Spectrum-BERT: Pretraining of Deep Bidirectional Transformers for Spectral Classification of Chinese Liquors

Yansong Wang®, Yundong Sun®, Yansheng Fu®, Dongjie Zhu®, and Zhaoshuo Tian®, Member, IEEE

Abstract—Counterfeit Chinese liquor incidents in China have significantly disrupted market order and jeopardized the health of consumers. Currently, deep learning-based spectral detection techniques are extensively leveraged in noninvasive food inspection. Excessive reliance on labels severely limits its application in real scenarios. To make better use of limited samples, we are the first to use the “unsupervised pretraining & supervised fine-tuning” paradigm in combining the Transformer architecture for feature extraction and classification of the Chinese liquor spectrum, and propose Spectrum-BERT, which represents Bidirectional Encoder Representations from Transformers for Spectrum. Specifically, we creatively propose spectral curve partitioning and 1-D convolutional layer mapping to maintain the model’s sensitivity to characteristic peak locations and local information of spectral curves. Moreover, the paradigm of “unsupervised pretraining & supervised fine-tuning” addresses the limitation of label deficiency, thus improving the model’s applicability. Finally, we have conducted extensive experiments on the real liquor spectral dataset. Comparative experiments demonstrate that Spectrum-BERT outperforms all baselines on all metrics using only 70% of supervised signal. The limit experimental results show that Spectrum-BERT can still maintain its lead using only the 10% supervised signal. Thanks to the more efficient model architecture, Spectrum-BERT’s model parameters and FLOPs are only 1.568 and 1.322 of those of baselines.

Index Terms—BERT, deep learning, liquor detection, pretraining, spectral detection.

Manuscript received 31 August 2023; revised 20 December 2023; accepted 9 February 2024. Date of publication 7 March 2024; date of current version 6 May 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFE0201500 and in part by the Fundamental Research Funds for the Central Universities under Grant HIT.NSRIF.201714. The Associate Editor coordinating the review process was Dr. Sabrina Grassini. (Yansong Wang and Yundong Sun contributed equally to this work.) (Corresponding author: Dongjie Zhu.)

Yansong Wang and Dongjie Zhu are with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: yansongwang0629@163.com; zhudongjie@hit.edu.cn).

Yundong Sun is with the Department of Electronic Science and Technology, Harbin Institute of Technology, Harbin 150001, China, and also with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: hitffmy@163.com).

Yansheng Fu is with the School of Instrument Science and Engineering, Harbin Institute of Technology, Harbin 150001, China, and also with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: fuyansheng123@gmail.com).

Dongjie Zhu is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: zhudongjie@hit.edu.cn).

Zhaoshuo Tian is with the Institute of Marine Optoelectronic Equipment, Harbin Institute of Technology, Weihai 264209, China (e-mail: tianzhaoshuo@126.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIM.2024.3374300, provided by the authors. Digital Object Identifier 10.1109/TIM.2024.3374300

I. INTRODUCTION

At present, food safety incidents occur frequently all over the world, which has seriously damaged the normal market operation order, violated the rights and interests of regular manufacturers and consumers, and even caused serious damage to the health of consumers [1]. According to a survey by the World Health Organization, unsafe food can cause 600 million illnesses and 50,000 deaths worldwide each year.1 Because of the low cost and high profit of Chinese liquor counterfeiting, it has become one of the most significant food safety issues.

Traditional detection methods, such as chromatography [2], [3] and electronic nose [4], [5], are relatively demanding on the environment, and are easily affected by the temperature and chemical substances of the detected objects, leading to inaccurate results. Meanwhile, the detection process is complicated and costly. Spectral detection technology is widely used in food detection due to its advantages of simple operation, high sensitivity, long life, low cost, and no pollution to the environment and samples [6], [7]. Fig. 1 shows the spectral curves of different Chinese liquors. Differences in the composition and content of various Chinese liquors result in distinct spectral distributions. Based on these spectral differences, different Chinese liquor detection algorithms can be developed.

Traditional matching algorithms [8], [9], [10], based on spectral curve characteristic peak matching, dominate the existing food spectrum detection. However, those methods require manual extraction, setting, and adjustment of characteristic peaks based on domain knowledge, without the ability to learn and extract features automatically. At the same time, the detection target must be compared with a large number of samples in the database, which is also time-consuming and cannot be applied in massive data scenarios. The deep learning algorithm can realize complex feature extraction and predictive analysis tasks without manual feature extraction, and it has better generalization ability than artificially designed features. In recent years, the combination of spectral detection technology and deep learning algorithms has significantly improved the performance of food detection.

However, supervised deep learning models for spectral detection are limited in their applicability to scenarios because of the label deficiency. On the one hand, models for detecting
Chinese liquor cannot be trained using open-source large-scale datasets due to the unavailability of publicly accessible Chinese liquor spectral data. On the other hand, there are considerable brands and types of existing liquors in the market, and the price is high. It is difficult for the researchers to build and label the spectrum dataset of liquors, and the quality of the samples cannot be guaranteed. It should be noted that to ensure the correctness of the labels of liquor samples, researchers need to use other traditional auxiliary testing methods to verify the authenticity of samples. This step requires a high experimental cost and is time-consuming. In conclusion, there is not enough high-quality data in practical application scenarios to support training efficient deep learning models [11], i.e., the label deficiency problem. As the most effective scheme for label deficiency, the “unsupervised pretraining & supervised fine-tuning” paradigm has achieved great success in natural language processing (NLP) and computer vision (CV) [12], [13], [14], [15], but no researchers have tried it in spectral detection.

Many researchers [16], [17], [18] have improved the accuracy of spectral detection using convolutional neural networks (CNNs). However, the previous study [19] has also shown that the max pooling operation used in CNNs may cause the model to lose local information during downsampling. The spatial translation invariance of CNN also makes the captured spectral curve features incapable of sensitivity to the position of characteristic peaks. Therefore, we need a new curve feature extraction approach that can accurately capture the local features of the spectrum while maintaining great sensitivity to the position of the characteristic peaks.

To solve the aforementioned problems, in this article, we propose the Spectrum-BERT model inspired by the pretraining idea of the BERT model [20] in the NLP field. Specifically, based on the Self-Attention mechanism in BERT, we creatively propose spectral curve partitioning and 1-D convolutional layer mapping to maintain the model’s sensitivity to characteristic peak locations and local information of spectral curves. Similar to the training method of BERT, the training of Spectrum-BERT is also divided into two stages: pretraining and fine-tuning. In the pretraining stage, we elaborately design two pretraining tasks, next curve prediction (NCP) and masked curve model (MCM), so that the model can effectively utilize unlabeled samples to capture the potential knowledge of spectral data, breaking the restrictions of the insufficient labeled samples.

To the best of our knowledge, this article is the first attempt in the field to use the Transformer model architecture and the “unsupervised pretraining & supervised fine-tuning” paradigm for feature extraction and classification of the Chinese liquor spectrum. The contributions of this article can be summarized as the following three points.

1) The Novel “Unsupervised Pretraining & Supervised Fine-Tuning” Paradigm: We apply the “unsupervised pretraining & supervised fine-tuning” paradigm for feature extraction and classification of the Chinese liquor spectrum for the first time. The advanced paradigm allows Spectrum-BERT to effectively utilize unlabeled samples to capture the potential knowledge of spectral data, breaking the restrictions of the insufficient labeled samples, and improving the applicability and performance of the model in practical scenarios.

