reference spectrum y_i, and is computed as shown in Equation (18).

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 \tag{18}$$

2) Customized Loss

The Customized Loss is a composite metric based on the PCC[51] and CS[52]. By integrating these two metrics, the loss function effectively captures the pattern similarity between the predicted and reference spectra, encouraging the model to replicate the actual spectral shape more accurately during training.

The PCC is defined in Equation (19) and measures the linear correlation between the predicted spectrum and the ground truth.

$$PCC = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(\hat{y_i} - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2} \cdot \sqrt{\sum_{i=1}^{N}(\hat{y_i} - \bar{\hat{y}})^2}}, \tag{19}$$

where y_i and $\hat{y_i}$ denote the ground truth and predicted values, respectively, and $\bar{y}$ and $\bar{\hat{y}}$ denote their means. A PCC value closer to 1 indicates a strong positive correlation, implying that the two vectors exhibit highly similar overall shapes.

The CS is defined in Equation (20) and measures the directional similarity between two vectors.

$$CS = \frac{\sum_{i=1}^{N} y_i \hat{y_i}}{\sqrt{\sum_{i=1}^{N} y_i^2} \cdot \sqrt{\sum_{i=1}^{N} \hat{y_i}^2}}. \tag{20}$$

A CS value closer to 1 indicates that the predicted and reference spectra are aligned in direction, suggesting high similarity in relative intensity distribution across spectral points.

The Customized Loss is defined as the negative average of PCC and CS, as shown in Equation (21).

$$L_{Custom} = -\frac{1}{2}(PCC + CS). \tag{21}$$

This formulation ensures that the loss decreases as the predicted spectrum becomes more similar to the reference, both in overall shape and direction. Accordingly, during training, the Customized Loss encourages the model to produce outputs that faithfully reflect the structural patterns of the target spectra.

3) Jensen–Shannon Divergence Loss

Jensen–Shannon Divergence (JSD)[53] Loss is a statistical measure of similarity between two probability distributions. It is employed to reduce discrepancies between the predicted and ground truth spectra, thereby guiding the model to preserve the structural integrity of spectral peaks. The loss is defined in Equation (22) as follows.

$$L_{JSD} = \frac{1}{2}\sum_{i=1}^{N} P_i \log\left(\frac{P_i}{M_i}\right) + \frac{1}{2}\sum_{i=1}^{N} Q_i \log\left(\frac{Q_i}{M_i}\right), \quad (22)$$

where P and Q represent the ground truth and predicted distributions, respectively, and $M = \frac{1}{2}(P+Q)$ denotes their average distribution. By minimizing L_{JSD}, the model learns to generate outputs that approximate the reference spectra in both distributional form and peak structure.

4) Peak Intensity Ratio Inconsistency Loss

The Peak Intensity Ratio Inconsistency (PIRI) Loss is designed to enhance correction performance in peak regions by enforcing the preservation of relative intensities among major peaks. To achieve this, the PIRI Loss is computed through the following procedure. First, a set of major peak positions $P = \{p_1, p_2, \cdots, p_K\}$ is extracted from the reference spectrum. Then, for each peak position p_i, the relative intensity difference between the model output $\hat{y}_{p_i}$ and the ground truth y_{p_i} is calculated. The overall PIRI Loss is defined based on this difference, as shown in Equation (23).

$$L_{PIRI} = \frac{1}{K}\sum_{i=1}^{K}\left(\frac{|y_{p_i} - \hat{y}_{p_i}|}{y_{p_i} + \epsilon} \cdot y_{p_i}\right), \quad (23)$$

where ϵ is set to 10^{-7} to avoid division by zero, and the term y_{p_i} multiplied with the fraction serves as a weight to emphasize the contribution of larger peaks during the loss calculation.

5) Dynamic Loss Weighting

Algorithm 2 Dynamic loss weighting strategy	
input:	Training data D; Initial model parameters θ; Learnable task-specific uncertainty parameters $\sigma = \{\sigma_{MSE}, \sigma_{Customized}, \sigma_{JSD}, \sigma_{PIRI}\}$
Output:	Optimized model θ^*, Optimized weights σ^*
1:	Initialize θ and σ.
2:	**While** not converged **do**:
3:	Sample batch (x, y) from D.
4:	Predict baseline $\hat{y} = Model(x;\theta)$.
5:	Calculate individual losses: $L_{MSE}, L_{Customized}, L_{JSD}, L_{PIRI}$.
6:	Compute total weighted loss: $L_{total} = \sum(\frac{L_i}{\sigma_i^2} + \log\sigma_i^2)$
7:	Update θ and σ using backpropagation (Adam optimizer).
8:	**End While**

[Table 4-4] Algorithm for Training BC-Former using Dynamic Loss Weighting strategy

In static-weight approaches, the relative importance of each loss function must be manually specified, and the predefined weights are uniformly applied throughout the entire training process. However, the contribution of each loss term can vary dynamically during training, and fixed weights may hinder optimal learning. To address this, we adopt the DLW strategy[54], which allows the model to automatically adjust the importance of each loss component throughout training. This strategy is grounded in the concept of task-specific homoscedastic uncertainty, which adaptively balances each loss term according to its variance.

The detailed training procedure using this strategy is summarized in Algorithm of [Table 4-4]. As outlined in Line 1, the process begins by initializing the model parameters θ and the learnable task-specific uncertainty parameters σ. The iterative training phase, spanning Lines 2 through 8, involves sampling a data batch and generating the baseline

prediction $\hat{y}$ as depicted in Lines 3 and 4. Following this prediction, Line 5 specifies the calculation of individual loss components, including $L_{MSE}, L_{Customized}, L_{JSD}, L_{PIRI}$. The total weighted loss is then derived in Line 6 according to Equation (24).

$$L_{total} = \sum_{i=1}^{n} \left(\frac{L_i}{\sigma_i^2} + \log \sigma_i^2 \right), \tag{24}$$

where L_i denotes the i-th loss function and $\log \sigma_i^2$ is a learnable log-variance parameter. Finally, as detailed in Line 7, the optimization step simultaneously updates both the model parameters θ and the uncertainty weights σ via backpropagation using the Adam optimizer. This formulation effectively suppresses the influence of loss terms with high uncertainty while penalizing excessive variance, enabling BC-Former to robustly learn diverse baseline and peak distortions. A detailed performance comparison between static and DLW strategies is provided in Section 4.2.2.1.

4.2 Evaluation

In this section, we present the experimental procedures and results to systematically validate the performance of BC-Former. We first describe the implementation settings used for performance evaluation, and then provide a comprehensive evaluation of BC-Former through various experiments.

4.2.1 Implementation details

All experiments were conducted in a consistent computing environment. The hardware setup included four RTX A6000 GPUs, an AMD Ryzen CPU running at 7.0 GHz, and 46 GB of RAM, operating on Ubuntu 20.04 LTS with Linux kernel version 5.15. The software environment was built using Python 3.8.18 and PyTorch[55] 2.4.1 with

CUDA 12.1 support. Commonly used libraries such as NumPy[42], Pandas[43], SciPy[38], and scikit-learn[56] were used for data preprocessing and analysis. In this environment, the BC-Former model was implemented based on the architecture described in Section 4.1.4.1, with detailed configuration summarized in [Table 4-5].

Stage	Layer	Description	Output Dimensiion
Input	-	Normalized 1D Raman spectrum	$(size_{batch}, 1, length_{sequence})$
Embedding	Linear	Projects 1D input to 64D embedding space	$(size_{batch}, length_{sequence}, 64)$
Positional Embedding	- (sinusoidal)	Added to embedding	$(size_{batch}, length_{sequence}, 64)$
Transformer Encoder	4 layers, 4 heads, 512 feedforward dimension, dropout 0.1	Multi-head self-attention and feedforward processing	$(size_{batch}, length_{sequence}, 64)$
Decoder	Linear	Projects back to original 1D space	$(size_{batch}, 1, length_{sequence})$

[Table 4-5] Detailed architecture of BC-Former

For training, a synthetic dataset comprising 8,000 training samples, 1,000 validation samples, and 1,000 test samples was used. The experimental dataset consisted of simulated Raman spectra of 2-chloroethyl ethyl sulfide, diisopropyl fluorophosphate, and dimethyl methylphosphonate dissolved in tetrahydrofuran, measured under four accumulations with a 532 nm laser operating at 100% power (66 mW).

