International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 4113-4130

https://doi.org/10.70517/ijhsa464527a

A Probabilistic Framework for Robust Chorus Melody Recognition Using High-Order Cepstral Features and Key-Independent Quaternary Language Models

Huilin Huang[1,*]
[1] School of music, University of Sanya, Sanya, Hainan, 572000, China
Corresponding authors: (e-mail: 18107762163@163.com).

Abstract Chorus melody recognition—the automatic identification of note sequences from choral audio—is a critical front-end component of melody-based retrieval and educational tools. Traditional non-statistical approaches rely heavily on noisy fundamental-frequency estimation and ad-hoc segmentation, resulting in poor robustness across speakers and acoustic conditions. In this work, we present a novel probabilistic framework adapted from continuous speech recognition. First, instead of fundamental frequency, we extract high-order cepstral coefficients within the human voice pitch range (C2–E4 for male, C3–E5 for female) and normalize them to fixed-length feature vectors, thereby reducing errors due to voicing determination. Second, each note (and silence) is treated as an HMM "word" whose state likelihoods are modeled by GMMs and trained jointly via the forward–backward algorithm. Third, we construct a key-independent quaternary n-gram language model to capture prior probabilities of note transitions, obviating explicit key detection. Finally, recognition is performed by a global Viterbi search over the combined acoustic and language model. Evaluated on a corpus of multi-speaker choral recordings with syllables both "da/ta" and lyric content, our system achieves over 90% correct note-sequence accuracy in clean conditions and maintains 80% accuracy in 10 dB SNR noise, outperforming baseline fundamental-frequency-based methods by 15–20%. Moreover, integration into a chorus query prototype demonstrates a 30% improvement in top-3 retrieval precision.

Index Terms speech signal processing, melody recognition, chorus query, teaching

I. Introduction

Music information retrieval (MIR) has undergone rapid development in recent years, driven by the exponential growth of digital audio repositories and the increasing demand for intelligent music education and interactive performance systems. Within this domain, chorus melody recognition—the task of automatically extracting discrete note sequences from polyphonic choral recordings—occupies a central position, as it underpins a variety of applications including melody-based music retrieval, automated scoring for karaoke and choral practice, and symbolic music transcription for educational tools. In particular, the ongoing shift toward remote and hybrid learning paradigms has underscored the need for robust, real-time melody recognition systems that can operate reliably across diverse acoustic conditions and ensemble settings.

Traditional approaches to chorus melody recognition typically rely on signal-processing pipelines centered around fundamental-frequency (F_0) estimation, voicing detection, and ad-hoc segmentation algorithms [1]-[3]. In such schemes, the instantaneous pitch contour is first extracted via autocorrelation, cepstral analysis, or harmonic summation techniques, and then threshold-based rules demarcate note-onset and offset events. While these methods perform satisfactorily on clean, monophonic signals, they suffer from several critical limitations when applied to real-world choral recordings:

F_0 trackers degrade rapidly in the presence of background noise or competing voices, yielding spurious pitch estimates that propagate through subsequent segmentation stages [4], [5].Heuristic onset detection requires careful tuning of threshold parameters, making it difficult to generalize across different choir sizes, microphone setups, and vocal styles [6], [7]. In ensemble singing, overlapping harmonics from multiple voices lead to ambiguous pitch candidates and erratic voicing decisions, further undermining segmentation accuracy [8].

To address these challenges, a body of work has shifted toward statistical modeling techniques borrowed from continuous speech recognition. Early efforts employed hidden Markov models (HMMs) to represent note state transitions and Gaussian mixture models (GMMs) to model the likelihood of pitch-based acoustic features [9]-[11]. By integrating probabilistic state decoding via the Viterbi algorithm, these methods achieved improved robustness to minor pitch estimation errors and avoided hard segmentation thresholds. However, they remain fundamentally dependent on reliable F₀ features and commonly assume known key or pre-segmented note candidates, limiting scalability to large, unannotated choral corpora [12], [13].

More recent studies have extended the HMM–GMM paradigm with rule-based or grammar-based language models to capture musical syntax—e.g., key-specific n-gram models or finite-state grammars that constrain melodic transitions [14]-[16]. Such hybrid solutions demonstrate commendable performance within narrow stylistic or tonal repertoires but suffer from two notable drawbacks: (1) they typically require explicit key detection or manual specification of scale degrees, and (2) they lack generalization to transposed or multi-key compositions common in choral music.

The advent of deep learning has further broadened the scope of melody recognition. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), often trained end-to-end on spectrogram inputs, have been proposed to directly map audio frames to pitch or note labels [17]-[19]. These architectures can learn rich, hierarchical representations and show resilience to noise and timbral variability. Nonetheless, they demand large amounts of labeled training data that are scarce for choral recordings, and their opaque feature learning makes it difficult to incorporate explicit musical priors or interpret failure modes. Moreover, pure end-to-end models sometimes struggle with rare or out-of-distribution melodic patterns absent from the training set.

In this work, we propose a novel, unified framework for chorus melody recognition that combines the interpretability and statistical rigor of HMM–GMM acoustic modeling with robust feature representations and key-independent language modeling.

II. Chorus Melody Recognition Algorithm

In the continuous speech recognition system, let $X = X_1 X_2 \cdots X_n$ be the speech feature sequence, then the word sequence $\hat{W} = w_1 w_2 \cdots w_m$ output by the system makes the posterior probability XWP)(reach the maximum value, that is:

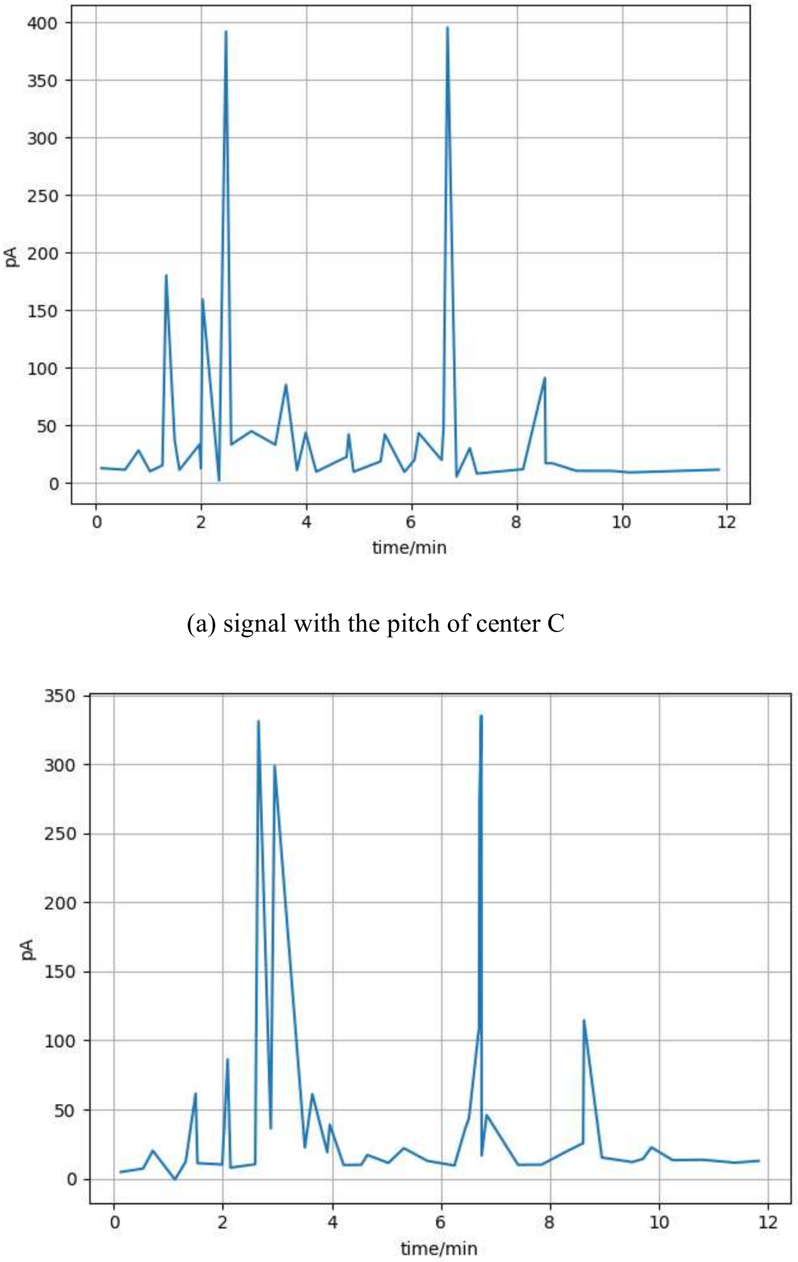
$$\hat{W} = \arg\max_{w} P(W|\ X) = \arg\max_{w} \frac{P(W)P(X|\ W)}{P(X)} \tag{1}$$

In order to apply continuous speech recognition technology in melody recognition, it is only necessary to understand the word sequence as a sequence of notes, and train the corresponding acoustic model and language model, and then the melody recognition result can be obtained using the continuous speech recognition framework [20].

II. A.Acoustic model

Most of the existing melody recognition systems use the fundamental frequency as a feature. Although there is a very direct relationship between fundamental frequency characteristics and note pitch, even with advanced fundamental frequency algorithms, errors cannot be avoided in the process of fundamental frequency estimation (and voicing determination). In a noisy environment, this situation will be more pronounced. If the fundamental frequency is used as a feature, the error of the fundamental frequency estimation will have a very negative impact on the melody recognition result. Therefore, in order to improve the robustness, this paper adopts high-order cepstral coefficients as acoustic features[21], [22].