2) The Elaborately Designed Model Architecture: To retain the model’s sensitivity to the characteristic peak position and local information of the spectral curve, we first attempt in the field to capture Chinese liquor spectrum features using the Transformer model architecture and the Self-Attention mechanism. The creatively proposed spectral curve partitioning and 1-D convolutional layer mapping make the application of the Self-Attention mechanism to spectral curves possible.

3) The Extensive Experimental Verification and Analysis: We have conducted extensive experiments on the real liquor spectral dataset. Comparative experiments demonstrate that Spectrum-BERT outperforms all baselines on all metrics using only 70% of supervised signal. Even when the supervised signals drop to 10%, Spectrum-BERT continues to maintain its lead. With the help of the Transformer architecture, Spectrum-BERT’s model parameters and FLOPs are only 1/568 and 1/322 of those of baselines.

II. RELATED WORK

In this section, we will introduce the research work related to this article, including spectral techniques for substance identification and the techniques for label deficiency.

A. Spectral Techniques for Substance Identification

Some researchers [17], [18], [21], [22] combine spectral techniques with traditional machine learning for food identification and classification. Hu et al. [21] apply the Synthetic Minority Over-sampling Technique (SMOTE) to cope with the imbalance of the wine data and use three traditional machine learning models as classifiers. To quickly determine the chemical components in milk, Sheng et al. [22] choose the gradient-boosted regression tree (GBRT) for the NIR absorption multispectra of milk after completing the comparison of eight traditional machine learning models.

Some studies [16], [17], [18] leverage deep CNN as a solution for spectral recognition. As an illustrative instance,
Fig. 2. Schematic representation of the shortcomings of using CNN to extract spectral curve features [19]. (a) Spectral curves with similar intensity characteristic peaks at different positions. (b) Spectral curves with similar characteristic peaks but mismatched details. The red curve represents the standard sample, and the black curve represents the sample to be detected.

Fig. 3. Samples of spectral curves in the dataset that could be misclassified by the CNN. Letter A–I represents some of the characteristic peaks in the samples. (a) Sample @1 and sample @2 with different labels have characteristic peaks with very similar morphologies at different pixel positions, i.e., A versus C and B versus D. (b) Sample @3 and sample @4 with different labels have characteristic peaks with very similar morphologies at the same pixel position, e.g., E versus G and F versus H, but some of the local characteristic peaks of @4 cannot correspond to @3, e.g., I.

Zhu et al. [18] use 3-D CNN and 2-D CNN to classify the near-infrared spectrum of candy and salmon filet muscle tissue, respectively, with good results. However, the previous study [19] shows that the max pooling operation used in CNN may cause the model to lose local information during downsampling. The spatial translation invariance of CNN also makes the captured spectral curve features incapable of sensitivity to the position of characteristic peaks. Specifically, spectral curves with similar intensity characteristic peaks at different positions [as shown in Figs. 2(a) and 3(a)] or spectral curves with similar characteristic peaks but partially mismatched details [as shown in Figs. 2(b) and 3(b)] may be misjudged as similar curves by CNN. Yan et al.’s [17] experiments which employ deep CNN and auto-CNN as classifiers confirm this phenomenon. Due to CNN’s inherent defects in extracting spectral curve features [17], [19], it is unsuitable as a primary model for spectral curve feature extraction.

Considering the mentioned problems, it is necessary to develop a deep learning model that can robustly capture features and adapt to limited labels during training.

B. Techniques for Label Deficiency

At present, solving the contradiction between deep learning model training requirements and label deficiency has become a research hotspot. As the most effective scheme for label deficiency, the “unsupervised pretraining & supervised fine-tuning” paradigm achieves great success in NLP and CV [20], [23], [24], [25], [26], [27], [28].

In the field of NLP, Devlin et al. [20] propose BERT as a classical model in the field, which utilizes the deep bidirectional Transformer encoder structure and performs significantly well in many typical NLP tasks. Several researchers [27], [28], [29] promote the performance of deep bidirectional Transformer encoder structure-based language models by improving pretraining tasks or proposing new ones. Furthermore, Dosovitskiy et al. [26] creatively apply the Transformer architecture to images for the first time based on supervised learning ideas and propose the Vision Transformer (ViT). Inspired by ViT, researchers [12], [24] break new records for several typical tasks in CV by improving the model training paradigm and optimizing image feature extraction.

The widespread application and outstanding performance of pretraining in the field of NLP and CV prove that the acquisition of general domain knowledge through pretraining can boost the model performance.
Meta-learning models can improve learning efficiency in scenarios with limited labeled samples by learning the optimal convergence process (meta-tasks) of traditional machine learning models on a sufficient number of samples. This provides a solution to the problem of label deficiency. While meta-learning does not place restrictions on whether a meta-task is supervised or not, most meta-tasks in the field of meta-learning currently fall into the category of supervised learning. This status restricts the application of meta-learning methods in the field of Chinese liquor spectral classification.

Semisupervised learning\(^2\) is also a technique that can address the issue of label deficiency by employing a combination of labeled and unlabeled samples during training. The SSDL, a semisupervised autoencoder based on the idea of multitask training, developed by Said et al. [11] is successfully applied to identify milk in the field of food identification. Furthermore, the Spectrum-BERT proposed in this article utilizes a substantial amount of unlabeled data in unsupervised pretraining and a limited amount of labeled data in supervised fine-tuning, respectively. Thus, Spectrum-BERT is also classified as a semisupervised model.

III. Spectrum-BERT: Pretraining of Deep Bidirectional Transformers for Spectral Classification of Chinese Liquors

This chapter will provide a detailed introduction to Spectrum-BERT and its training paradigm. Specifically, Section III-A will introduce the overall architecture of Spectrum-BERT in detail. In Section III-B, the key design of Spectrum-BERT that makes it sensitive to the characteristic peak position and local information of the spectral curve will be described. Section III-C introduces the Self-Attention mechanism, which is the most crucial module of Spectrum-BERT. Sections III-D and III-E introduce the “unsupervised pretraining & supervised fine-tuning” paradigm. This paradigm is the cornerstone of Spectrum-BERT to mitigate the label deficiency problem.

A. Model Architecture

As shown in Fig. 4, Spectrum-BERT comprises three parts: the input layer (Embedding Layer), the model layer (Transformer), and the downstream task layer. Prior to feeding into the Embedding Layer, the completion of spectral curve partitioning and input mask selection is necessary (as shown at the bottom of Fig. 4). The Embedding Layer of Spectrum-BERT performs 1-D convolution-based feature projecting, sample segmentation labeling, and addition of position embeddings (as shown in Fig. 5). Spectral curve partitioning and 1-D convolutional layer mapping are two creative improvements of Spectrum-BERT compared to BERT. The calculation method and architecture of the input layer will be introduced detailedly in Section III-B. Subsequently, Spectrum-BERT uses the Transformer architecture model layer to capture curve block association relationships, the global-scale feature, and the representation of latent knowledge. Moreover, Spectrum-BERT, with a distinct architecture, overcomes CNN’s challenge in extracting spectral curve characteristics.

The Spectrum-BERT size is mainly defined by the following three parameters: the number of layers \(L\) (i.e., the number of stacked Transformer blocks in the model layer), the hidden layer size \(H\) (i.e., the dimension of each token block output by Spectrum-BERT), and the number of Self-Attention heads \(A\). As shown at the top of Fig. 4, the primary role of the downstream task layer is to utilize the feature representation produced by the model layer to perform specific tasks (Sections III-C and III-D). Specifically, Spectrum-BERT takes our well-designed NCP task and MCM task as training objectives in the pretraining stage, and uses a multiclassifier built with fully connected layers for the multiclassification task in the fine-tuning stage.