The main training hyperparameters were a dropout rate of 0.1, a learning rate of 10^{-5}, and a batch size of 32. Training was performed for up to 1,000 epochs, with early stopping applied if the validation mean absolute error (MAE) did not improve for 100 consecutive epochs. The Adam[44] optimizer was employed with a weight decay of 10^{-5}.

Model performance was evaluated using multiple quantitative metrics.

PCC was calculated using the *scipy.stats.pearsonr* function, and CS was computed using PyTorch's *cosine_similarity* function. ED[57] was defined as the L2 norm between spectral vectors. Additionally, to evaluate baseline correction quality from a distributional perspective, a 1D adaptation of the FID was implemented. Together, these metrics provided a comprehensive assessment of the baseline correction performance of BC-Former.

4.2.2 Performance evaluation results

This section presents a comprehensive evaluation of BC-Former using a series of experiments. The analysis includes an examination of the loss weighting strategy, an assessment of performance across different hyperparameter configurations, comparisons with existing algorithms and models, and an evaluation of adaptability to previously unseen baseline patterns. Both quantitative and qualitative results are reported, providing thorough validation of the effectiveness of BC-Former.

4.2.2.1 Effectiveness of Dynamic Loss Weighting

This experiment evaluates the effectiveness of the DLW technique proposed in Section 4.1.4.2 (5) by comparing baseline correction performance under different loss weighting strategies. The comparison was conducted across three main groups. First, in the single static loss setting, only one loss function was used at a time. The vectors (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), and (0, 0, 0, 1) correspond to training with only MSE Loss, Customized Loss, JSD Loss, or PIRI Loss, respectively. Second, in the multiple static loss setting, several loss functions were combined, with one assigned a higher weight. Third, the proposed DLW technique was applied, allowing the model to automatically adjust the importance of each loss term during training.

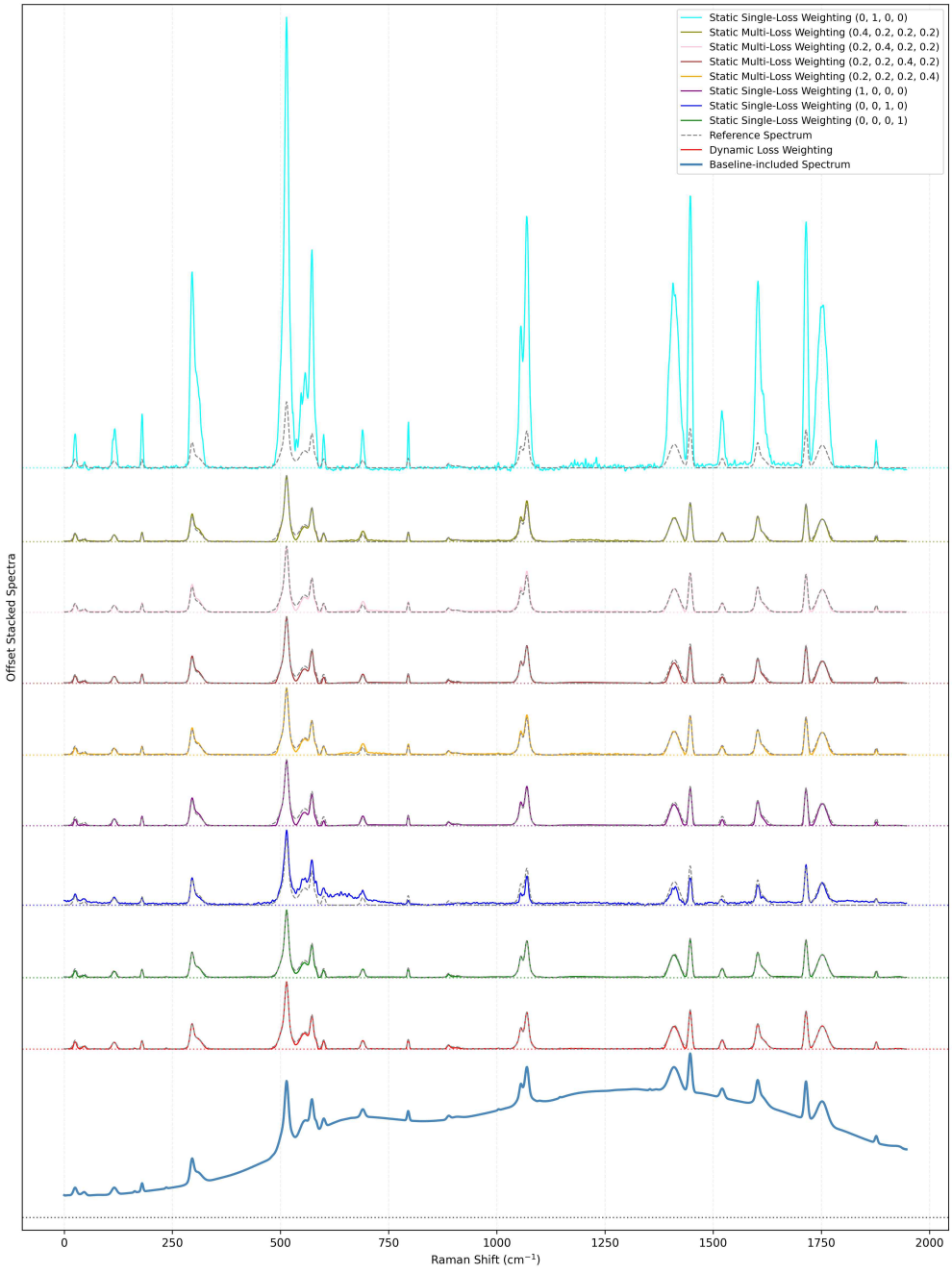
Method	CS(↑)	PCC(↑)	FID(↓)	ED(↓)
Static Multi-Loss Weighting (0.4, 0.2, 0.2, 0.2)	0.9920	0.9905	0.1151	0.5290
Static Multi-Loss Weighting (0.2, 0.4, 0.2, 0.2)	0.9919	0.9904	0.1175	0.5325
Static Multi-Loss Weighting (0.2, 0.2, 0.4, 0.2)	0.9922	0.9909	0.1128	0.5239
Static Multi-Loss Weighting (0.2, 0.2, 0.2, 0.4)	0.9914	0.9899	0.1231	0.5283
Static Single-Loss Weighting (1, 0, 0, 0)	0.9917	0.9903	0.1188	0.5495
Static Single-Loss Weighting (0, 1, 0, 0)	0.9927	0.9919	155.7237	22.1093
Static Single-Loss Weighting (0, 0, 1, 0)	0.9513	0.9503	0.7199	1.4296
Static Single-Loss Weighting (0, 0, 0, 1)	0.9909	0.9900	0.1195	0.5734
Dynamic Loss Weighting	0.9978	0.9974	0.0370	0.2861

[Table 4-6] Quantitative performance comparison of different loss weighting strategies

[Table 4-6] presents a quantitative comparison of different loss weighting strategies. Among the static settings, the multiple static configurations, which integrate several loss functions, consistently outperformed the single static configurations that rely on only one loss function. Notably, the configuration that assigns a higher weight to the JSD Loss, represented by the weight vector (0.2, 0.2, 0.4, 0.2), demonstrated the most balanced performance and achieved the best average results across the evaluation metrics, excluding the DLW. This indicates that emphasizing spectral distribution similarity plays a critical role in enhancing baseline correction accuracy. While these static configurations show strong performance, the proposed DLW strategy outperformed all of them across every evaluation metric.