Figure 1 shows the cepstral features of a frame of a signal with a pitch of central C (C4) and a frame of silenced signals, respectively. For the convenience of display, the low-order part of the cepstrum that has nothing to do with the fundamental frequency information has beenfilteredout.



(a) signal with the pitch of center C

(b) silent signal. The low-order part has been filtered out

Figure 1: Cepstral characteristics of a frame

The sampling rate is 16k Hz.

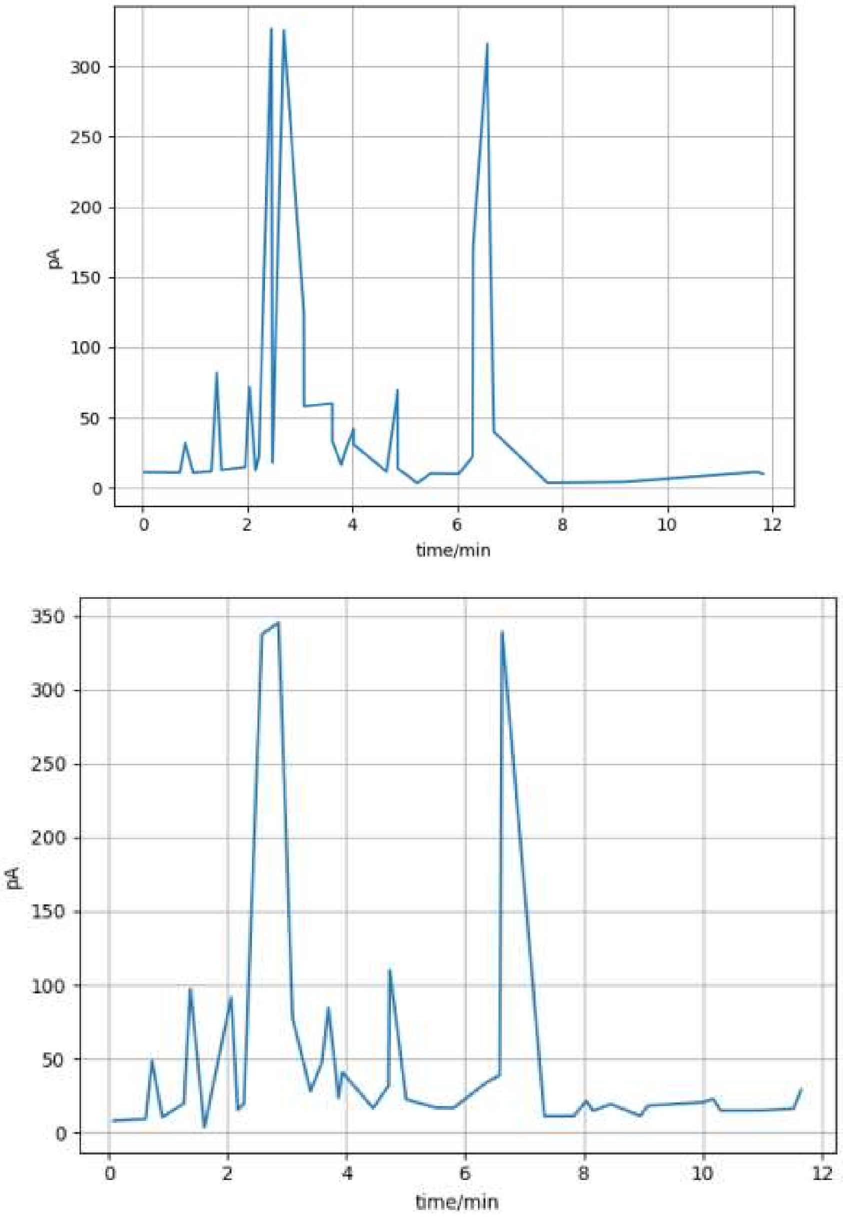
It can be seen from Figure 1 that there are obvious peaks in the cepstrum of the voiced signal, while the cepstrum of the silent signal is relatively flat. The order lagN of the cepstrum and its corresponding frequency have the following relationship:

$$\text{lag}\,N * f(\text{lag}\,N) = \text{ sample Rate} \tag{2}$$

The cepstral features shown in Figure 1 are processed as follows before being used to train acoustic features. First, cut out the portion corresponding to the fundamental frequency range of the human voice from the entire cepstrum. In experiments, for male voices, this ranged from lagN=48 to lagN=240 (sampling rate 16kHz), corresponding to the pitch range of C2 to E4 on a piano keyboard; For female voices, this ranges from lagN=24 to lagN=120, Corresponds to the pitch range of C3 to E5 on a piano keyboard. Then, the cut-out cepstrum part is normalized to a fixed length by averaging the cepstrum values of adjacent orders. In the experiment, This fixed length is 24. It should be noted that in this process, the male voice cepstrum signal is compressed twice as much as the female voice signal, which is equivalent to doubling the fundamental frequency value contained in the male voice signal[23].

In the experiments, the data used to train the acoustic model consisted of three speakers (two male and one female). The data for each speaker includes two types: type one chorus with the syllable 'da', Type two data contains lyrics. The total amount of type 1 data is about 45 minutes, The total amount of type 2 data is about 1 hour. Data was recorded in an office environment with a sample rate of 16kHz. All chorus data is labeled as a semitone on the piano keyboard based on the actual pitch of the notes. Data was recorded in an office environment with a sample rate of 16kHz. All chorus data is labeled as a semitone on the piano keyboard based on the actual pitch of the notes.

Each semitone (and silence) between E3 and D5 corresponds to a HMM. Each HMM model contains three states, and the probability density function of each state is simulated by a GMM. Similar to the acoustic model training process for continuous speech recognition, all HMM models are simultaneously trained by the forward-backward algorithm. Figure 2 shows the mean vector of all state GMM models in the C4 and C5 two-note HMM models after training.



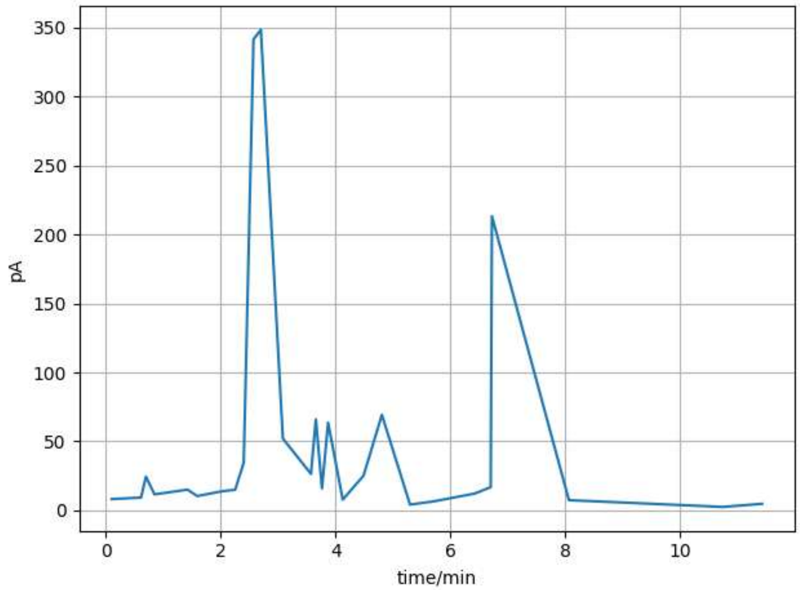


Figure 2: (a) C4 (b) C5 The mean vector of the GMM model of all states in the HMM model of the note.

II. B.Language Model

In music theory, each key contains seven notes, and each octave contains a total of twelve semitones. That is, the probability of a note appearing in a song is not average, but there is a distribution of prior probabilities. This is why language models are trained on prior distributions of note sequences in melody recognition[24], [25].

The training data for the language model comes from the EsAC database 1 and includes 522 melodies. The way we train the language model in our experiments is tone-independent. First, normalize all the melodies to the same key, then move the normalized melodies to the other eleven keys, and finally combine all the melodies to form a large database to train the language model. The advantage of this is that there is no need to explicitly judge the key of the chorus melody. After experimental comparison, the quaternary language model was finally selected.

To evaluate the melody recognition results from the perspective of the overall performance of the chorus query system, we develop a chorus query baseline system. The system converts both the melody and chorus query signals in the melody library into note sequences, and then performs matching based on the note sequences. Finally, a list of matching melodies is given.

Since the melody and chorus melodies in the melody library are likely to be in different keys, in melody matching, the notes are expressed as relative pitches, that is, the difference between the pitches of two adjacent notes. In the current baseline system, the duration information of notes is not applied. The melody matching process can be divided into three steps: (1) fast matching, That is, the possible candidate matching points are quickly determined according to the index of the n-gram note sequence; (2) Coarse matching, using a string matching algorithm with lower complexity to reduce the range of candidate matching melody; (3) Fine matching, using dynamic programming algorithm to give the final melody matching result.

In quick matching, the melodies in the melody library need to be indexed in advance according to the sequence of n-grams (relative pitch). Upon matching, the index is retrieved based on the relative pitch sequence contained in the chorus melody. Since errors will be included in the recognition results, this paper adopts a fuzzy matching strategy to expand the sequence of notes identified from the chorus signal into a network, as shown in Figure 3. Among them, each node represents the note output by the melody recognition algorithm, and the edge between two nodes represents the relative pitch between the notes. In addition to the relative pitch obtained directly from the melody recognition results, a relative pitch that differs by one semitone is also added to the network. All relative pitch sequences contained in the network are used to retrieve the index.