B. Spectral Curve Partition and Input Layer

In contrast to the discrete, independent, and countable words found in natural language sequences, spectral curves exhibit a continuous pattern of composition. We cannot process the spectral curves using methods similar to the pretrained word embedding models [20] in NLP to obtain Token Embeddings. Therefore, to retain the model’s sensitivity to the characteristic peak position and local information of the spectral curve, we innovatively partition the spectral curve into multiple blocks. The hyperparameter of token_size determines the length of the block. Then 1-D convolution is used to map the block (we call it token) into Token Embeddings with length \(H\) (the preset hidden dimension). Specifically, we define the feature projecting process using

\[
E_{\text{Token}_i} = \text{trans} \left( \text{Token}_i \right) = \text{Cov1d} \left( \text{Token}_i \right)
\]  

\(^2\)https://en.wikipedia.org/wiki/Weak_supervision#Semi-supervised_learning
where Token is the i-th token in the sequence, $E_{\text{Token}}$ is the Token Embeddings corresponding to the i-th token, and $i \in [1, A_n + B_n]$. $A_n(B_n)$ is the number of curve $A(B)$ tokens. Specifically, the original spectral curve feed to the input layer will first be partitioned into token_size-dimensional tokens, and a 1-D convolutional layer converts token_size-dimensional tokens into $H$-dimensional Token Embeddings. Through the aforementioned operation, the continuous curve is transformed into discrete tokens, which is acceptable to the Self-Attention mechanism. The 1-D convolutional layer mapping serves to control the model complexity and unify the feature space. To better retain the features contained, we will determine a reasonable token_size for spectral curve partitioning through experiments. This will reduce the likelihood of characteristic peak fragmentation or over-concentration.

To distinguish pairs of input curves, we utilize two methods: special marks and Segment Embedding. First, we insert the [SEP] token as a delimiter between the two curves and at the end of the entire token sequence (only the latter is performed for single-curve inputs). Next, we attach learnable Segment Embedding to each token to indicate that the current token comes from either curve A or curve B. As shown in Fig. 5, we use $E_A$ to represent that the current token comes from curve A. For the design of Position Embeddings, we follow the idea of Vaswani et al. [23] and leverage the linear transformation of sin and cos to add position information to each spectral curve block

$$E_{\text{PE}}^d = \begin{cases} \sin \left( \frac{\text{pos}}{10000^{\frac{k}{d}}} \right), & d = 2k \\ \cos \left( \frac{\text{pos}}{10000^{\frac{k}{d}}} \right), & d = 2k + 1 \end{cases} \quad (2)$$

where $E_{\text{PE}}^d$ is the k-th dimension in Position Embeddings corresponding to the i-th token, pos $\in [0, \text{max_seq_length})$ is the position of the token in the token sequence, $k \in [0, H/2)$ denotes the k-th dimension in Position Embedding, and max_seq_length is the maximum length of the token sequence. The Position Embeddings is independent for each dimension of Token Embedding in the token sequence, making the Spectrum-BERT characteristic peak position sensitive. Finally, we sum the three parts (Token Embeddings, Segment Embeddings, and Position Embeddings) corresponding to the same token block to get the final input representation sequence

$$E_{\text{Input}} = \begin{cases} E_{\text{Token}} + E_A + E_{\text{PE}}, & \text{if Token}_i \text{ from } A \\ E_{\text{Token}} + E_B + E_{\text{PE}}, & \text{if Token}_i \text{ from } B \end{cases} \quad (3)$$

where $E_{\text{Input}}$ and $E_{\text{PE}}$, respectively, denote the final input representation and Position Embedding of the i-th token in the input token sequence.

It should be noted that the beginning of each token sequence processed by the input layer is a [CLS] mark. Meanwhile, the output $C \in \mathbb{R}^H$ corresponding to this token represents the aggregated representation of the entire sequence of input tokens. Similarly, we define the feature representation corresponding to the i-th curve block behind the [CLS] mark as $T_i \in \mathbb{R}^H$. Therefore, the length $n$ of the input token sequence is the sum of $A_n + B_n$ and the special tokens such as [SEP] and [CLS]. We stack the sequence of final input representations $E_{\text{Input}}$ in the order of the inputs as a matrix $E \in \mathbb{R}^{n \times H}$.

C. Transformer and Self-Attention

The core of Spectrum-BERT consists of multiple Transformer layers based on the Self-Attention mechanism. Unlike the original design of the Transformer [23], Spectrum-BERT uses only the encoder portion of the Transformer for spectral curve feature encoding. The use of the Self-Attention mechanism is crucial for Spectrum-BERT to accurately capture intricate associability between input token sequences. Meanwhile, the Self-Attention mechanism does not incorporate the max pooling operation, thus avoiding the loss of local information. To capture complex semantic information more easily, in practice, researchers usually extend the Self-Attention mechanism into a Multi-head Self-Attention (MSA) mechanism. For simplicity of description, we describe the computation of the Self-Attention mechanism by using the single-head Self-Attention mechanism as an example. We first use the learnable mapping matrices $W^Q \in \mathbb{R}^{H \times H_K}$, $W^K \in \mathbb{R}^{H \times H_K}$, and $W^V \in \mathbb{R}^{H \times H_V}$ to project the feature matrix $E$ to three independent feature spaces, namely $Q$, $K$, and $V$

$$Q = EW^Q, \quad K = EW^K, \quad V = EW^V \quad (4)$$

where $H_K$ is the dimension of the feature spaces $Q$ and $K$, and $H_V$ is the dimension of the feature space $V$. Next, we use the inner product operation between the feature spaces $Q$ and $K^\top$ to obtain the association of each token with others, i.e., the attention weights. Finally, we use row-wise softmax to
normalize the attention weights and then multiply with the feature space \( V \) to acquire the updated feature representation \( E' \)

\[
E' = \text{softmax} \left( \frac{Q K^T}{\sqrt{H_K}} \right) V.
\]  

After the multilayer Transformer layer MSA computation, the feature representation \( E' \) becomes the token representation described in Section III-B, i.e., \( T_i \) and \( C \).

**D. Pretraining Spectrum-BERT**

In the pretraining stage, we design the following two pretraining tasks.

1) **Task #1: Masked Curve Model:** We expect that the model can infer or predict the unknown missing content based on the known context like a human, but simply connecting two unidirectional models cannot reach the expected goal. Therefore, we design the MCM pretraining task. Specifically, each token in the input sequence of tokens processed by the input layer will be masked with a 15% probability. The feature representation corresponding to the masked token produced by the model layer needs to be as similar as possible to the Token Embedding before being masked. We define the loss function \( L_{\text{MCM}} \) for the MCM task as follows:

\[
L_{\text{MCM}} = \frac{1}{|S|} \sum_{i \in S} (T_i - E_{\text{inpt}})^2
\]  

where \( S \) is the serial number set of the masked token from the curve pair. In the MCM task, we expect the output of the Spectrum-BERT to approximate the masked token as closely as possible, rather than reconstructing the entire input token sequence. This measure avoids the problem of information leakage in traditional deep bidirectional learning and enables the model to predict missing tokens from two directions (before and after the current position) of the curve, thereby capturing the composition patterns of the input token sequence.