[Figure 4-8] visually compares the baseline correction results obtained using different loss weighting strategies. As shown in the figure, the static single-loss configurations clearly illustrate the limitations of

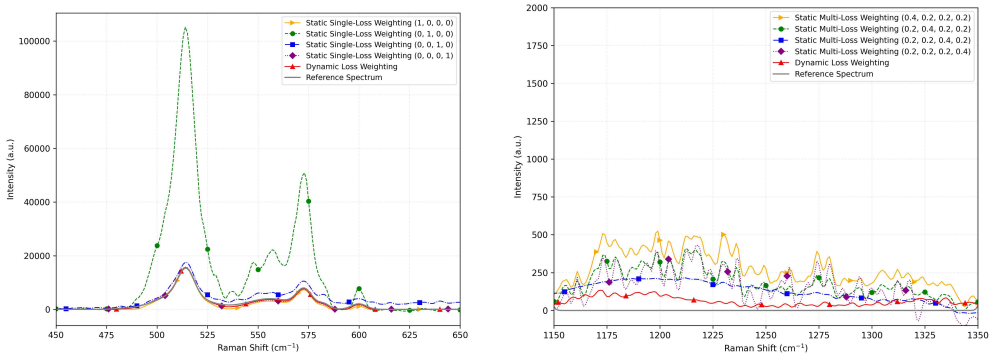
this approach, which struggles to capture the complex characteristics of Raman spectra. These results indicate that a single loss function is insufficient to comprehensively characterize Raman spectra with peaks of varying intensities and shapes. On the other hand, static multi-loss configurations generally achieved better correction performance than single-loss settings. However, they still exhibited notable limitations. For instance, the configuration emphasizing JSD Loss (0.2, 0.2, 0.4, 0.2) resulted in overcorrection of peaks in the 500-750 cm^{-1} range. In contrast, the proposed DLW method effectively restored complex peak structures in the 500, 1100, 1450, and 1700 cm^{-1} regions. Among all configurations, it produced results that most closely matched the reference spectrum.



[Figure 4-8] Visual comparison of baseline correction results under different loss weighting strategies

[Figure 4-9] presents a magnified view of two Raman shift regions

from the baseline correction results in [Figure 4-8], enabling a more detailed visual comparison. These regions are particularly challenging, one due to overlapping shoulder peaks and the other being a peak-free interval that demands clean baseline removal, making them suitable benchmarks for evaluating correction accuracy. In [Figure 4-9] (a), the static single-loss configurations fail to model complex peak structures, resulting in exaggerated shapes or overcorrection. In contrast, the DLW strategy yields results most closely matching the reference spectrum, accurately restoring fine-grained structures such as peak height, position, and distribution. Similarly, [Figure 4-9] (b) shows that minor residual baselines remain in certain areas, whereas DLW achieves near-complete restoration of the reference spectrum with minimal residual artifacts.



(a) Results in the $450-650cm^{-1}$ region (b) Results in the $1150-1350cm^{-1}$ region

[Figure 4-9] Visual comparison of baseline correction results under different loss weighting strategies in key spectral regions

Taken together, the results from [Table 4-6], [Figure 4-8] and [Figure 4-9] demonstrate that the proposed DLW strategy achieves the highest restoration precision both at the full-spectrum level and in localized regions with either critical peaks or peak-free intervals. These results highlight that combining complementary loss functions with an

adaptive mechanism enables robust and generalizable baseline correction.

4.2.2.2 Performance Impact of Hyperparameter Configurations

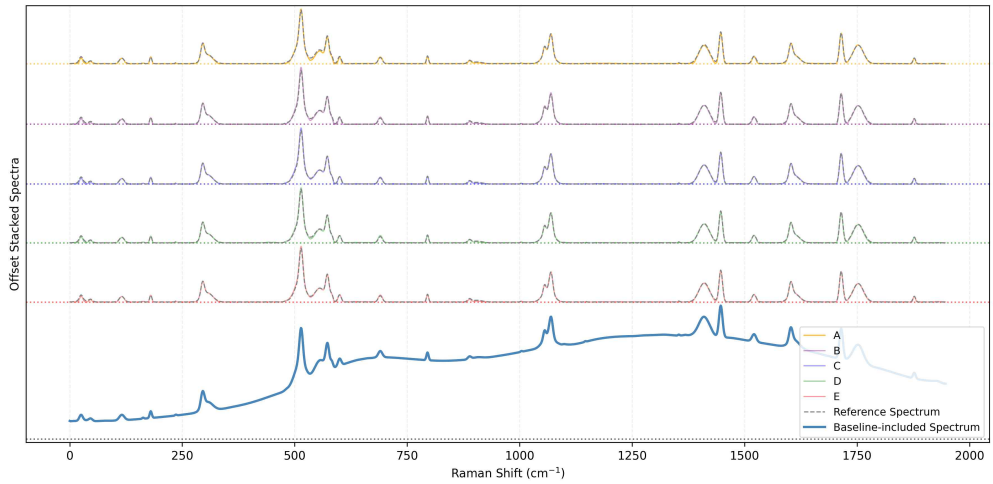
ID	nhead	emb_dim	ff_dim	CS(↑)	PCC(↑)	FID(↓)	ED(↓)
A	4	64	512	0.9978	0.9974	0.0370	0.2861
B	8	128	512	0.9981	0.9978	0.0375	0.3047
C	8	128	1024	0.9981	0.9978	0.0312	0.2635
D	16	256	1024	0.9984	0.9981	0.0267	0.2468
E	16	256	2048	0.9983	0.9979	0.029	0.2582

[Table 4-7] Performance comparison under different hyperparameter configurations

In this experiment, we examine the impact of key hyperparameter values on the baseline correction performance of BC-Former. Under the same DLW setting, we systematically varied the number of attention heads (nhead), the embedding dimension (emb_dim), and the feedforward network dimension (ff_dim) to analyze performance trends and identify the point at which improvements plateau. [Table 4-7] summarizes the quantitative results across different hyperparameter configurations. As shown in the table, *D* achieved the best overall performance, yielding the highest PCC and CS values and the lowest FID and ED scores. Notably, even the smallest configuration *A* delivered highly competitive results compared to larger models, suggesting that compact architectures can still provide efficient and robust performance in Raman spectral baseline correction. [Figure 4-10] presents a visual comparison of baseline correction results across different hyperparameter settings. *D* and *E* showed strong overall performance, while *A* also produced stable correction results, confirming its relatively robust performance.

Overall, both quantitative and qualitative evaluations indicate that *D* achieves the best performance. However, considering computational cost and practical applicability, the smaller configuration *A* offers sufficiently

competitive performance. Based on these findings, all subsequent experiments employed configuration A to enhance computational efficiency and facilitate deployment in real-world scenarios.



[Figure 4-10] Visual comparison of baseline correction results under different hyperparameter configurations

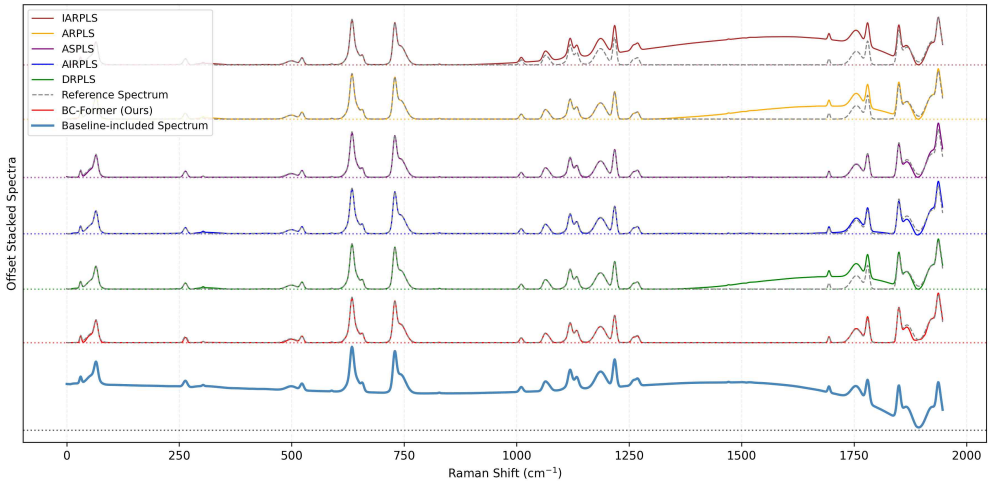
4.2.2.3 Performance Comparison with Signal Processing-Based Methods

In this experiment, we compared the performance of BC-Former with several signal processing-based methods[58]. Among the available algorithms, we selected five representative methods that are widely recognized for their robust performance, namely AIRPLS, DRPLS, ARPLS, ASPLS and IARPLS. [Table 4-8] presents the quantitative comparison between BC-Former and these five signal processing methods. The Rank column indicates the average ranking of each method across the three evaluation metrics (FID, CS, and ED), where a lower average rank corresponds to better overall performance. Since PCC and CS are highly correlated and capture the same notion of spectral similarity, PCC was excluded from the ranking calculation to avoid redundancy. As shown in

the table, existing methods showed considerable variability across metrics. These results highlight a key limitation of conventional signal processing methods, namely their limited representational capacity in handling complex baseline and spectral patterns. In contrast, BC-Former's architecture allows flexible adaptation to diverse baseline patterns and noise conditions. Notably, BC-Former consistently achieved superior performance with the lowest average rank of 1.24. It also outperformed all methods across quantitative indicators such as global distribution similarity (FID), absolute spectral difference (ED), relative peak alignment (PCC), and intensity ratio consistency (CS), delivering substantial improvements in overall baseline correction quality.