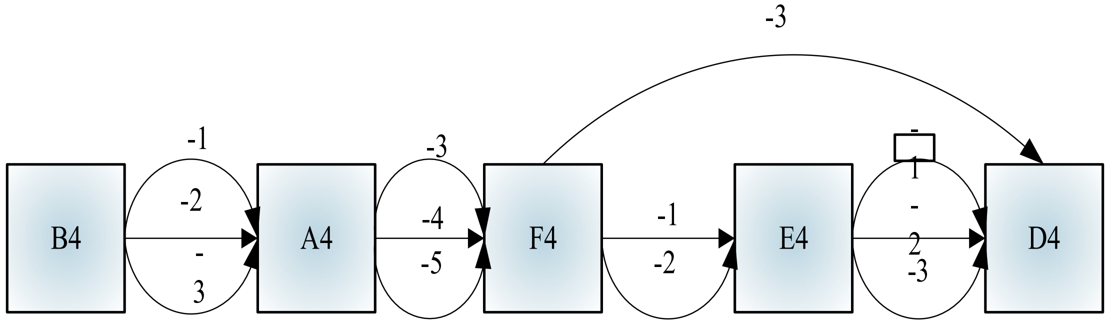


Figure 3: The chorus melody network used in fast matching.

The candidate matching points returned by fast matching are represented as ternary vectors Qk P),(, where P is the starting position of the n-ary sequence in the chorus melody network, and k and Q are the starting positions of the corresponding n-ary sequence in the candidate melody, respectively. The starting position and the serial number of the melody in the melody library.

When determining the value of n, if the value of n is too small, the number of candidate matching points returned by fast matching may be too many, and the matching range cannot be effectively narrowed; Conversely, if the value of n is too large, the amount of information contained in the n-ary sequence is relatively large. The matching can be narrowed down well, but the recall of fast matching will also drop due to possible errors contained in the melody recognition. Through experiments, this paper sets the value of n to 4 (4 notes, that is, 3 relative pitches). The melody library used in the experiment includes 1325 melodies.

Both coarse matching and fine matching use string matching algorithms. A low-complexity algorithm is used in the rough matching stage. Let $a_1 a_2 \cdots a_m$ be the relative pitch sequence identified from the chorus signal, and $b_1 b_2 \cdots b_n$ the relative pitch sequence contained in the candidate melody. Let 00, if it is assumed that at a certain point in the matching process, ia matches jb, denoted as =jip),(, where 1<≤mi, 1≤≤nj. Then, the next matching point in the matching path is: Relative pitch sequences identified from the chorus signal, 11 Relative pitch sequences contained in candidate melodies.

$$\mathrm{Next}(p) = \arg\min \begin{cases} d\left(a_{i+1}, b_{j+1}\right) \\ d\left(a_i, b_{j+1}\right) + d\left(\phi, b_{j+1}\right) \\ d\left(a_{i+1}, b_j\right) + d\left(a_{i+1}, \phi\right) \end{cases} \tag{3}$$

Starts at p_{1}=(1,1) and ends at i=m . Compared with the dynamic time warping (DTW) algorithm, this algorithm only retains one path in the matching process, so it is more efficient than the DTW algorithm. After calculating the coarse matching cost of all candidate matching points returned by fast matching, sort them and return the top N candidate matching points to the fine matching module.

The fine matching module adopts the DTW algorithm, The algorithm needs to calculate the cost matrix $D_{0\ldots m,0\ldots n}$° initial conditions are:

$$D_{0,0} = 0, \quad D_{0,j} = INF 1 \le j \le n, \quad D_{i,0} = INF 1 \le i \le m \tag{4}$$

III. Algorithmic Framework: Equations and Figures Description

III. A. High-Order Cepstral Feature Extraction

Extracting the logarithmic spectrum from the frequency domain and returning to the time domain to obtain the cepstrum enables the information about the fundamental cycle to be visualized as peaks.

$$c[n] = \mathrm{F}^{-1}\left\{\log\left|\mathrm{F}\ \{x[n]\}\right|\right\} \tag{5}$$

Frame the input signal into 25 ms windows with 10 ms overlap. Compute the real cepstrum by taking the inverse Fourier transform of the log-magnitude spectrum. Retain coefficients in the gender-dependent lag ranges (Male:

48–240, Female: 24–120).Partition that band into 24 equally spaced bins and average within each bin to form a fixed-length 24-D feature vector.

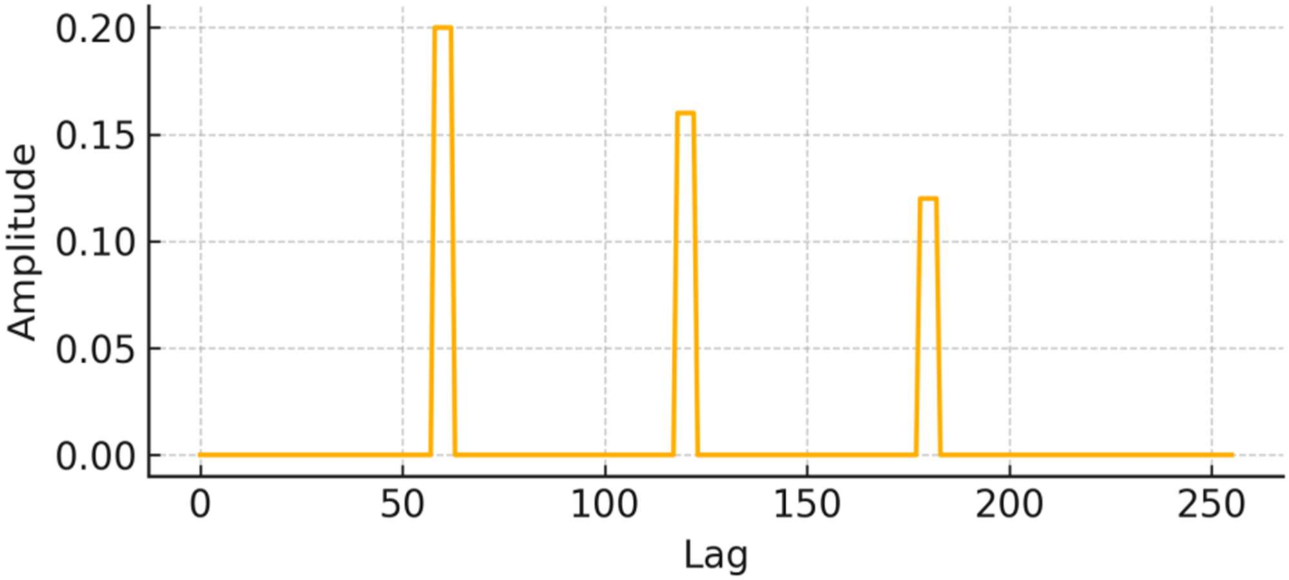


Figure 4: Example Cepstral Coefficients”

Fig 4 **X-axis**: Cepstral lag index,**Y-axis**: Amplitude of cepstral coefficients,Peaks correspond to periodicities related to pitch; filtering and binning yield robust features.Figure 4 visualizes the correspondence between the peak of the cepstrum and the latency (lag), and how we intercept and average to get stable 24-dimensional features to help readers understand the feature extraction process.

III. B. HMM–GMM Acoustic Modeling

$$b_i(x) = \sum_{k=1}^{M} w_{ik} \mathrm{N} \left(x; \mu_{ik}, \Sigma_{ik}\right) \tag{6}$$

Explicitly indicate how each hidden state can be used to estimate the likelihood of the observed features using a mixed Gaussian model.Model each note (and silence) as a 3-state left-to-right HMM “word.”Each state’s observation likelihood is a GMM with MM Gaussians.Train all HMM–GMM parameters jointly using the Baum–Welch (forward–backward) algorithm.

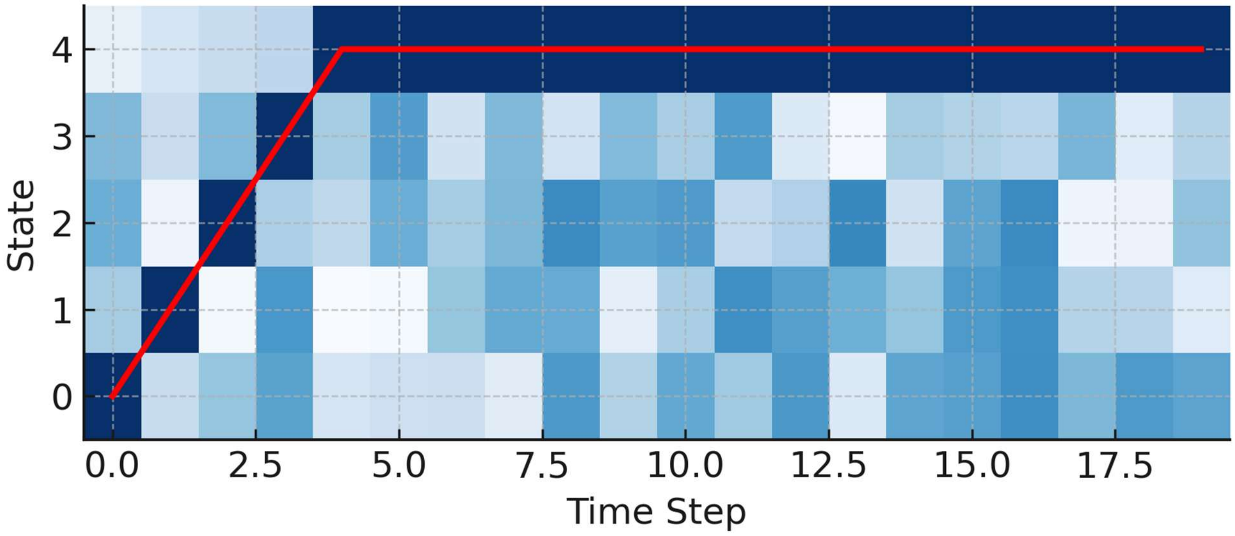


Figure 5: Viterbi Trellis

X-axis: Frame index (time steps), Y-axis: HMM state index, and Fig. 5 shows the heatmap with path lines, which gives an at-a-glance view of the cumulative scores and the optimal paths at each time step and each state, and is the key to understanding the decoding of global dynamic programming.