It should be noted that the token masking strategy implemented in the MCM task is in line with that of BERT, meaning that it is unnecessary to substitute all masked tokens with [MASK]. If each selected token is masked with a fixed probability in the pretraining stage, the model will completely lose the ability to perform prediction and reasoning based on the semantics of the masked token, which is detrimental to the feature extraction and semantic expression capabilities of the model. We expect that Spectrum-BERT can consider the semantics of the masked token while attempting to learn the association information around and away from the masked token, so that Spectrum-BERT can model the full compositional patterns of the curve.

2) **Task #2: Next Curve Prediction:** In the field of NLP, we expect a good language model to capture the relationship between sentences while performing language modeling. Similarly, we anticipate that Spectrum-BERT will capture the correlation between different curves while modeling the characteristics of the spectral curves. Therefore, we propose the NCP task.

In the NCP pretraining task, we take the spectral curve pair as the input of Spectrum-BERT, and the aggregate representation \( C \in \mathbb{R}^H \) corresponding to [CLS] is used as the input of the NCP task in the downstream task layer. In the downstream task layer, we employ the softmax function and linear layer for binary classification tasks

\[
\hat{y} = \text{softmax}(\text{Linear}(C))
\]  

where \( \hat{y} \) represents the result of the binary classification task. In the experiment, we adopt the method of random combination to obtain curve pairs. Specifically, there is a 50% probability that spectral curves \( A \) and \( B \) belong to the same liquor (the label is 1), and there is a 50% probability that spectral curves \( A \) and \( B \) belong to different liquors (the label is 0). We define the loss function \( L_{\text{NCP}} \) for the NCP task as follows:

\[
L_{\text{NCP}} = -\frac{1}{B} \sum_{i=1}^{B} \log(\hat{y}_i), \quad y_i \in \{0, 1\}
\]  

where \( B \) is the batch size, \( y_i \) is the label of the \( i \)th sample, and \( \hat{y}_i \) is the probability that the model predicts the \( i \)th sample as belonging to class \( j \). The final pretraining loss function \( L_{\text{NCP+MCM}} \) for Spectrum-BERT is expressed as follows:

\[
L_{\text{NCP+MCM}} = L_{\text{NCP}} + L_{\text{MCM}}.
\]

**E. Fine-Tuning Spectrum-BERT**

The primary purpose of the fine-tuning stage is to adjust the model parameters obtained from the pretraining stage to better match different downstream tasks. Due to the universality of the structure of Spectrum-BERT, without changing the model architecture, it is only necessary to replace the downstream task layer to complete specific tasks in various scenarios (classification, matching, clustering tasks, etc.)

The multiclassification task of the Chinese liquor spectral curve is used as the fine-tuning object in this study. To narrow the training gap between the fine-tuning and pretraining stages, we keep the processing of model inputs in the fine-tuning stage consistent with the pretraining stage. We explore two different pretraining tasks in the fine-tuning stage, namely spectral curve classification based on [CLS] token (Fine-tuning-CLS) and spectral curve classification based on all tokens (Fine-tuning-All). In Fine-tuning-CLS, we take the aggregate representation \( C \in \mathbb{R}^H \) corresponding to the [CLS] token as the input of the classifier (this is also the scheme of BERT). In Fine-tuning-All, we take the entire feature representation sequence \( T_i \in \mathbb{R}^H \) as the input of the classifier. After experimental verification, we found that the performance of Fine-tuning-CLS is unsatisfactory. Therefore, in the experiment section, we prefer Fine-tuning-All.

**IV. Experiment**

This section first provides an overview of the experiment, including the instrumentation, datasets, definition of baselines, and other relevant information. Second, we demonstrate the contribution of this article through the comparative experiment conducted between Spectrum-BERT and baselines.
Furthermore, the effectiveness of Spectrum-BERT is validated through limit, imbalance, and ablation experiments. Finally, we conduct studies on some important hyperparameters such as token_size, number of layers ($L$), Self-Attention heads ($A$), and hidden layer dimension ($H$).

A. Experiment Instrument

The experimental system, which is our self-developed system, is shown in Fig. 6(a). The wavelength of the laser diode is 405 nm and we can switch the output mode between continuous and pulse as required. The laser output power of the spectrometer ranges from 8 to 100 mW. The charge coupled device (CCD) pixel is $1280 \times 720$, and the CPU chip is Atheros AR9331 SOC. The internal structure of the experimental system is shown in Fig. 6(b), which includes a laser spectrometer and a smart terminal. The laser spectrometer consists of a CPU, a spectroscopic system, convex lenses, and a battery. The CPU is respectively connected to the laser emission module, the CCD, and the wireless routing module. The laser emitted from the laser emission module excites the tested liquor through the wall of the bottle, and the optical signal generated by the spectroscopic system is then concentrated on the surface of the CCD by the convex lens. The CCD can convert the optical signal into an electrical signal, the CPU is responsible for receiving the control instructions of the smart terminal and controlling the laser emission module to emit laser with preset laser intensity, and then converting the electrical signal transmitted by the CCD into a digital image. The wireless router module undertakes the task of wireless communication between the smart terminal and the microspectrometer. When conducting experiments, the liquor samples are put in a clear glass bottle and placed in the shading box for testing. The smart terminal receives spectral data through the wireless router.

B. Dataset

The tested liquors are selected from 12 brands in the Chinese market, including 3 Maotai-flavor liquors, 3 Luzhou-flavor liquors, and 6 Fen-flavor liquors. To better simulate the objective factors that may affect the liquor sample in the application scenario (e.g., liquor storage time and conditions and spectrometer errors [30]), we group multiple samples of liquor from different sources but the same brand into the same category. We conduct 100 measurements for all liquor samples of the same brand, at intervals of at least one hour. In other words, we have collected 100 spectral samples with identical labels at different times, which are derived from distinct sources of the same brand. The laser intensity of the instrument is calibrated using pure water. After calibration, the laser output power of the spectrometer is 79 mW. Background deduction is achieved by subtracting the two results from the laser on/off conditions. To ensure the stability of the laser and the data transmission, the two conditions are kept for 100 ms, respectively. Each measurement represents the average of five determinations. The data are processed using min–max normalization to eliminate the effect of dimension on the calculation.

UV spectrum and NIR spectroscopy are becoming active research areas due to their simple pretreatment process. We select fluorescence and Raman spectra generated by laser excitation of the tested liquor as our samples. Compared to NIR spectroscopy, Raman spectroscopy exhibits clear, sharp characteristic peaks with specific physical significance for reflecting component information. Furthermore, Raman spectroscopy prevents the interference caused by water IR adsorption peaks in the experimental results [31] and fluorescence spectroscopy accurately identifies the types and proportions of substances in the sample. Therefore, as a complex mixture of chemicals such as ethanol, water, esters, and alcohols [32], Chinese liquor can be effectively detected and differentiated using fluorescence and Raman spectroscopy (as shown in Fig. 1).

C. Experiment Setup

1) Baselines and Model Definition: We select some representative traditional machine learning models and deep learning models using supervised or semisupervised techniques as baselines. Meanwhile, to verify the effectiveness of pretraining, we constructed Spectrum-BERT w/o pre.

1) SVM [17]: Following [17], the SVM is used for spectral curve classification. We set the parameters as follows: the kernel function is RBF, $C = 1$, and $\gamma = 1/(n\_features \cdot X \cdot \text{var})$, where $n\_features$ is the feature dimension and $X \cdot \text{var}$ is the data variance.

2) Random Forest [21], [22]: Based on the experience of other researchers [21], [22] in applying random forest (RF) in the field of food spectroscopy, we use the grid search algorithm to tune the RF model. Specifically, we search for $n\_estimators$ (the number of trees) in the range of [20, 140] with the step size 10, and max_depth (the depth of a single decision tree) in the range of [1, 20] with the step size 5, respectively. And finally, $n\_estimators$ is 110 and max_depth is 6.