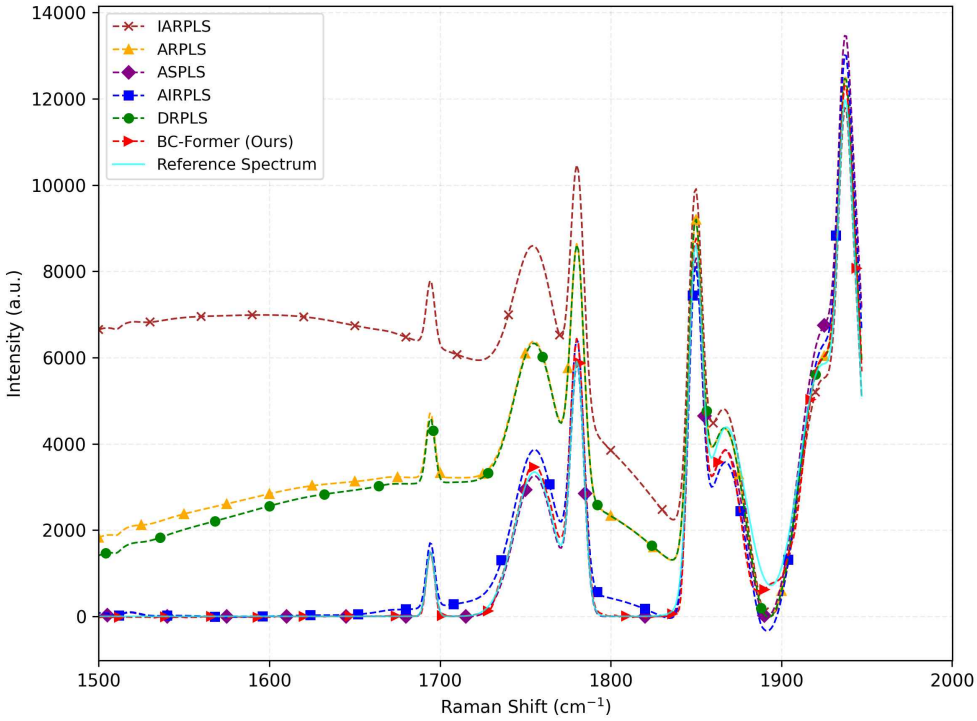
Method	CS(↑)	PCC(↑)	FID(↓)	ED(↓)	Rank(↓)
IARPLS	0.8754	0.8467	4.7624	2.9014	7.89
ARPLS	0.9668	0.9588	0.8603	0.9095	3.73
ASPLS	0.9956	0.9949	0.0713	0.4101	3.36
AIRPLS	0.9765	0.9729	0.5289	0.8126	4.64
DRPLS	0.9747	0.9686	0.6385	0.7546	3.09
BC-Former (Ours)	0.9978	0.9974	0.0370	0.2861	1.24

[Table 4-8] Performance comparison between BC-Former and signal processing-based methods



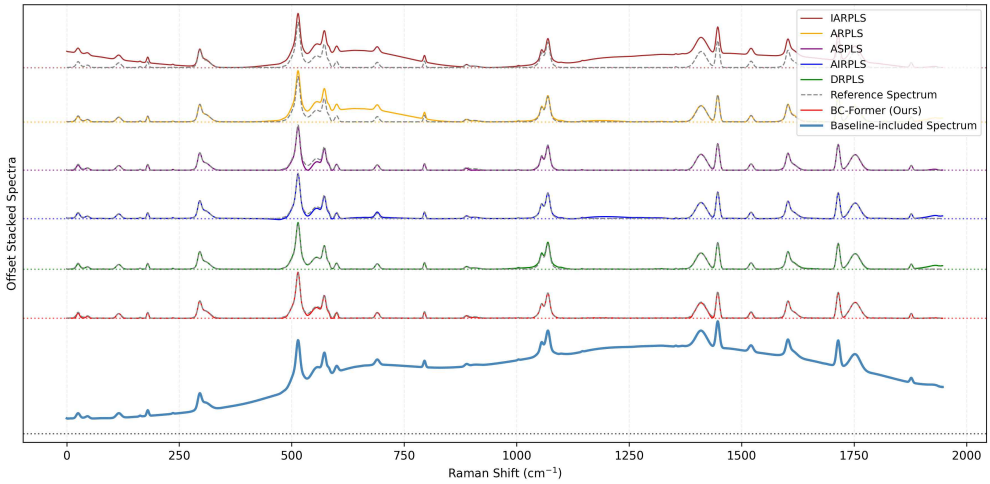
[Figure 4-11] Visual comparison of baseline correction performance between BC-Former and signal processing-based methods on the first synthetic data

[Figure 4-11]-[Figure 4-14] illustrate the results of comparison between BC-Former and various existing methods using synthetic datasets with diverse baseline patterns. [Figure 4-11] visualizes the baseline correction results for the first synthetic data. Among the traditional methods, DRPLS showed insufficient baseline removal in the 1500-1850 cm^{-1} region, which degraded the overall correction quality. AIRPLS caused noticeable distortion of peak shapes near 1750 cm^{-1}. ASPLS produced inconsistent baseline correction around 1900 cm^{-1}. By contrast, BC-Former delivers a clean, stable baseline with consistent peak fidelity across the spectrum.



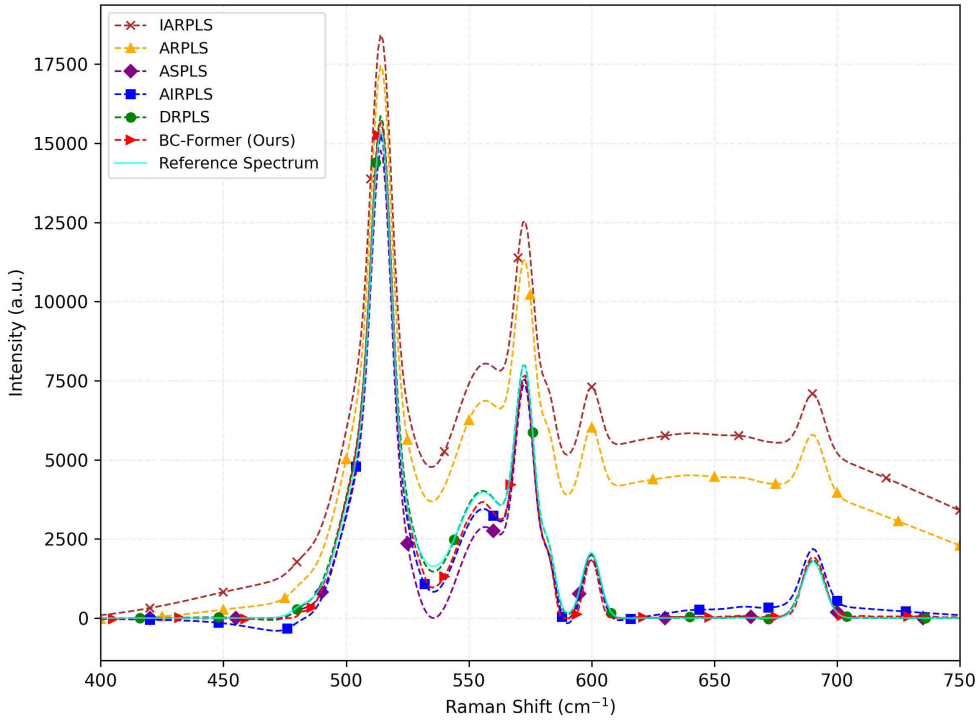
[Figure 4-12] Visual comparison of baseline correction performance between BC-Former and signal processing-based methods in the major peak region (1500-2000 cm^{-1}) of the first synthetic data

[Figure 4-12] provides a detailed comparison of the 1500-2000 cm^{-1} region from the previous figure, which includes multiple narrow and intricately overlapping peaks, making it a challenging case for evaluating fine reconstruction capabilities. IARPLS, ARPLS, and DRPLS tended to preserve excessive baseline components, resulting in elevated intensity levels and blurred separation between adjacent peaks. AIRPLS partially removed the baseline but introduced distortions in peak positions and shapes. ASPLS recovered some peaks successfully, but around 1900 cm^{-1}, it overcorrected the baseline. In contrast, BC-Former closely follows the reference spectrum across this range, preserving peak positions and separations while maintaining a flat baseline.