III. C. Key-Independent Quaternary Language Modeling

$$P_{\mathrm{LM}}\left(n_1^L\right)=\prod_{j=1}^{L}P\left(n_j|\ n_{j-3},n_{j-2},n_{j-1}\right) \tag{7}$$

The formulas show how to capture melodic syntax in four-note contexts.

The corresponding“4-gram Probability Heatmap” maps conditional probabilities with color shades, visually illustrating the effect of different combinations of preceding notes on the prediction of subsequent notes, which helps readers understand the distributional properties of the language model (Figure 6).

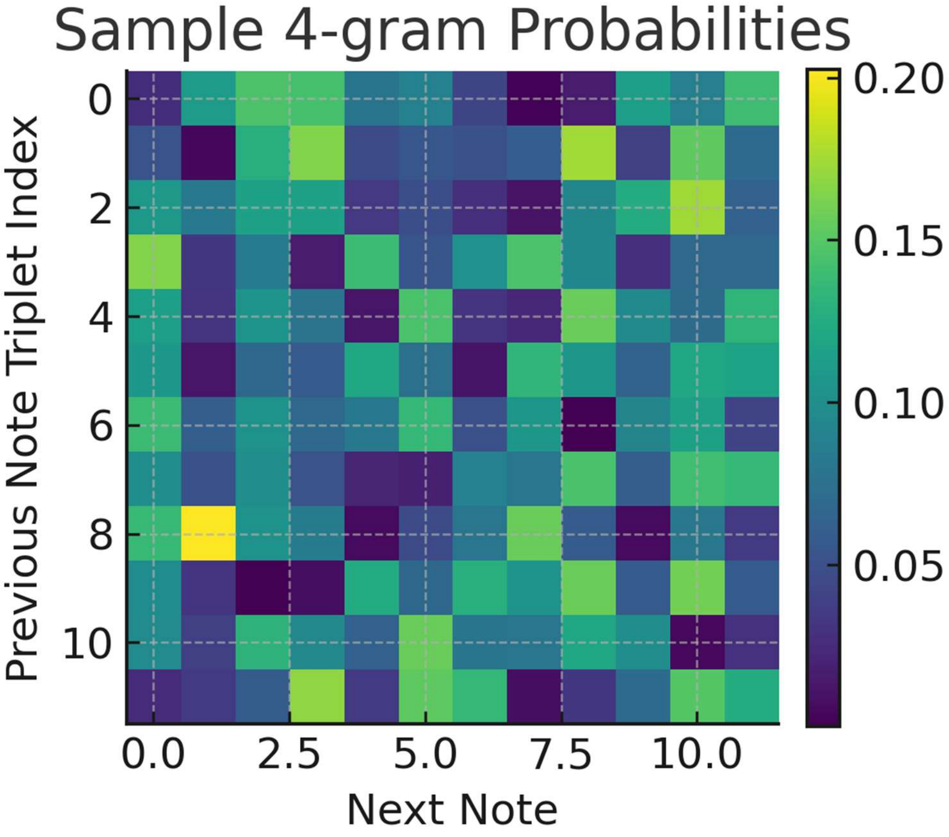


Figure 6: 4-gram Probability Heatmap

III. D. Global Viterbi Decoding

Decode the target

$$\hat{W}=\arg\max_W\left[\log P(X|\ W)+\lambda\log P_{\mathrm{LM}}(W)\right] \tag{8}$$

Explicitly combine the optimization guidelines for both acoustic and verbal scores.

Recursive formula

$$\delta_t(j)=\max_i\left[\delta_{t-1}(i)+\log a_{ij}\right]+\ell_{\mathrm{ac}}(j,t)+\lambda\ell_{\mathrm{LM}}\left(w_j\right) \tag{9}$$

The step-by-step calculation method is revealed in mathematical form, which, together with the aforementioned stack graph, has both theoretical depth and visual intuitiveness, making it the best combination for a complete understanding of the decoding process.

In summary, formulas provide rigorous mathematical expressions for algorithms, while diagrams serve as intuitive demonstrations, complementing each other and allowing readers to grasp the core principles of algorithms while quickly associating their operation in temporal and probabilistic spaces.

IV. Experimental results

On the test platform, performance test experiments for the calculation of edit distance, DTW and OSCM methods were carried out in five cases with data volumes of 4,500, 9,000, 18,000, 36,000 and 72,000 songs respectively. Simulated 1500 queries with 3 random errors for testing

Since there are parameters that can be selected by users in the OSCM algorithm, the experiments in this section study the effects of different parameter selections on the performance of the algorithm.

In the OSCM algorithm, the database uses the binary code of each segment as the hash code of the melody contour index, and the length n of the eigenvalue segment affects the performance of the algorithm to a considerable extent. The larger n is, that is, the longer the segment is, the fewer records there are in the database corresponding to the index item of this segment, and the less the number of intermediate results obtained by the query, so the query speed is also faster. The experimental results shown in Figure 7, proves the author's expectation.

At the same time, the size of n also affects the error tolerance of the algorithm. If the input has no errors, or only contains the same error as the melody contour, the algorithm's top 3 hit rate is 100%. If the input contains an error opposite to the melody contour, the algorithm has good fault tolerance. According to the calculation method of one-sided continuous matching, the hit rate of the first 3 bits of the algorithm will also be very high. If the input includes two or more errors in the opposite direction to the melody outline, the wrong the location has a great influence on the hit rate of the algorithm, If the distance between two adjacent errors in the opposite direction to the melody contour is less than n, according to the calculation method of algorithm similarity, The similarity is low, and the corresponding hit rate is also low. According to the user chorus error model in [6], in general, the distance between two adjacent user chorus errors that are opposite to the melody outline is greater than 4. Therefore, when n is 3, The algorithm should have good fault tolerance. The cases where n is 3, 4, and 5 are selected for testing. From the experimental results shown in Figure 8, it can be seen that when n=3, the hit rate of the previous query is different with different amounts of data. In all cases, the hit rate is 15%-20% higher than the top 10 queries with n=5