3) BP [22]: We reference the backpropagation neural network (BPNN) architecture used by Sheng et al. [22] in the field of food identification. The number of neurons in the input layer, output layer, and two hidden layers is determined to be 1000, 13, 250, and 50. After the experiments, we choose that the batch size is 64, the learning rate is 0.001, and the maximum epoch is 1000.
We apply the early stopping mechanism (patience value is 20) in the experiment.

4) 1-D-Deep-CNN [33]: We apply the 1-D convolution-based deep CNN model (1-D-Deep-CNN) proposed by Sang et al. [33] to liquor spectral curve classification. We build the 1-D-Deep-CNN using five convolutional blocks and two fully connected blocks. We adjusted the original hyperparameter settings of the 1-D-Deep-CNN: the learning rate is 0.005, the batch size is 64, and the maximum epoch is 1000. We set the patience value of the early stopping mechanism to 30.

5) ResNet18-1D [34]: To solve the degradation problem of deep neural networks, He et al. [34] proposed ResNet in 2015. We change the convolution in the original ResNet to 1-D convolution to make it suitable for the spectral curve. After extensive experiments, we chose the best-performing 18-layer ResNet as the baseline, i.e., ResNet18-1D. We determine that the batch size is 64, the learning rate is 0.005, and the maximum epoch is 1000. We set the patience value of the early stopping mechanism to 30.

6) SSDL [11]: Due to the significant differences in the spectral distributions of milk and Chinese liquor (as shown in Fig. 1), we omitted the data preprocessing step in the original design of the SSDL. The autoencoder structure with the highest performance, 100-80-13, was selected. The number of principal components, batch size, learning rate, and maximum epoch were determined to be 100, 64, 0.001, and 1000, respectively. We set the patience value of the early stopping mechanism to 20.

7) Spectrum-BERT/Spectrum-BERT$_{w/o	ext{-}pre}$: Spectrum-BERT is the model proposed in this article using the Transformer model architecture and the “unsupervised pretraining & supervised fine-tuning” paradigm. To demonstrate the effectiveness of the “unsupervised pretraining & supervised fine-tuning” paradigm proposed in this article, we constructed a supervised learning model, Spectrum-BERT$_{w/o	ext{-}pre}$, by skipping the pretraining phase of Spectrum-BERT.

2) Basic Settings: The parameters of the Spectrum-BERT are initialized by random initialization and the Adam optimizer [35] is used to optimize the model. The learning rate of Adam is set to 0.001, weight decay is 0.01, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The early-stop strategy is employed during the training process, with the patience value of 20, and the max_epoch = 2000. We leverage the grid search strategy to determine the optimal values of $L$, $A$, $H$, and token_size as 2, 1, 64, and 100, respectively. The batch size is set to 64 and 128 for the pretraining and fine-tuning phases, respectively. To ensure the stability of the experimental results, all the results in this article are the average of ten experimental results, and the standard deviations are recorded.

3) Dataset Partitioning: We partition the dataset into train:valid:test = train_rate:(1 − train_rate)/2:(1 − train_rate)/2 with hyperparameter train_rate (0.8 by default). To ensure the fairness of the experiments, the proportion of samples used in the training or pretraining phase of all models follows the above settings. The parameter fine_tuning_rate (default value: 0.7) is configured for Spectrum-BERT and SSDL, which can be trained with labeled and unlabeled samples. In the fine-tuning phase of Spectrum-BERT, the proportion of the supervised signal is train_rate × fine_tuning_rate. The training phase of SSDL also uses the same amount of supervised signals. In other words, training Spectrum-BERT and SSDL models requires fewer labeled samples than supervised models.

4) Metrics: Referring to the study of the white wine quality classification problem in imbalanced data [21], we chose accuracy, sensitivity, specificity, and error rate as the evaluation metrics for our experiments.

D. Comparative Experiment

We conducted comparative experiments on the model according to the settings defined in Section IV-C. According to the results of Table I, we can draw the following conclusions.

1) With only 70% of supervised signals, our proposed method achieves the best performance on all metrics while requiring fewer parameters and less computational budget. Due to its advanced model architecture and efficient computation, Spectrum-BERT required only 1/568 of the parameters and 1/322 of the FLOPs compared to other baselines (maximum savings). In addition, the low standard deviation of the repeated experiments proves the unparalleled stability of Spectrum-BERT. This clearly demonstrates how our proposed Spectrum-BERT model architecture with lightweight is capable of effectively capturing latent knowledge. A notable observation is the excellent performance of Spectrum-BERT when the train_rate is set to 0.6 and 0.8, reaching 100%. To delve deeper into the convergence process of Spectrum-BERT on train, valid, and test datasets during the training phase, we conduct detailed experiments and analysis (see the Supplementary Material for more details).

2) The effectiveness of our proposed “unsupervised pretraining & supervised fine-tuning” paradigm is confirmed. Comparing Spectrum-BERT and Spectrum-BERT$_{w/o	ext{-}pre}$, the experimental results indicate that unsupervised learning with a large amount of unlabeled data helps the model to capture latent knowledge during the pretraining phase; by using a limited amount of the supervised signal (70%) for fine-tuning, Spectrum-BERT exhibits excellent performance and stability in certain tasks. While both SSDL and Spectrum-BERT employ semisupervised techniques, the performance of SSDL is constrained by the limitations of the PCA data compression algorithm and the weak architecture. This is in line with our conjecture in Section II-A.

3) The advantage of Spectrum-BERT becomes more obvious with a low pretraining sample ratio. With a decrease of train_rate to 0.2, the performance of Spectrum-BERT only suffers a 0.58% decrease (compared to the best) while maintaining good stability. In contrast, both 1-D-Deep-CNN and ResNet18-1D exhibited a decline

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE I
RESULTS OF THE COMPARATIVE EXPERIMENT ON THE SPECTRAL CLASSIFICATION TASK (%) • SVM AND RF DO NOT BELONG TO DEEP NEURAL NETWORKS, SO THE NUMBER OF MODEL PARAMETERS AND CALCULATION AMOUNT CANNOT BE COUNTED. **FINE_TUNING_RATE** IS FIXED TO 0.7. **BOLD** FOR ‘THE BEST,’ AND **UNDERLINE** FOR ‘THE SECOND BEST.’ ‘↑’ INDICATES THE HIGHER THE BETTER, AND ‘↓’ VICE VERSA

| Metrics       | train_rate | RF     | SSDL   | BP     | SVM    | ResNet18-1D | 1-D-Deep-CNN | Spectrum-BERT  |
|---------------|------------|--------|--------|--------|--------|-------------|--------------|---------------|
|               |            |        |        |        |        |             |              |               |
| #Params(M)↓    | -          | -      | 0.02   | 0.26   | 3.88   | 68.13       | 0.12         | 0.12          |
| #FLOPs(M)↓    | -          | 1.09   | 1.05   | -      | 6201.85| 47774.41    | 148.59       | 148.59        |