[Figure 4-13] Visual comparison of performance between BC-Former and signal processing based methods on the second synthetic dataset

[Figure 4-13] shows the baseline correction results for the second synthetic data. Similar to the results in [Figure 4-11] and [Figure 4-12], while the traditional methods generally provided relatively strong correction performance, several limitations became evident upon closer examination. For instance, IARPLS tended to overestimate peak intensities across the spectrum, and ARPLS distorted peak shapes in the 500-750 cm^{-1} region. ASPLS excessively removed shoulder peaks between 500-600 cm^{-1}, thereby damaging the peak structures, whereas AIRPLS maintained overall consistency but exhibited minor baseline correction errors in the 1100-1250 cm^{-1} region. Although DRPLS demonstrated relatively stable performance, it left a residual baseline beyond the 1900 cm^{-1} region. In contrast, BC-Former closely preserved the peak patterns of the reference spectrum and achieved stable and accurate correction across the entire spectral range.



[Figure 4-14] Visual comparison of performance between BC-Former and the signal processing based methods in the key peak region of the second synthetic data

[Figure 4-14] presents a magnified view of the 400-750 cm^{-1} region from the previous figure, which contains a mixture of strong main peaks and complex shoulder peaks requiring highly precise correction. IARPLS and ARPLS failed to adequately remove the baseline, resulting in exaggerated peak intensities, while AIRPLS left residual structures and ASPLS excessively removed shoulder peaks. In contrast to the first synthetic dataset where its performance was limited, DRPLS shows favorable results here, closely matching the reference spectrum and reproducing the 520 cm^{-1} main peak and the 560-600 cm^{-1} multiplet in both height and position. This contrast highlights the variability of DRPLS performance depending on characteristics of baseline. BC-Former, on the

other hand, delivers consistently clean baseline removal and maintains reliable correction quality in this dataset as well.

From [Figure 4-11]-[Figure 4-14], it becomes evident that the signal processing methods exhibit significant performance variation depending on the characteristics of the data. For instance, on the first dataset, ASPLS demonstrated stable restoration of complex peak structures, whereas DRPLS failed to adequately remove the baseline, resulting in inflated intensity levels and reduced peak separability. In contrast, while DRPLS showed relatively strong performance on the second synthetic dataset, ASPLS experienced overcorrection in regions with shoulder peaks. Similar patterns were observed across other methods, illustrating the limitations of algorithms that are overly tuned to specific baseline shapes. Their generalization ability tends to decline when confronted with unfamiliar or more complex baseline variations.
In comparison, BC-Former consistently achieved superior performance across both synthetic datasets. These results validate the generalizability and robustness of our BC-Former, which benefits from training on baseline-included data generated using diverse synthetic patterns. They also highlight the effectiveness of the DLW strategy in allowing the model to flexibly adapt to a wide range of baseline conditions.

4.2.2.4 Performance Comparison with Other Neural Network-Based Models

This experiment compares BC-Former with two alternative neural network-based models for baseline correction. The comparison examines how different architectures affect the quality of Raman spectrum correction and evaluates their strengths and limitations from both quantitative and qualitative perspectives. The first baseline model is an autoencoder-based approach[59], which removes the baseline by

compressing and reconstructing the input spectrum. The second is the Transformer-based model, referred to here as 1dTrans[19]. Built on the Transformer architecture, it is tailored for spectral data and leverages self-attention mechanisms to capture both local and global dependencies along the spectral axis.

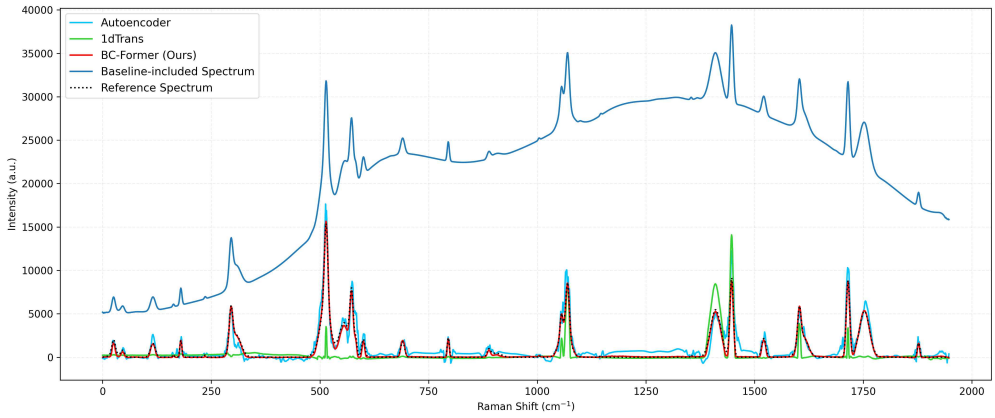
Method	CS(↑)	PCC(↑)	FID(↓)	ED(↓)
Autoencoder	0.9606	0.9526	0.7387	1.2664
1dTrans	0.9528	0.8089	2.2361	2.5124
BC-Former (Ours)	0.9978	0.9974	0.0370	0.2861

[Table 4-9] Quantitative performance comparison between BC-Former and other neural network-based models

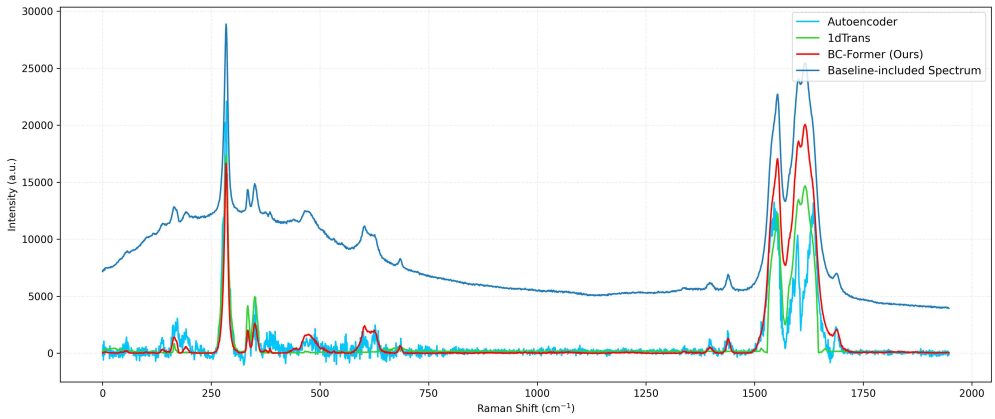
[Table 4-9] presents a quantitative comparison of baseline correction performance among BC-Former, the Autoencoder, and the 1dTrans models. The results indicate that BC-Former achieved the highest similarity in terms of CS and PCC, effectively restoring both the overall spectral shape and peak positions. BC-Former also achieved the lowest values in FID and ED, demonstrating its superior ability to reconstruct the global distribution as well as fine-grained spectral structures. In contrast, the 1dTrans exhibited limited performance, as its triple skip concatenation structure passed baseline-included spectra multiple times via skip connections. Consequently, the predicted baseline tended to resemble the input spectrum more closely than the ground-truth baseline, leading to excessive correction.