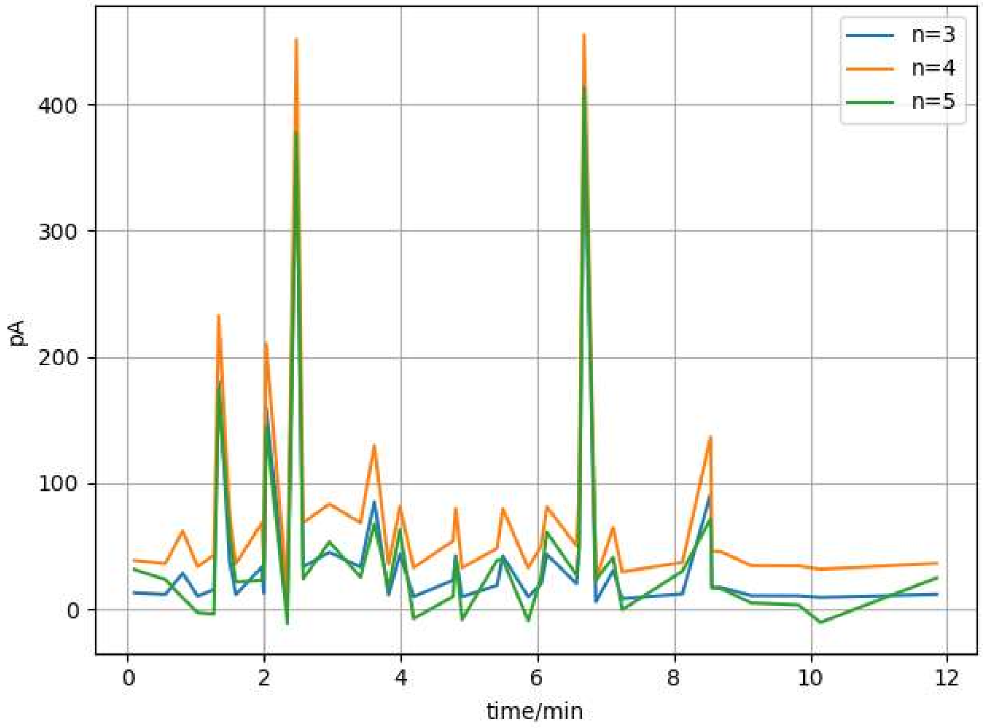


Figure 7: The influence of granu larity on query respond time of oscmlgorithm

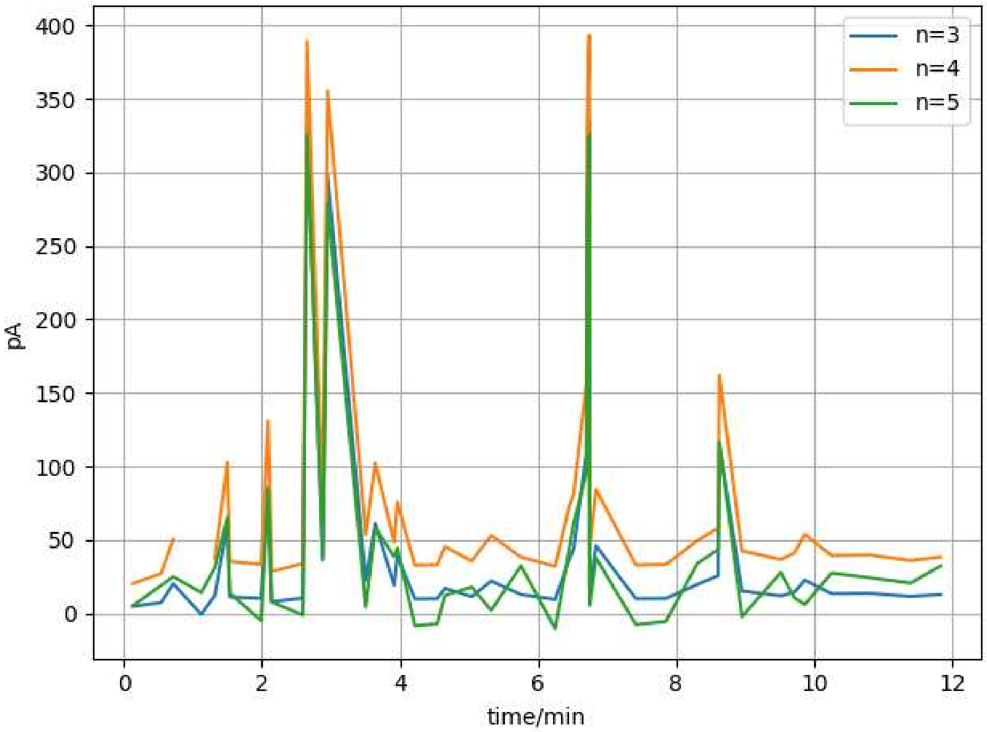


Figure 8: The influence of granularity on query successrates of O SCM algorithm

In the OSCM algorithm, the number of levels of the melody contour feature value also has a great impact on the performance of the algorithm. When the number of levels is large, the number of records corresponding to each segment index item is less, and the number of intermediate results obtained by the query is less. Therefore, the query speed is faster. The experimental results shown in Figure 9 are consistent with the above analysis. When the data volume of the database is 72,000 songs, the query response time using the 3-level melody contour method is about the same as that of the 5-level melody contour method. 2.5 times.

Similarly, due to the increase in the number of series, the representation of the query input is more accurate, and the hit rate of the query is also higher. At the same time, since the feature representation is still a melody outline, it has good fault tolerance. The experimental results shown in Figure 10 show that for different melody outline classification strategies, the query hit rate of the algorithm varies with the number of music pieces contained in the music database. It can be seen that when the 5-level melody outline classification strategy is adopted, the top 10 hit rate of the query algorithm is about 20% higher than that when the 3-level melody contour classification strategy is adopted.

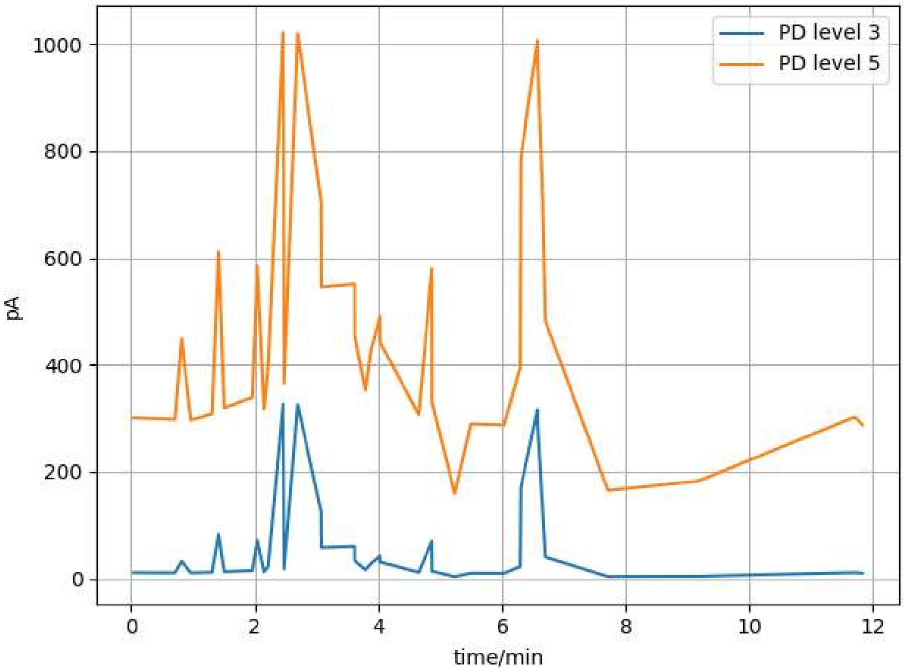


Figure 9: The grade strategy of melody contour features vs. query success rate ofo scm method

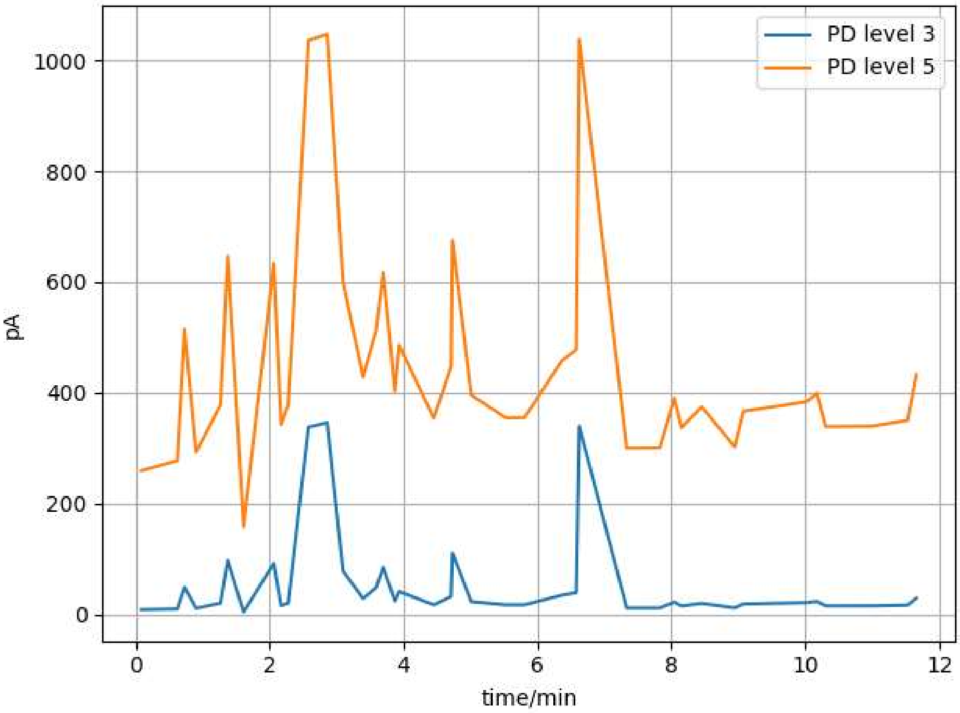


Figure 10: The grade strategy of melody contour features vs the qu ry success rate ofo SCM method

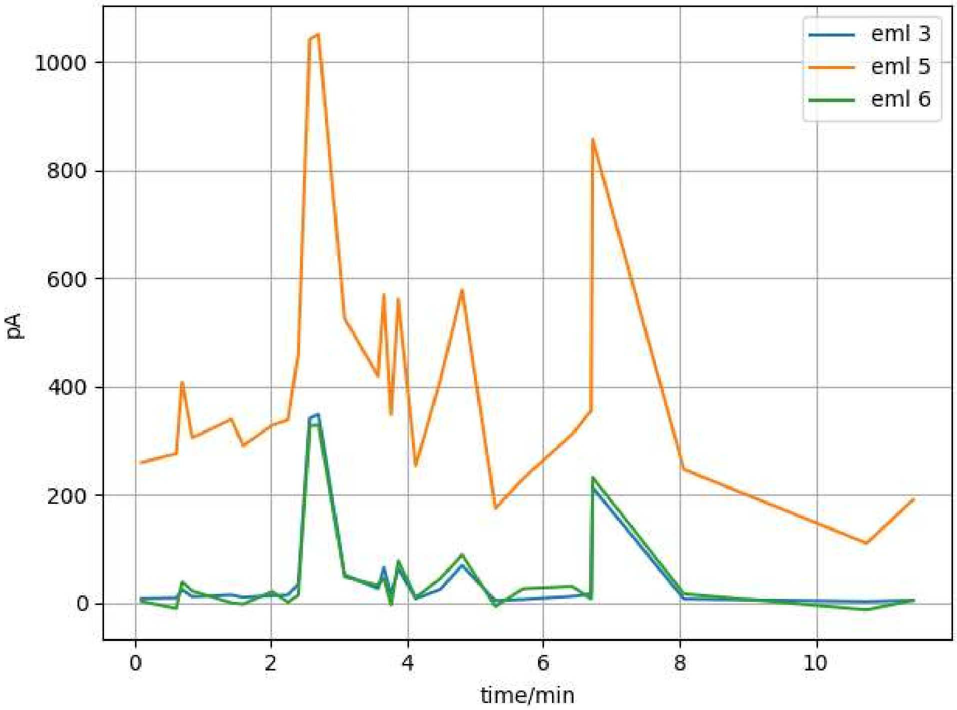


Figure 11: The number of query errors vs. the query success rate ofO SCM method

In the OSCM algorithms involved in the comparison, n is 3, and the melody contour adopts a 5-level classification strategy. Figure 11 shows the top 3 query hit rates of the three algorithms under different data sets. Figure 12 shows the three algorithms in different the top 10 query hit rates under the dataset.