Accuracy↑
20% 58.02 ± 3.44 86.69 ± 7.09 86.13 ± 7.54 90.92 ± 1.14 90.23 ± 10.41 90.48 ± 27.60 98.55 ± 0.24 99.42 ± 0.29
40% 64.28 ± 3.58 88.56 ± 5.32 86.15 ± 7.76 91.41 ± 1.03 96.49 ± 2.87 99.67 ± 0.28 97.70 ± 0.42 99.54 ± 0.08
60% 64.73 ± 3.77 87.04 ± 7.42 89.15 ± 6.94 91.77 ± 1.10 96.92 ± 3.22 99.80 ± 0.26 99.63 ± 0.49 100.00 ± 0.00
80% 65.08 ± 3.71 88.15 ± 6.33 88.54 ± 3.78 93.69 ± 1.57 99.31 ± 1.16 99.62 ± 0.71 99.70 ± 0.50 100.00 ± 0.00

Sensitivity↑
20% 56.71 ± 4.69 86.83 ± 7.63 85.88 ± 9.24 85.98 ± 8.89 82.10 ± 12.85 89.79 ± 29.74 98.87 ± 0.21 99.47 ± 0.23
40% 61.38 ± 3.35 89.64 ± 5.10 81.28 ± 6.82 90.73 ± 4.18 97.30 ± 1.91 99.69 ± 0.26 98.03 ± 0.27 99.88 ± 0.06
60% 63.62 ± 3.06 86.87 ± 8.64 84.86 ± 8.87 93.33 ± 2.87 97.63 ± 3.57 99.82 ± 0.24 99.73 ± 0.40 100.00 ± 0.00
80% 60.77 ± 5.96 88.13 ± 8.47 84.62 ± 4.17 95.29 ± 1.20 99.48 ± 0.80 99.66 ± 0.61 99.67 ± 10.00 100.00 ± 0.00

Specificity↑
20% 96.57 ± 0.28 98.91 ± 0.57 98.93 ± 0.58 99.29 ± 0.09 98.44 ± 0.81 98.50 ± 4.43 99.88 ± 0.03 99.95 ± 0.02
40% 97.10 ± 0.40 99.06 ± 0.43 98.92 ± 0.46 99.33 ± 0.08 99.72 ± 0.23 99.97 ± 0.02 99.81 ± 0.03 99.99 ± 0.01
60% 97.14 ± 0.32 98.95 ± 0.59 99.17 ± 0.53 99.35 ± 0.08 99.75 ± 0.26 99.98 ± 0.02 97.14 ± 0.04 100.00 ± 0.00
80% 97.18 ± 0.41 99.04 ± 0.50 99.11 ± 0.30 99.49 ± 0.12 99.94 ± 0.09 99.97 ± 0.06 99.17 ± 2.50 100.00 ± 0.00

Error Rate↓
20% 41.98 ± 3.44 13.31 ± 7.09 13.87 ± 7.54 9.08 ± 1.14 19.77 ± 10.41 9.52 ± 27.60 1.45 ± 0.34 0.58 ± 0.29
40% 35.72 ± 3.58 11.44 ± 5.32 13.85 ± 5.76 8.59 ± 1.03 3.51 ± 2.87 0.87 ± 0.28 2.03 ± 0.42 0.16 ± 0.08
60% 35.27 ± 3.77 12.96 ± 7.42 10.85 ± 6.94 8.23 ± 1.10 5.08 ± 3.22 0.13 ± 0.26 9.38 ± 0.49 0.00 ± 0.00
80% 34.92 ± 3.71 11.85 ± 6.34 11.46 ± 4.79 6.31 ± 1.57 0.69 ± 1.16 0.48 ± 0.72 2.50 ± 7.50 0.00 ± 0.00

Fig. 7. Result of the limit experiment. The performance of Spectrum-BERT was evaluated with train_rate fixed at 0.8 and fine_tuning_rate varying at levels of 0.1, 0.2, 0.4, and 0.6, respectively. The four subgraphs are (a) accuracy, (b) sensitivity, (c) specificity, and (d) error rate.

in performance by 9.17% and 19.21% (compared to the best), respectively, along with a sharp decrease in stability. This demonstrates that despite a significant decrease in train samples, Spectrum-BERT still effectively captures influential latent knowledge during the pretraining phase, which positively impacts downstream tasks. However, the limitations of 1-D-Deep-CNN and ResNet18-1D become more apparent when there are not enough train samples. This aligns with our proposition in Section II-A.

E. Limit Experiment
This section conducts a limit experiment to investigate the performance of Spectrum-BERT in scenarios where the availability of supervised signals is severely limited. Specifically, we fixed the train_rate to 0.8 for all experiments. Meanwhile, we adjusted the fine_tuning_rate for SSDL and Spectrum-BERT within {0.1, 0.2, 0.4, 0.6}. Fig. 7 demonstrates that Spectrum-BERT achieves the optimal and most stable performance in all experiments. Spectrum-BERT maintains its lead even when the percentage of supervised signals drops to 10%. The experiments indicate that the “unsupervised pretraining & supervised fine-tuning” paradigm proposed by us allows Spectrum-BERT to handle downstream tasks using only a few labeled samples. This represents substantial progress compared to supervised models such as 1-D-Deep-CNN. This essentially boosts Spectrum-BERT’s applicability in real-world scenarios.

F. Ablation Experiment
The curve partitioning strategy presented in this article is a critical technique that enables Spectrum-BERT to preserve sensitivity to characteristic peak locations and local information of spectral curves. The effectiveness of the strategy is verified through an ablation experiment. When the curve partitioning strategy is not working, each point of the spectral curve is treated as a separate token. Table II demonstrates that Spectrum-BERT experienced a substantial performance decline in all train_rate conditions when the curve partitioning strategy is not employed, and this effect is particularly noticeable when pretraining samples were limited (train_rate = 0.4 or 0.2). The results show that under the guidance of our proposed curve partitioning strategy, the spectral curve features are reasonably dispersed across independent tokens, enabling
Table II

Results of Curve Partitioning Strategy Ablation Experiments (%). "-" Indicates the Nonutilization of the Curve Partitioning Strategy

| token_size | Accuracy | Sensitivity | Specificity | Error Rate |
|------------|----------|-------------|-------------|------------|
| 25%        | 83.21 ± 3.04 | 96.32 ± 0.24 | 83.02 ± 0.24 | 96.32 ± 0.24 |
| 45%        | 92.34 ± 0.23 | 99.35 ± 0.02 | 97.34 ± 0.23 | 99.35 ± 0.02 |
| 65%        | 94.32 ± 0.24 | 99.35 ± 0.02 | 97.34 ± 0.24 | 99.35 ± 0.02 |

Fig. 8. Sample label distribution of the constructed imbalanced dataset.

Table III

Classification Performance of Spectrum-BERT and Baselines on Imbalanced Datasets (%). Bold for "the Best"

| Methods       | Accuracy | Sensitivity | Specificity | Error Rate |
|---------------|----------|-------------|-------------|------------|
| RF            | 78.71 ± 4.65 | 61.79 ± 5.73 | 98.23 ± 0.39 | 21.29 ± 4.63 |
| SSDL(65)      | 80.14 ± 7.91 | 72.39 ± 8.40 | 98.40 ± 0.67 | 19.86 ± 7.91 |
| RP            | 88.86 ± 7.08 | 79.82 ± 9.64 | 99.14 ± 0.55 | 11.14 ± 7.08 |
| 1-D-Deep-CNN  | 91.00 ± 9.11 | 85.90 ± 4.12 | 99.93 ± 0.07 | 9.00 ± 9.11 |
| SVM           | 91.43 ± 25.71 | 90.11 ± 29.67 | 98.57 ± 4.29 | 8.57 ± 25.71 |
| ResNet18-1D   | 97.86 ± 4.62 | 95.32 ± 8.17 | 99.83 ± 0.36 | 2.14 ± 4.62 |
| Spectrum-BERT | 98.71 ± 0.43 | 92.45 ± 5.25 | 99.90 ± 0.03 | 1.29 ± 0.43 |