[Figure 4-15] presents visual comparisons of baseline correction results on synthetic and real Raman spectral data. In the synthetic dataset ([Figure 4-15] (a)), the Autoencoder model introduced noticeable distortions, including loss of peak height and shape in regions such as 250-400, 500-600, and 1400-1600 cm^{-1}, with residual noise remaining after

correction. The 1dTrans often produced excessive correction and suppressed genuine peaks, causing the corrected spectra to deviate from the reference in several regions. In contrast, BC-Former consistently preserved peak shapes and positions across the Raman shift range, yielding corrected spectra that most closely matched the reference. On the real spectral data ([Figure 4-15] (b)), the Autoencoder introduced substantial noise across the entire range, with strong baseline fluctuations and oscillations. The 1dTrans, while less noisy than the Autoencoder, tended to overcorrect the spectra. In particular, the main peak in the 1500 -1750 cm^{-1} region was excessively suppressed, and the small peaks in the 400-700 cm^{-1} range were almost entirely removed, eliminating potentially informative features. In contrast, BC-Former consistently produced stable and accurate restorations of main peaks, especially in the 250-400 and 1500-1750 cm^{-1} regions. Its corrected baselines were also smoother and exhibited less noise.



(a) Comparison of BC-Former, 1dTrans, and Autoencoder on synthetic data



(b) Comparison of BC-Former, 1dTrans, and Autoencoder on experimental data

[Figure 4-15] Visual comparison of baseline correction performance between BC-Former and Autoencoder/1dTrans models on synthetic/experimental data

In conclusion, the results from [Table 4-8] and [Figure 4-15] confirm that the Autoencoder failed to reconstruct complex patterns accurately, while the 1dTrans model suffered from overcorrection, often suppressing or removing meaningful spectral peaks. In contrast, BC-Former consistently outperformed the other models, achieving superior correction performance with high accuracy and robustness under noisy real-world conditions.

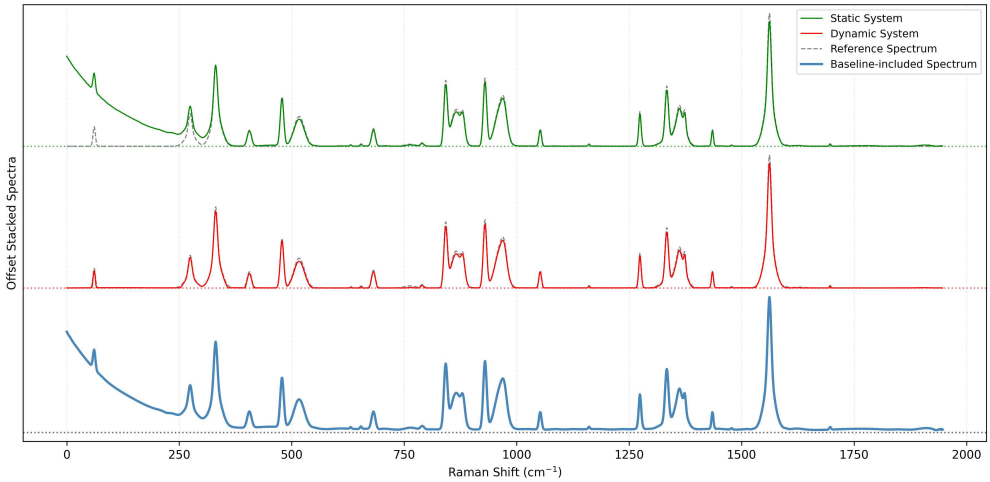
4.2.2.5 Evaluation of the Dynamic System's Adaptability to Unseen Baseline Patterns

In real-world Raman spectroscopy applications, various types of unseen data may be encountered, making it essential for a baseline correction system to effectively adapt to unfamiliar patterns. To address such scenarios, we further extend BC-Former into a dynamic adaptive system (hereafter referred to as the dynamic system) that can flexibly

adjust to previously unseen baseline patterns. The dynamic system performs the following steps when data containing previously unseen baselines is collected. First, an initial baseline is extracted from the unseen spectrum using a traditional algorithmic approach commonly adopted in baseline correction. Since ground truth baselines are unavailable for experimentally collected spectra, an approximate initial estimate is introduced to provide a practical starting point for modeling unseen patterns. Next, a large set of synthetic Raman spectra is generated using the initially estimated baseline, enabling the system to model the new baseline pattern. Prior to fine-tuning, the loss-weighting parameters are initialized from pre-trained values and updated jointly with the model, allowing adaptation while preserving balanced loss contributions. Lastly, only the embedding layer and the first Transformer encoder block are frozen, while the remaining encoder blocks and the decoder block are fine-tuned to enable rapid adaptation.

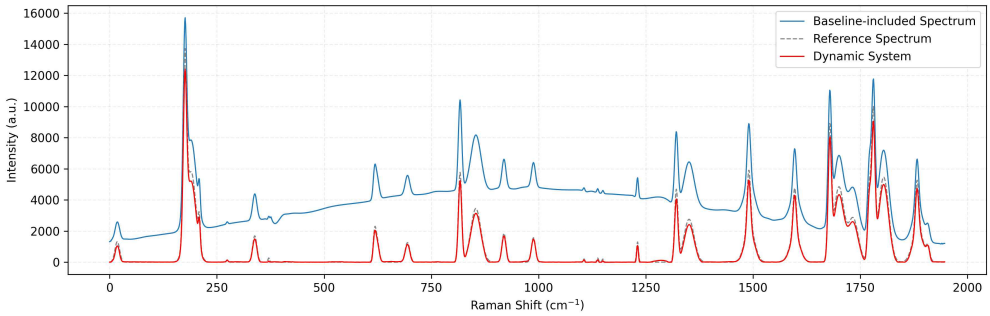
In the following experiment, we evaluate the adaptability of our dynamic system when presented with Raman spectra containing baseline types not seen during training. Specifically, we assess whether the system can rapidly adapt to these new patterns while maintaining correction performance. For this purpose, we compare two systems. The first is a static system trained only on Raman spectra containing predefined baseline types, and the second is the proposed dynamic system, which supports on-the-fly adaptation to newly collected unseen data. In this experiment, the static system was trained using synthetic spectra containing only Gaussian and polynomial baseline types. By contrast, the dynamic system further fine-tuned the pre-trained static model using synthetic spectra generated from a single unseen Raman spectrum with an exponential-type baseline. The remaining 9 spectra with the same exponential baseline type, which were unseen during

training, were used for evaluation.

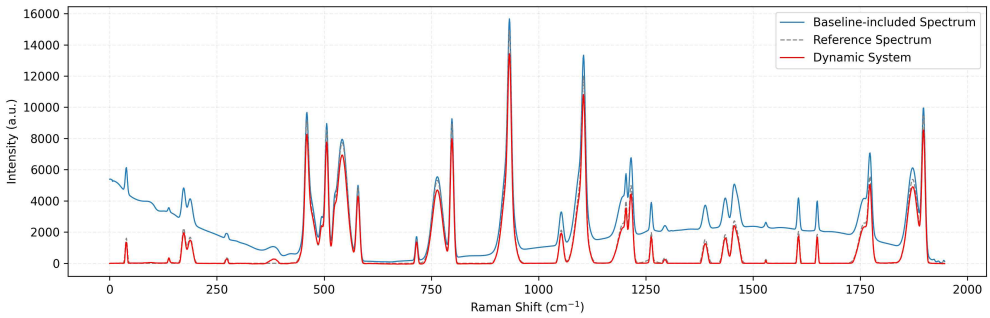


[Figure 4-16] Visual comparison of baseline correction results between the Static and Dynamic systems on unseen data

Based on this experimental setup, the evaluation results are presented in [Figure 4-16]. The static system was unable to remove the baseline in the 0-500 cm^{-1} region, whereas the dynamic system successfully removed the baseline and preserved the peak structures, closely matching the reference spectrum.



(a) Result of the dynamic system on data with a Gaussian-distributed baseline



(b) Result of the dynamic system on data with a Polynomial-distributed baseline

[Figure 4-17] Baseline correction results of the Dynamic system on data with Gaussian- and Polynomial-distributed baseline

[Figure 4-17] further shows that the dynamic system, even after being fine-tuned on exponential baseline data, retained its correction capability for Gaussian ([Figure 4-17] (a)) and polynomial ([Figure 4-17] (b)) baselines. This indicates that the model both adapted to new patterns and preserved previously acquired knowledge, effectively preventing catastrophic forgetting. In conclusion, this experiment confirms that the proposed dynamic system can promptly adapt to previously unseen baseline patterns while maintaining high-quality correction. Remarkably, the system achieved high performance by fine-tuning with synthetic data generated from only a single unseen spectrum, accurately removing the baseline while preserving the original peak shape and intensity. These results highlight the practical applicability of our dynamic system to real-world Raman spectroscopy, where diverse and unpredictable baseline conditions are often encountered.