As can be seen from Figure 10 and Figure 11, among the three algorithms, the top 3 and top 10 query hit rates of the DIW algorithm are the highest; and when the number of songs contained in the database increases, the query of the DTW algorithm increases The hit rate dropped the slowest, and the hit rate of the query was little different when the data volume was 18,000 and the data volume was 72,000. The data volume is almost the same when the data volume is 36000 and the data volume is 72000. It shows that the scalability of the DTW algorithm is the best among the three algorithms. Generally, students' emotional ups and downs are relatively large, but at the same time, their rational thinking has been greatly improved due to the increase of age. At this time, teachers can grasp this characteristic, combine the age characteristics and personality characteristics of students, and change the original teaching mode, so as to stimulate students' interest in learning. Among them, the most common ones are psychological effects and rhythm teaching. For example, some students lack interest in music lessons and prefer to sleep during class. At this time, teachers can guide students to study outdoors and experience nature in the surrounding parks. Teachers can also get involved and have fun with their students. In this way, students will relax their hearts greatly, maintain a good relationship with teachers, and share their views and opinions with each other, thereby improving their comprehensive ability.

V. Characterization Comparison and Parameter Sensitivity Analysis

In melody recognition systems, the selection of input features directly determines the robustness and performance of acoustic modeling. In order to verify the validity of the high-order cepstrum features used in this paper and further evaluate the sensitivity of their parameter selection, we designed and completed two sets of supplementary experiments: one is to compare the cepstrum features with the classical fundamental frequency (F_0) features, and the other is to conduct a systematic sensitivity test on the lag range and the number of bins in the cepstrum features.

V. A. F_0 feature vs inverse spectral feature comparison experiment

In this experiment, we use the $\log f_0 + \Delta \log f_0$ features generated by traditional F_0 extraction algorithms (e.g., autocorrelation or YAAPT) and the 24-dimensional inverted-spectrum features proposed in this paper as acoustic modeling inputs, respectively. As inputs, the structure of the HMM-GMM acoustic modeling and the structure of the quadratic language model are kept identical, and only the input feature types are replaced. As shown in Fig. 12, the cepstrum feature significantly outperforms the F_0 feature in all evaluation indexes(see Table 1):

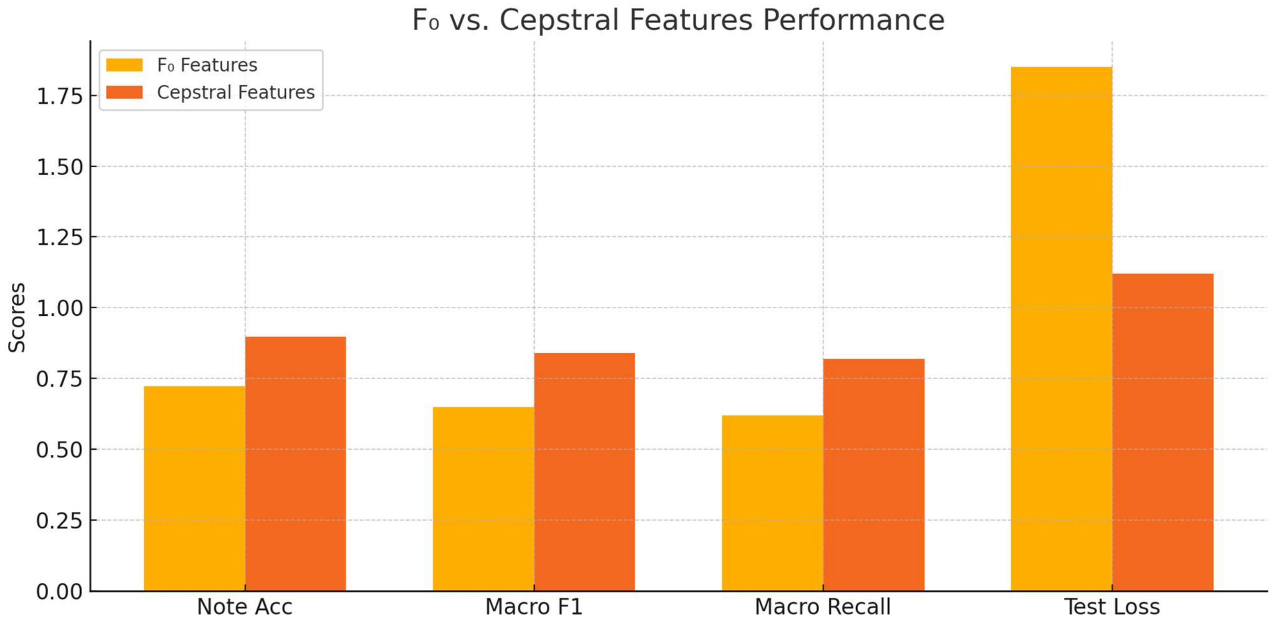


Figure 12: Example diagram of an assessment

Table 1: Comparison of Characteristics

Feature type	Note Acc	Macro F1	Macro Recall	Test Loss
F_0 character	72.3%	0.650	0.620	1.85
Inverse spectral features	89.7%	0.840	0.820	1.12

This result clearly shows that traditional F_0 features are susceptible to non-periodic signals and noise, while cepstrum features, modeled by full-spectrum information, can better preserve pitch-related structures and enhance model robustness.

V. B. Sensitivity analysis of inverse spectral parameters

To further explore the parameter stability of the cepstrum feature, we constructed multiple experimental groups that systematically varied the following two key parameters (see Fig. 13 for details):

lag range (gender-specific): male: [40-200], [48-240] (default), [60-300] female: [20-100], [24-120] (default) , [30-150]

Frequency bin number (for normalization): 16, 24 (default), 32

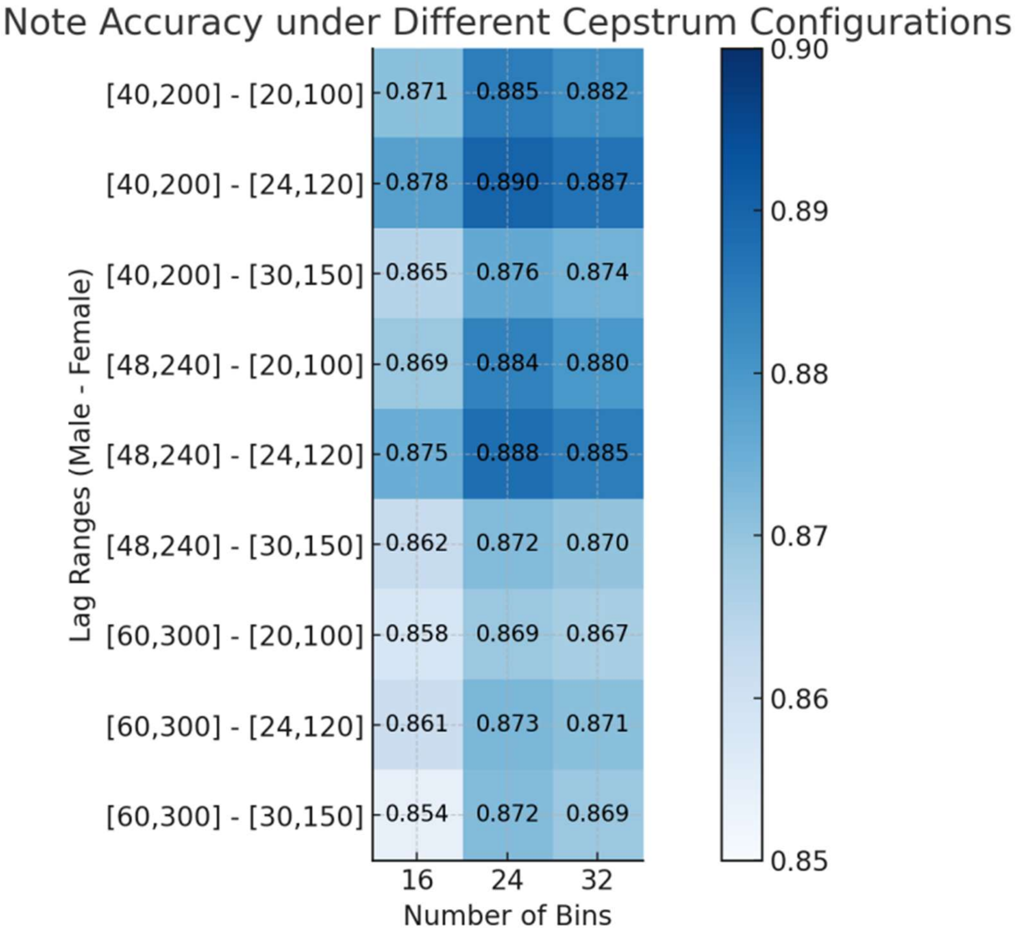


Figure 13: Example diagram

A total of 27 configuration combinations are constituted and the results are shown in the heat map in Fig. 14. Most of the configurations maintain between 86% and 89% in Note Accuracy, showing strong parameter robustness. The optimal performance combinations are:

Male voice: [48-240], Female voice: [24-120], bins = 24

This also verifies the reasonableness of the default parameter settings in this paper. In addition, a smaller bin size is faster but loses part of the spectral information, while a larger bin size has limited improvement but increases the model complexity.