G. Imbalance Experiment

Uneven distribution of sample categories in train datasets is common in practical application scenarios [21]. We experimented to evaluate Spectrum-BERT’s capability in dealing with the imbalanced dataset. Specifically, we randomly removed samples from the dataset to create an imbalanced dataset, as demonstrated by the distribution in Fig. 8. Table III shows the classification performance of Spectrum-BERT and baselines on the imbalanced dataset. It should be noted that the number of principal components produced by SSDL through PCA can be reduced only up to 69 due to the reduced sample size. The experimental results demonstrate the following:

1) Spectrum-BERT can maintain its lead on imbalanced datasets with outstanding stability. This proves that the Transformer architecture used by Spectrum-BERT is robust in addition to its excellent feature capture capability. Spectrum-BERT can learn crucial information from limited samples by its characteristic peak locations and local information sensitivity. In contrast, both ResNet18-1D and 1-D-Deep-CNN, which presented good performance in Section IV-D, exhibited considerable performance deterioration and fluctuations when handling imbalanced datasets (the rates of decline are 1.5% and 8.2%, respectively). This demonstrates that while the convolution limitation can be mitigated by utilizing the larger structure and complex computation with adequately and uniformly distributed samples, these limitations significantly affect the model’s performance when the sample distribution is imbalanced. This is in line with our conjecture in Section II-A.

2) ResNet18-1D, which achieves a slightly weaker performance in Section IV-D, surpasses 1-D-Deep-CNN and becomes the second-best model in the imbalance experiment. We believe that the residual connectivity helps ResNet18-1D better resistance to overfitting. When the sample distribution is imbalanced, ResNet18-1D exhibits better robustness in capturing sample features, particularly for underrepresented classes such as #4 in Fig. 8. However, the computational consumption of ResNet18-1D is excessively high.

H. Hyperparameter Studies

1) Token_size: The curve partitioning strategy proposed in this article is one of the primary algorithms that enable Spectrum-BERT to be sensitive to characteristic peak locations and local information. As a critical parameter of the curve partitioning strategy, we study the effect of the hyperparameter token_size on the performance of Spectrum-BERT. Fig. 9 illustrates that varying the token_size has minimal impact on the performance of Spectrum-BERT, but unexpected performance fluctuations occur with token_size of 50 and 200. Overall, Spectrum-BERT has a classification error rate that remains below 0.9% under all conditions, while maintaining accuracy, sensitivity, and specificity above 99%. The advanced mechanism for feature capture proposed in this article enables Spectrum-BERT to learn by processing tokens with different feature densities. After evaluating the model’s performance and stability in various scenarios, the default token_size of 100 for Spectrum-BERT is a superior choice.

2) Model Structure: Generally speaking, the more quantity of layers (L) and Self-Attention heads (A), the stronger ability to capture features, and the better downstream task performance. However, by examining Fig. 10(a) and (b), we observe that increasing the number of layers L and Self-Attention heads A does not bring significant performance gain to Spectrum-BERT. It is believed that training becomes more complicated when the L is too large. This is due to impeded learning of some model parameters, preventing better values. On the other hand, the study shows that the MSA mechanism separates the feature matrix into multiple independent subspaces to be computed individually during computation. This computation undoubtedly destroys the continuity of the spectral curves and may lead to the fragmentation
Fig. 9. Performance of Spectrum-BERT under different train_rate conditions when token_size is set to 50, 100, and 200, respectively. The four subgraphs are (a) accuracy, (b) sensitivity, (c) specificity, and (d) error rate.

Fig. 10. Performance of Spectrum-BERT with different values of hyperparameters. (a) Fixed $H = 256$, $A = 8$ to explore the impact of different layers $L$ on the model performance. (b) Fixed $L = 2$, $H = 256$ to explore the effect of different numbers of Self-Attention heads $A$ on the classification performance of Spectrum-BERT. (c) Fixed $A = 1$, $L = 2$ to explore the effect of different hidden layer sizes $H$ on the classification performance.

of characteristic peaks. It is commonly assumed that enhancing the performance of a model is also possible by increasing the dimension of the hidden layer $H$. The experimental results displayed in Fig. 10(c) demonstrate that Spectrum-BERT can achieve performance saturation when $H = 64$. Increasing the hidden layer dimension leads to extra computational resource consumption. Beyond the saturation point, a larger hidden layer dimension introduces additional noise, degrading the model performance. Taking into account performance, stability, complexity, and resource consumption, it is believed that the optimal values for $L$, $A$, and $H$ are 2, 1, and 64, respectively.
V. CONCLUSION

Facing the label deficiency problem in spectral curve classification of Chinese liquors, this article is the first to propose Spectrum-BERT. In this method, we adopt the Transformer model architecture and “unsupervised pretraining & supervised fine-tuning” paradigm in the field of spectral feature extraction and classification. We innovatively partition the curve into multiple blocks to retain the model’s sensitivity to the characteristic peak position and local information of the spectral curve. The two well-designed pretraining tasks, NCP and MCM, allow the model to effectively utilize unlabeled samples to capture the potential knowledge of the spectral data, breaking the limitations of insufficiently labeled samples and improving the applicability and performance of the model in practical scenarios. As expected, Spectrum-BERT with lower parameters retains its lead in terms of performance despite using less supervised signal fine-tuning. Moreover, ablation and imbalance experiments demonstrate the superiority of the proposed Transformer architecture and curve partitioning strategy in extracting spectral curve features. Furthermore, Spectrum-BERT uses significantly fewer model parameters and FLOPs than other baselines, with only 1/568 and 1/322, respectively.

In the future, we will improve the block partition strategy of the spectrum curve, so that the model can adaptively determine a more reasonable token_size and reduce the impact of human factors on the model. According to our survey, expectation-maximum algorithm-based [36], reinforcement learning-based [37], and recurrent neural network-based [38] token block optimization strategies are three potentially valuable research directions. Furthermore, the dynamic token_size-based token partitioning strategy is also an interesting research direction.

REFERENCES

[1] Y. Tian, Y. Sun, Y. Wang, X. Li, and D. Zhu, “Development of a handheld system for liquor authenticity detection based on laser spectroscopy technique,” J. Spectrosc., vol. 2022, pp. 1–8, Apr. 2022.

[2] M.-M. Sun, “Application of computer-assisted blending in the production of Taishan Tequ liquor,” Liquor-Making Sci. Technol., vol. 120, no. 6, p. 53, 2003.

[3] S. Wang, H. Chen, and B. Sun, “Recent progress in food flavor analysis using gas chromatography-ion mobility spectrometry (GC-IMS),” Food Chem., vol. 315, Jun. 2020, Art. no. 126158.

[4] H. Hou, Q. Meng, P. Qi, and T. Jing, “A hand-held electronic nose system for rapid identification of Chinese liquors,” IEEE Trans. Insrum. Meas., vol. 70, pp. 1–11, 2021.

[5] Y. Shi, H. Yuan, Q. Zhang, A. Sun, J. Liu, and H. Men, “Lightweight interleaved residual dense network for gas identification of industrial polypropylene coupled with an electronic nose,” IEEE Trans. Insrum. Meas., vol. 70, pp. 1–10, 2021.

[6] C. Cozzolino, “Advantages, opportunities, and challenges of vibrational spectroscopy as tool to monitor sustainable food systems,” Food Anal. Methods, vol. 15, no. 5, pp. 1390–1396, May 2022.