Ⅴ. Conclusion

The motivation for this research stemmed from the critical challenges encountered in processing 1D signal data, which constitutes a fundamental data type across various scientific and industrial fields. In real-world scenarios, these signals are frequently compromised by background interference and non-stationary baseline drift, which significantly hinder accurate quantitative analysis. Previous methodologies often exhibited considerable sensitivity to noise and lacked the robust capacity to model intricate nonlinear baseline patterns, failing to fully capture the intrinsic complexity of the data. To effectively mitigate these artifacts and ensure high-fidelity signal reconstruction, this paper proposes robust baseline correction techniques through a two-step systematic approach. While Raman spectroscopy was employed as the primary validation domain to demonstrate the efficacy of these methods, the proposed frameworks are designed with the versatility to be applicable to a broader range of 1D signal processing tasks.

In the first approach, we proposed a three-stage baseline correction system, APON-BC, to address the limitations of the existing baseline correction method. The proposed system integrates three key components: synthetic data generation and balanced labeling to ensure balanced training, CNN-based parameter prediction to automate the tuning process, and robust baseline correction using the predicted parameter values. These innovations enable adaptive and highly accurate baseline correction, clearly distinguishing the proposed system from existing approaches. Through extensive experiments, we demonstrated that the proposed system outperforms conventional methods in spectral

baseline correction. In particular, the introduction of a balanced labeling strategy effectively mitigated data imbalance, further enhancing correction performance. By replacing the manual iterative parameter adjustment process with a neural network-based prediction mechanism, APON-BC achieved more stable and consistent corrections. Furthermore, computational comparisons with other neural network-based methods show that APON-BC requires significantly fewer parameters and lower FLOPs, highlighting its suitability for resource-constrained applications.

As the second approach, we propose BC-Former to establish a comprehensive, end-to-end deep learning framework, thereby culminating the structural evolution of first approach. Although APON-BC achieves strong automation and efficiency, it remains constrained by dependence on the target algorithm. This limitation suggests a clear path forward, which is transitioning to an end-to-end deep learning framework. Such a shift would enable direct signal reconstruction beyond the fixed mathematical constraints of traditional algorithms, allowing the system to fully exploit the global modeling capabilities of neural networks. To transcend this algorithmic dependency and unlock the full potential of data-driven modeling, BC-Former is designed as a purely end-to-end Transformer-based model capable of learning the optimal signal reconstruction mapping directly from the data. Through its self-attention architecture, the model effectively captured long-range dependencies in spectral data and modeled diverse spectral and nonlinear baseline patterns. To improve generalization, we constructed a realistic synthetic training dataset combining varied baseline shapes and bottom artifacts. In addition, the DLW strategy was introduced to balance multiple loss components during training, enhancing both precision and stability. As a result, BC-Former consistently outperformed conventional signal processing and neural network-based methods. We also proposed a

dynamic adaptive system incorporating BC-Former, which showed strong adaptability and robustness under realistic Raman spectroscopy scenarios.

The primary contributions of this paper are as follows. First, we resolved the efficiency-precision trade-off through a dual-method architecture: APON-BC for resource-constrained real-time processing and BC-Former for state-of-the-art accuracy in high-performance scenarios. Second, we demonstrated a methodological evolution from hybrid parameter-driven systems to unified end-to-end learning, establishing a complete pathway for advancing data-driven signal processing. Third, we enhanced automation and generalization through strategic data engineering innovations. By implementing balanced labeling, synthetic data generation, and DLW, both methods achieve robust performance without manual tuning. Finally, comprehensive benchmarks confirmed that both methods surpass existing approaches in their respective domains, establishing new standards for automated baseline correction.

In future work, we plan to design an integrated pipeline that combines baseline correction with downstream tasks such as peak quantification, classification, and clustering, aiming to achieve end-to-end automation and improved performance in spectral analysis. Additionally, we intend to explore advanced synthetic data generation techniques to further enhance realism and diversity, and develop lightweight model architectures to support real-time applications.

References

1. International References

Mohammadi Foumani, N., Miller, L., Tan, C. W., Webb, G. I., Forestier, G., & Salehi, M. (2024). Deep learning for time series classification and extrinsic regression: A current survey. ACM Computing Surveys, 56(9), 1-45.

Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., ... & Petitjean, F. (2020). Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery, 34(6), 1936-1962.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. Mechanical systems and signal processing, 151, 107398.

Liu, X., An, H., Cai, W., & Shao, X. (2024). Deep learning in spectral analysis: Modeling and imaging. TrAC Trends in Analytical Chemistry, 172, 117612.

Gloaguen, Y., Kirwan, J. A., & Beule, D. (2022). Deep learning-assisted peak curation for large-scale LC-MS metabolomics. Analytical chemistry, 94(12), 4930-4937.

Strodthoff, N., Wagner, P., Schaeffter, T., & Samek, W. (2020). Deep learning for ECG analysis: Benchmarks and insights from PTB-XL. IEEE journal of biomedical and health informatics, 25(5), 1519-1528.

Yogurtcu, B., Cebi, N., Koçer, A. T., & Erarslan, A. (2024). A Review of Non-Destructive Raman Spectroscopy and Chemometric Techniques in the Analysis of Cultural Heritage. Molecules,

29(22), 5324.

Chen, C., Qi, J., Li, Y., Li, D., Wu, L., Li, R., ... & Sun, N. (2024). Applications of Raman spectroscopy in the diagnosis and monitoring of neurodegenerative diseases. Frontiers in Neuroscience, 18, 1301107.

Orlando, A., Franceschini, F., Muscas, C., Pidkova, S., Bartoli, M., Rovere, M., & Tagliaferro, A. (2021). A comprehensive review on Raman spectroscopy applications. Chemosensors, 9(9), 262.

Hu, J., Chen, G. J., Xue, C., Liang, P., Xiang, Y., Zhang, C., ... & Shum, P. P. (2024). RSPSSL: A novel high-fidelity Raman spectral preprocessing scheme to enhance biomedical applications and chemical resolution visualization. Light: Science & Applications, 13(1), 52.

Fang, S., Wu, S., Chen, Z., He, C., Lin, L. L., & Ye, J. (2024). Recent progress and applications of Raman spectrum denoising algorithms in chemical and biological analyses: A review. TrAC Trends in Analytical Chemistry, 172, 117578.

Vulchi, R. T., Morgunov, V., Junjuri, R., & Bocklitz, T. (2024). Artifacts and anomalies in Raman spectroscopy: A review on origins and correction procedures. Molecules, 29(19), 4748.

Kashima, A., Urashima, S. H., & Yui, H. (2024). Analytical method for stable background reduction for Raman spectra of carbon-containing meteorite and terrestrial samples suffering from intense fluorescence. Meteoritics & Planetary Science, 59(2), 338-350.

Zhang, Z. M., Chen, S., & Liang, Y. Z. (2010). Baseline correction using adaptive iteratively reweighted penalized least squares. Analyst, 135(5), 1138-1146.

Baek, S. J., Park, A., Ahn, Y. J., & Choo, J. (2015). Baseline correction using asymmetrically reweighted penalized least squares

smoothing. Analyst, 140(1), 250−257.

Han, M., Dang, Y., & Han, J. (2024). Denoising and baseline correction methods for Raman spectroscopy based on convolutional autoencoder: a unified solution. Sensors, 24(10), 3161.

Liu, Y. (2021). Adversarial nets for baseline correction in spectra processing. Chemometrics and Intelligent Laboratory Systems, 213, 104317.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Zhao, J., Woznicki, T., & Kusnierek, K. (2025). Estimating baselines of Raman spectra based on transformer and manually annotated data. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, 330, 125679.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541−551.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time−scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30.

Berthold, M. R., & Höppner, F. (2016). On clustering time series using euclidean distance and pearson correlation. arXiv preprint arXiv:1601.02213.

Dong, Y., Sun, Z., & Jia, H. (2006). A cosine similarity−based negative selection algorithm for time series novelty detection. Mechanical Systems and Signal Processing, 20(6), 1461−1472.