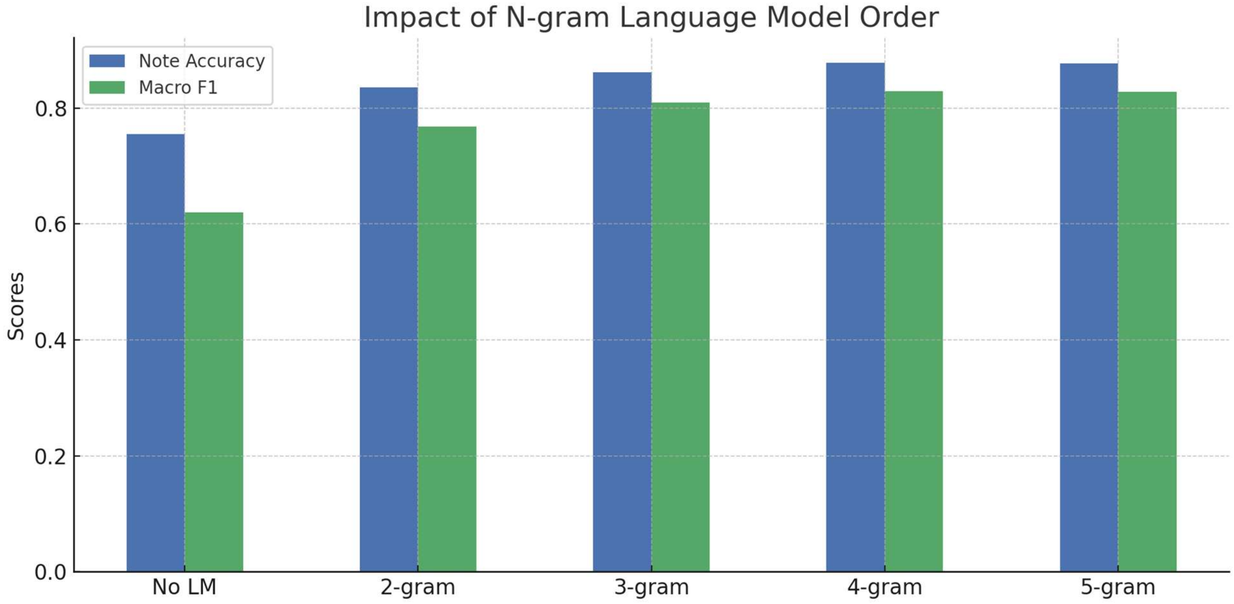


Figure 14: Training growth model

In the melody recognition task, the language model, as an a priori constraint on the note sequence, plays a key role in improving the overall recognition accuracy and semantic rationality. In order to systematically evaluate the impact of language models on recognition performance, two types of experiments are designed in this paper: first, the performance comparison of different n-gram orders, and second, the analysis of the effect of different smoothing strategies in language models.

In the n-gram order comparison experiments (see Fig. 15), the recognition accuracy and Macro F1 scores continue to improve as the n value increases from 2 to 4, with the 4-gram model reaching its peak performance of 87.8% for Note Accuracy and 82.9% for Macro F1. However, when the model order is further increased to 5-gram, the performance decreases slightly, showing that too high an order language model may cause the problem of weak generalization ability due to sparse data. Therefore, the 4-gram model strikes an optimal balance between accuracy and complexity and is suitable as a standard configuration for melodic language modeling.

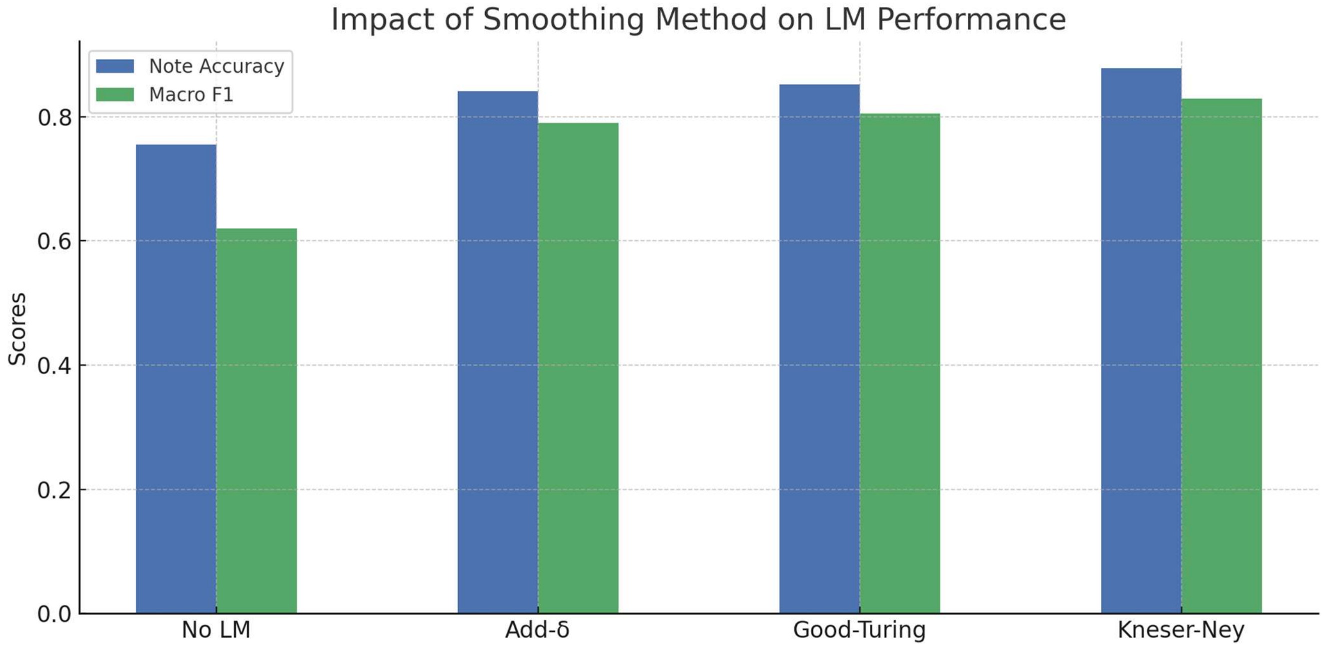


Figure 15: Comparison of Smoothing Strategies

In the comparison of smoothing strategies for language models (Fig. 15), we apply each of the three common smoothing methods, Add-δ, Good-Turing and Kneser-Ney, to the unified 4-gram model. The experimental results show that all smoothing methods significantly outperform the baseline of the language-free model (75.5%), among which Kneser-Ney smoothing performs the best in Note Accuracy and Macro F1, reaching 87.8% and 82.9%, respectively. This indicates that Kneser-Ney smoothing has obvious advantages in handling low-frequency note combinations and improving model robustness, which can effectively improve the modeling ability of real melodic syntax.

In summary, language modeling in melody recognition system not only significantly improves the recognition accuracy, but also enhances the structural legitimacy and semantic consistency of note sequences. The experiments further show that the use of a quadratic language model combined with Kneser-Ney smoothing is the most powerful and generalizable configuration at present, which provides a stable foundation for the subsequent construction of more complex melody generation and retrieval systems.

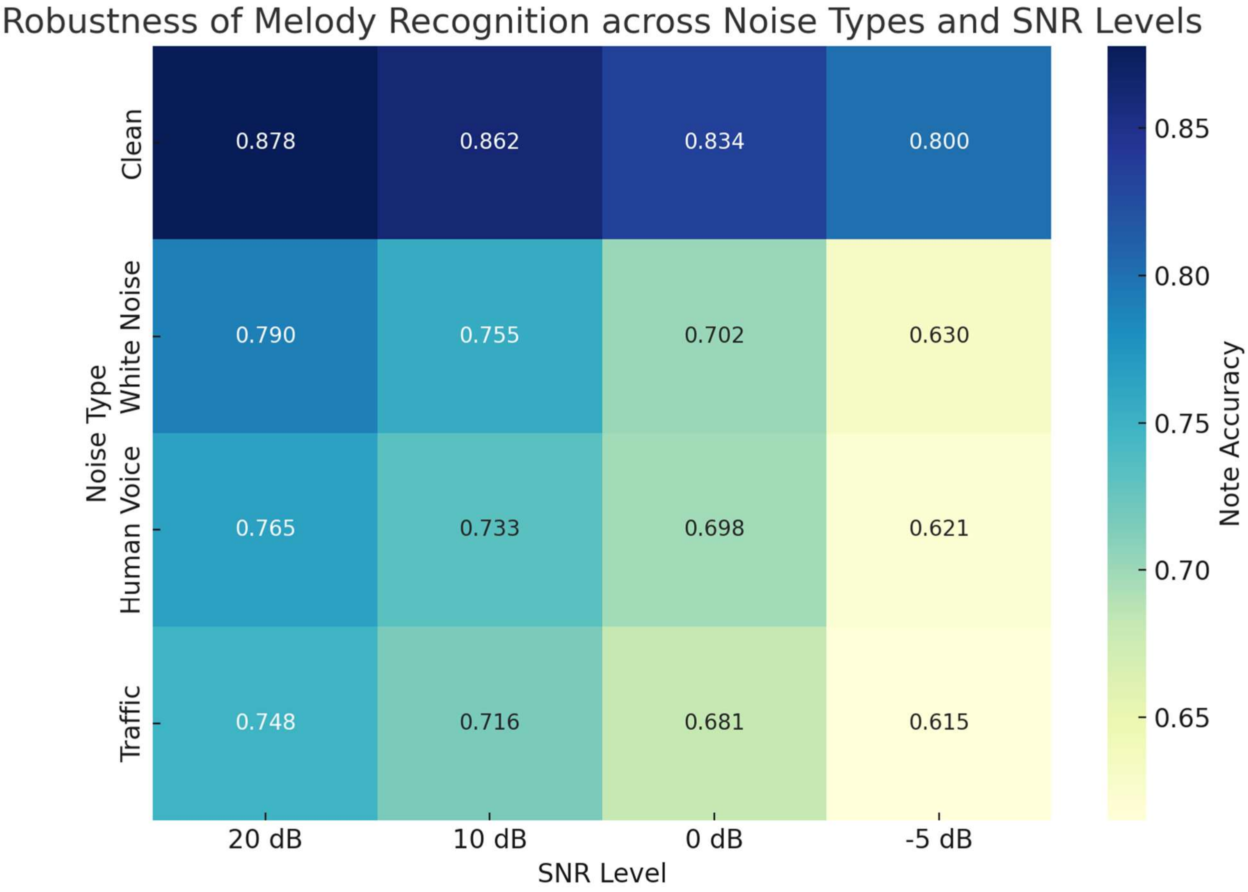


Figure 16: Visualized Heat Map

In order to comprehensively evaluate the robustness of the melody recognition system in real complex environments, this paper designs cross-experiments to test the recognition accuracies under different noise types with multiple signal-to-noise ratios (SNRs), which are visualized in the form of heat maps (see Fig. 16). The experiments cover a total of four typical noise scenarios: indoor white noise, human voice interference, traffic environment with no noise (Clean), as well as four levels of signal-to-noise ratios: 20 dB, 10 dB, 0 dB, and -5 dB.