[7] S. Yin, Z. Cui, Z. Bi, H. Li, W. Liu, and Z. Tian, “Wide-range thickness determination of oil films on water based on the ratio of laser-induced fluorescence to Raman,” IEEE Trans. Insrum. Meas., vol. 71, pp. 1–11, 2022.

[8] H. Jin, Q. Lu, X. Chen, H. Ding, H. Gao, and S. Jin, “The use of Raman spectroscopy in food processes: A review,” Appl. Spectrosc. Rev., vol. 51, no. 1, pp. 12–22, Jan. 2016.

[9] J. Qin, K. Chao, and M. S. Kim, “Raman chemical imaging system for food safety and quality inspection,” Trans. ASABE, vol. 53, no. 6, pp. 1873–1882, 2010.

[10] J. Qin et al., “Subsurface inspection of food safety and quality using line-scan spatially offset Raman spectroscopy technique,” Food Control, vol. 75, pp. 246–254, May 2017.

[11] M. Said, A. Wahba, and D. Khalil, “Semi-supervised deep learning framework for milk analysis using NIR spectrometers,” Chemometrics Intell. Lab. Syst., vol. 228, Sep. 2022, Art. no. 104619.

[12] M. Chen et al., “Generative pretraining from pixels,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1691–1703.

[13] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” in Proc. Int. Conf. Learn. Represent., 2021. [Online]. Available: https://openreview.net/forum?id=Z9hCDWe6ke

[14] W. Liu, “K-BERT: Enabling language representation with knowledge graph,” in Proc. AAAI 34th AAAI Conf. Artif. Intell., vol. 34, no. 3, 2020, pp. 2901–2908.

[15] X. Wang et al., “KEPLER: A unified model for knowledge embedding and pre-trained language representation,” Trans. Assoc. Comput. Linguistics, vol. 9, pp. 176–194, Mar. 2021.

[16] Y. Liu, H. Fu, and D.-W. Sun, “Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices,” Trends Food Sci. Technol., vol. 113, pp. 193–204, Jul. 2021.

[17] Y. Yan, J. Ren, J. Tschannerl, H. Zhao, B. Harrison, and F. Jack, “Non-destructive phenolic compounds measurement and origin discrimination of peated barley malt using near-infrared hyperspectral imagery and machine learning,” IEEE Trans. Insrum. Meas., vol. 70, pp. 1–15, 2021.

[18] H. Zhu, A. Gowen, H. Feng, K. Yu, and J.-L. Xu, “Deep spectral–spatial features of near infrared hyperspectral images for pixel-wise classification of food products,” Sensors, vol. 20, no. 18, p. 5322, Sep. 2020.

[19] J. Liu, M. Osadchy, L. Ashton, M. Foster, C. J. Solomon, and S. J. Gibson, “Deep convolutional neural networks for Raman spectrum recognition: A unified solution,” Analyst, vol. 142, no. 21, pp. 4067–4074, 2017.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., vol. 1, Minneapolis, MN, USA, J. Burstein, C. Duran, and T. Solorio, Eds., Jun. 2019, pp. 1–15.

[21] G. Hu, T. Xi, F. Mohammed, and H. Miao, “Classification of wine quality with imbalanced data,” in Proc. IEEE Int. Conf. Ind. Technol. (ICIT), Mar. 2016, pp. 1217–1712.

[22] T. Sheng, S. Shi, Y. Zhu, D. Chen, and S. Liu, “Analysis of protein and fat in milk using multilayer gradient-boosted regression tree,” IEEE Trans. Insrum. Meas., vol. 71, pp. 1–10, 2022.

[23] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, I. Guyon et al., Eds., Red Hook, NY, USA: Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/files/paper/2017/file/3f58243854d7ee910b1f053c1c4a845aa-Paper.pdf

[24] H. Bao, L. Dong, S. Xiao, and F. Wei, “BEiT: BERT+ pre-training of image transformers,” in Proc. Int. Conf. Learn. Represent., 2022. [Online]. Available: https://openreview.net/forum?id=p-RhZ52904

[25] M. E. Peters et al., “Knowledge enhanced contextual word representations,” in Proc. Conf. Empirical Methods Natural Lang. Process, 9th Int. Joint Conf. Natural Lang. Process., K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 45–54.

[26] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Represent., 2021. [Online]. Available: https://openreview.net/forum?id=8vYcbfNNTY

[27] R. Alec, N. Karthik, S. Tim, and S. Ilya, “Improving language understanding with unsupervised learning,” Citado, vol. 17, pp. 1–12, Oct. 2018.

[28] Y. Sun et al., “ERNIE: Enhanced representation through knowledge integration,” 2019, arXiv:1904.09223.

[29] M. Joshi et al., “Deformable DETR: Improving pre-training by representing and predicting spans,” Trans. Assoc. Comput. Linguistics, vol. 8, pp. 64–77, Dec. 2020.

[30] Y. Sun et al., “SRPAIS: Spectral matching algorithm based on Raman peak alignment and intensity selection,” in Artificial Intelligence and Security, Cham, Switzerland: Springer, 2022, pp. 386–399.

[31] Z. Zhang, J. Jiang, G. Wang, H. Wang, J. Liu, and H. Wang, “Application of two-dimensional correlation UV-vis spectroscopy in Chinese liquor Moutai discrimination,” Amer. J. Anal. Chem., vol. 6, no. 5, pp. 395–401, 2015.
[32] Z. Li, P.-P. Wang, C.-C. Huang, H. Shang, S.-Y. Pan, and X.-J. Li, “Application of Vis/NIR spectroscopy for Chinese liquor discrimination,” Food Anal. Methods, vol. 7, no. 6, pp. 1337–1344, Jul. 2014.

[33] X. Sang, R.-G. Zhou, Y. Li, and S. Xiong, “One-dimensional deep convolutional neural network for mineral classification from Raman spectroscopy,” Neural Process. Lett., vol. 54, no. 1, pp. 677–690, Feb. 2022.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[36] T. K. Moon, “The expectation-maximization algorithm,” IEEE Signal Process. Mag., vol. 13, no. 6, pp. 47–60, Nov. 1996.

[37] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” J. Artif. Intell. Res., vol. 4, no. 1, pp. 237–285, Jan. 1996.

[38] Y. Wang, K. Li, R. Huang, S. Song, L. Yang, and G. Huang, “Glance and focus: A dynamic approach to reducing spatial redundancy in image classification,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 2432–2444.

Yansong Wang is currently pursuing the master’s degree with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai, China. His research interests include machine learning, graph representation learning, recommender system, and spectral data analysis.

Yundong Sun is currently pursuing the Ph.D. degree with the Department of Electronic Science and Technology, Harbin Institute of Technology, Harbin, China. He has published several articles in journals such as TKDE, IoTJ, and Neurocomputing. His research interests include machine learning, graph representation learning, recommender system, and spectral data analysis.

Yansheng Fu is currently pursuing the Ph.D. degree with the School of Instrument Science and Engineering, Harbin Institute of Technology, Harbin, China. His research interests include multimodal representation learning, machine learning, and graph neural network.

Dongjie Zhu received the Ph.D. degree in computer architecture from Harbin Institute of Technology, Weihai, China, in 2015. He is an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology. He has published more than 60 articles in several journals or conferences. His research interests include machine learning and spectral data analysis.

Zhaoshuo Tian (Member, IEEE) is a Professor with the Department of Electronic Science and Technology, Harbin Institute of Technology, Weihai, China. His research interests include laser technology and marine laser detection technology.