Jirasek, A., Schulze, G., Yu, M. M. L., Blades, M. W., & Turner, R.

F. B. (2004). Accuracy and precision of manual baseline determination. Applied spectroscopy, 58(12), 1488-1499.

Lieber, C. A., & Mahadevan-Jansen, A. (2003). Automated method for subtraction of fluorescence from biological Raman spectra. Applied spectroscopy, 57(11), 1363-1367.

Eilers, P. H., & Boelens, H. F. (2005). Baseline correction with asymmetric least squares smoothing. Leiden University Medical Centre Report, 1(1), 5.

Ye, J., Tian, Z., Wei, H., & Li, Y. (2020). Baseline correction method based on improved asymmetrically reweighted penalized least squares for the Raman spectrum. Applied Optics, 59(34), 10933-10943.

Xu, D., Liu, S., Cai, Y., & Yang, C. (2019). Baseline correction method based on doubly reweighted penalized least squares. Applied optics, 58(14), 3913-3920.

Zhang, F., Tang, X., Tong, A., Wang, B., Wang, J., Lv, Y., ... & Wang, J. (2020). Baseline correction for infrared spectra using adaptive smoothness parameter penalized least squares method. Spectroscopy Letters, 53(3), 222-233.

Liu, J., Sun, J., Huang, X., Li, G., & Liu, B. (2015). Goldindec: a novel algorithm for Raman spectrum baseline correction. Applied spectroscopy, 69(7), 834-842.

Han, Q., Peng, S., Xie, Q., Wu, Y., & Zhang, G. (2018, July). Iterative reweighted quantile regression using augmented Lagrangian optimization for baseline correction. In 2018 5th International Conference on Information Science and Control Engineering (ICISCE) (pp. 280-284). IEEE.

Park, A. R., Park, J. K., Ko, D. Y., Kim, S. G., & Baek, S. J. (2019). Decision function for optimal smoothing parameter of

asymmetrically reweighted penalized least squares. Journal of the Korea Academia-Industrial cooperation Society, 20(3), 500-506.

Zhang, F., Tang, X., Tong, A., Wang, B., & Wang, J. (2020). An automatic baseline correction method based on the penalized least squares method. Sensors, 20(7), 2015.

Chen, T., Son, Y., Park, A., & Baek, S. J. (2022). Baseline correction using a deep-learning model combining ResNet and UNet. Analyst, 147(19), 4285-4292.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Cham: Springer international publishing.

Diakogiannis, F. I., Waldner, F., Caccetta, P., & Wu, C. (2020). ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. ISPRS Journal of Photogrammetry and Remote Sensing, 162, 94-114.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & Van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature methods, 17(3), 261-272.

Jiao, Q., Guo, X., Liu, M., Kong, L., Hui, M., Dong, L., & Zhao, Y. (2023). Deep learning baseline correction method via multi-scale analysis and regression. Chemometrics and Intelligent Laboratory Systems, 235, 104779.

Mozaffari, M. H., & Tay, L. L. (2021, December). Convolutional

Neural Networks for Raman spectral analysis of chemical mixtures. In 2021 5th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI) (pp. 1-6). IEEE.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. Computing in science & engineering, 13(2), 22-30.

McKinney, W. (2010). Data structures for statistical computing in Python. scipy, 445(1), 51-56.

Adam, K. D. B. J. (2014). A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 1412(6).

Xu, Y., Du, P., Senger, R., Robertson, J., & Pirkle, J. L. (2021). ISREA: an efficient peak-preserving baseline correction algorithm for Raman spectra. Applied Spectroscopy, 75(1), 34-45.

Gebrekidan, M. T., Knipfer, C., & Braeuer, A. S. (2021). Refinement of spectra using a deep neural network: Fully automated removal of noise and background. Journal of Raman Spectroscopy, 52(3), 723-736.

Bayko, D. P., & Lucas, P. (2023). Structural analysis and chemical stability of Ge and As telluride glasses by Raman spectroscopy. Journal of Non-Crystalline Solids: X, 18, 100186.

Wilczynski, K., Gertych, A. P., & Zdrojek, M. (2023). Explaining Mysterious “Shoulder” Raman Band in TiS2 by Temperature-dependent Anharmonicity and Defects. The Journal of Physical Chemistry C, 127(42), 20870-20880.

Coca-Lopez, N. (2024). An intuitive approach for spike removal in

Raman spectra based on peaks' prominence and width. Analytica Chimica Acta.

Mozaffari, M. H., & Tay, L. L. (2021). Raman spectral analysis of mixtures with one-dimensional convolutional neural network. arXiv preprint arXiv:2106.05316.

Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation coefficients: appropriate use and interpretation. Anesthesia & analgesia, 126(5), 1763-1768.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620.

Lin, J. (2002). Divergence measures based on the Shannon entropy. IEEE Transactions on Information theory, 37(1), 145-151.

Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7482-7491).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge university press.

Georgiev, D., Pedersen, S. V., Xie, R., Fernández-Galiana, Á., Stevens, M. M., & Barahona, M. (2024). RamanSPy: An

open-source Python package for integrative Raman spectroscopy data analysis. Analytical Chemistry, 96(21), 8492-8500.

Kuester, J., Gross, W., & Middelmann, W. (2021). 1D-convolutional autoencoder based hyperspectral data compression. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, 15-21.

국 문 초 록

신경망 기반 자동화 접근을 통한 1차원 신호 데이터 처리 시스템

한 성 대 학 교 대 학 원
A I 응 용 학 과
A I 응 용 학 전 공
김 예 란

1차원 신호 데이터는 분광학, 화학 분석, 생체 신호 모니터링 등 광범위한 과학 및 산업 분야에서 필수적인 데이터 유형이다. 그러나 다양한 도메인의 1차원 신호 측정 과정에서 공통적으로 발생하는 비선형적인 기저선 변동(baseline drift)과 배경 잡음(background noise)은 데이터의 정량적 해석을 어렵게 만드는 주요 요인이다. 기존 기법들은 수동 튜닝에 의존하거나 일반화 성능이 부족하다는 한계가 있어, 본 연구는 이를 극복하기 위한 범용적인 자동화 처리 시스템을 제안한다. 또한, 제안된 시스템을 실증적으로 검증하기 위해 대표적인 1차원 신호인 라만 스펙트럼(Raman spectrum) 데이터를 통해 그 유효성을 검증한다.

본 논문은 두 가지 상호 보완적인 접근법을 제시한다. 첫째, 연산 효율성을 위해 제안된 APON-BC(Automated Parameter Optimization Network for Baseline Correction)는 신경망을 통해 최적의 알고리즘 매개변수를 자동으로 예측하는 하이브리드 시스템으로, 리소스가 제한된 환

경에서 신속한 처리를 가능하게 한다. 실험 결과, APON-BC는 0.002 GFLOPs의 낮은 연산량으로 기존 기법 대비 FID와 ED를 각각 84.71%, 64.65% 감소시키며 탁월한 효율성과 안정성을 입증하였다. 둘째, 정밀한 복원을 위해 제안된 BC-Former(Baseline Correction with Transformer)는 트랜스포머(Transformer) 기반의 End-to-End 딥러닝 모델로, 현실적인 합성 데이터와 동적 손실 가중치 전략을 통해 복잡한 비선형 기저선을 효과적으로 보정한다. BC-Former는 CS와 PCC가 0.998에 도달하고 FID와 ED를 각각 0.037, 0.286으로 최소화하는 등 타의 추종을 불허하는 정밀도와 보지 못한(unseen) 데이터에 대한 적응력을 보였다.

결론적으로, 본 연구는 효율성과 정밀도 간의 상충 관계를 해결할 뿐만 아니라, 리소스가 제한된 임베디드 장치부터 고성능 컴퓨팅 환경까지 아우르는 폭넓은 운영 환경에 최적화된 이원화된 솔루션을 제공한다. 또한, 하이브리드 접근 방식에서 완전한 End-to-End 딥러닝 프레임워크로의 방법론적 진보를 제시함으로써, 다양한 1차원 신호 데이터 분석의 완전 자동화와 신뢰성을 확립하는 새로운 표준을 마련하였다.

[주요어] 1차원 신호 처리, 기저선 보정, 트랜스포머, 매개변수 최적화, 합성 데이터 생성, 동적 손실 가중치