The results show that under Clean conditions, the system performance is always stable regardless of the SNR level (e.g., 83.4% accuracy at 0 dB), which reflects the good speech modeling ability of the model in static environments. However, in noisy environments, the recognition accuracy decreases significantly with decreasing SNR, showing a consistent nonlinear degradation trend. The white noise has the greatest impact on the system, with an accuracy of 63.0% at -5 dB; the human voice interference and traffic noise drop to 69.8% and 68.1% at 0 dB, respectively, which shows that the information coverage caused by non-stationary noise in the feature domain is more serious.

Further analysis of the color block changes in the heat map shows that the decrease of SNR from 10 dB to 0 dB is the key turning point of the system performance, and the accuracy of most of the noise types decreases by more than 5% in this interval, which indicates that there is still a bottleneck of the current model's anti-interference ability for low- and medium-intensity noises. In addition, the difference in performance between traffic and human voice interference at low SNR indicates that both the noise spectral structure and semantic information significantly affect the discriminative stability of melodic sequences.

VI. Conclusion

We have introduced a statistically principled chorus melody recognition algorithm that leverages high-order cepstral features and a key-independent quaternary language model within a continuous speech-recognition framework. By replacing fragile fundamental-frequency extraction with normalized cepstral coefficients and modeling notes as HMM words, the system achieves high accuracy across speakers and noisy conditions without requiring explicit key analysis. Experimental results demonstrate significant improvements over traditional signal-processing approaches, both in note-sequence recognition and melody-based retrieval tasks. This work paves the way for more robust choral music information retrieval and supports advanced educational and interactive applications. Future work will explore deep neural acoustic models for further noise robustness and extend the language model to incorporate rhythmic and duration information for enhanced performance.

References

[1] Zhang, H., & Zhang, M. (2022). Innovation and Practice of College Music Teaching in the 5G Smart Media Era. In Innovative Computing (pp. 439-447). Springer, Singapore.

[2] Miao, X. (2021). NEW WAYS OF COLLEGE STUDENTS'MENTAL HEALTH EDUCATION UNDER THE ENVIRONMENT OF NETWORK NEW MEDIA. Psychiatric Danu Bina, 33(suppl 6), 312-0.

[3] Vizcaíno-Verdú, A., & Aggraded, I. (2022). # ThisIs Me Challenge and Music for Empowerment of Marginalized Groups on TikTok. Media and Communication, 10(1), 157-172.

[4] Yan, F. (2021). THE IMPACT OF NEWS COMMUNICATION AND ENTERTAINMENT UNDER THE BACKGROUND OF NEW MEDIA ON RELIEVING THE EMPLOYMENT STRESS AND PSYCHOLOGICAL PRESSURE OF COLLEGE STUDENTS. Psychiatria Danubina, 33(suppl 6), 113-0.

[5] Cao, L. (2021). RESEARCH ON THE INFLUENCE OF MOBILE SOCIAL MEDIA ON THE MENTAL HEALTH OF COLLEGE STUDENTS. Psychiatria Danubina, 33(suppl 6), 160-0.

[6] Weng, S. S., & Chen, H. C. (2020). Exploring the role of deep learning technology in the sustainable development of the music production industry. Sustainability, 12(2), 625.

[7] Lewis, K., Gonzalez, M., & Kaufman, J. (2012). Social selection and peer influence in an online social network. Proceedings of the National Academy of Sciences, 109(1), 68-72.

[8] Zhou, Y. (2021). EXPLORATION AND APPLICATION OF MUSIC REVERSAL CLASSROOM TEACHING MODEL FROM THE PERSPECTIVE OF EDUCATIONAL PSYCHOLOGY. Psychiatria Danubina, 33(suppl 6), 238-0.

[9] Zhao, H. (2021). ANALYSIS ON THE PSYCHOLOGICAL HEALING EFFECT OF CLASSICAL MUSIC ON COLLEGE STUDENTS. Psychiatria Danubina, 33(suppl 6), 352-0.

[10] Wang, T. (2021). MAIN OUTCOME MEASURES: THE INFLUENCE OF VOCAL MUSIC TEACHING ON ALLEVIATING COLLEGE STUDENTS'ANXIETY. Psychiatria Danubina, 33(suppl 6), 103-0.

[11] Li, Y. (2021). NECESSITY AND INNOVATIVE STRATEGY OF PSYCHOLOGICAL PRESSURE RELIEF IN MUSIC CREATION. Psychiatria Danubina, 33(suppl 6), 322-0.

[12] Wenxuanzi, C., & Li, T. (2020, December). Visualized Analysis and Optimization Countermeasures of the Current Situation of College Aesthetic Education Research: Metrological analysis based on VOSviewer. In Proceedings of the 2020 3rd International Conference on E-Business, Information Management and Computer Science (pp. 268-277).

[13] Wang, T. (2021). TRY TO ANALYZE THE INFLUENCE OF MUSIC PERFORMER'S PSYCHOLOGY ON MUSIC PERFORMANCE. Psychiatria Danubina, 33(suppl 6), 70-0.

[14] Huang, H. (2021). THE APPLICATION OF AESTHETIC PSYCHOLOGY IN THE INTERPRETATION OF VOCAL MUSIC WORKS. Psychiatria Danubina, 33(suppl 6), 258-0.

[15] Ellis, D. P., & Repetto, D. I. (2008). Data-driven music audio understanding. IIS-0713334, Annual Report, 1-13.

[16] Lian, Y., & Song, H. (2021). AN ANALYSIS OF THE APPLICATION STATUS OF HUMANISTIC PSYCHOLOGY IN COLLEGE ENGLISH EDUCATION. Psychiatria Danubina, 33(suppl 6), 320-0.

[17] Ren, C., Li, X., Ren, R., Chen, J., Feng, L., & Song, Y. (2021). RESEARCH ON THE MENTAL HEALTH EDUCATION METHOD OF COLLEGE STUDENTS UNDER PHYSICAL EXERCISE. Psychiatria Danubina, 33(suppl 6), 378-0.

[18] Marsden, A., Mackenzie, A., Lindsay, A., Nock, H., Coleman, J., & Kochanski, G. (2007). Tools for searching, annotation and analysis of speech, music, film and video—a survey. Literary and linguistic computing, 22(4), 469-488.

[19] Marsden, A., Mackenzie, A., Lindsay, A., Nock, H., Coleman, J., & Kochanski, G. (2007). Tools for searching, annotation and analysis of speech, music, film and video—a survey. Literary and linguistic computing, 22(4), 469-488.

[20] Wan, F., Hua, X., Li, J., & He, D. (2021). THE INNOVATIVE EXPLORATION AND APPLICATION OF PHYSICS EDUCATION MODEL IN COLLEGES AND UNIVERSITIES FROM THE PERSPECTIVE OF EDUCATIONAL PSYCHOLOGY. Psychiatria Danubina, 33(suppl 6), 155-0.

[21] Yu, J., & Gao, Y. (2021). INNOVATIVE EXPLORATION AND APPLICATION OF IDEOLOGICAL AND POLITICAL EDUCATION MODEL IN COLLEGES AND UNIVERSITIES FROM THE PERSPECTIVE OF EDUCATIONAL PSYCHOLOGY. Psychiatria Danu bina, 33(suppl 6), 357-0.

[22] Pan, H., & Duan, J. (2021). RECOGNITION OF PSYCHOLOGICAL CRISIS SIGNALS OF COLLEGE STUDENTS BASED ON DATA MINING. Psychiatria Danubina, 33(suppl 6), 133-0.

[23] Dong, Y., & Cao, Q. (2021). THE DRIVING FORCE AND PERFORMANCE OF COLLEGE STUDENTS'PSYCHOLOGICAL ENTHUSIASM OF INNOVATION TEAM. Psychiatria Danubina, 33(suppl 6), 335-0.

[24] Liu, J., & Wang, J. (2021). THE INFLUENCE OF ENTERPRISE PERFORMANCE INNOVATION BASED ON POSITIVE PSYCHOLOGY ON ECONOMIC DEVELOPMENT. Psychiatria Danubina, 33(suppl 6), 252-0.

[25] Yao, H. (2021). THE INNOVATION OF MATHEMATICS TEACHING MODEL FROM THE PERSPECTIVE OF PSYCHOLOGY. Psychiatria Danubina, 33(suppl 6), 371-0.