Signal Processing 212 (2023) 109134

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Polyphonic piano transcription based on graph convolutional network

Zhe Xiao [a,b], Xin Chen [a,b,*], Li Zhou [c]

[a] School of Automation, China University of Geosciences, Wuhan, 430074, P. R. China
[b] Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan, 430074, China
[c] School of Arts and Communication, China University of Geosciences, Wuhan, 430074, P. R. China

ARTICLE INFO

Article history:
Received 7 December 2022
Revised 28 April 2023
Accepted 5 June 2023
Available online 19 June 2023

Keywords:
Automatic music transcription
Deep learning
Graph convolutional network
Multi-label classification
Chord structure modeling

ABSTRACT

The automatic music transcription (AMT) task is designed to convert raw performance audio signals into digital representations of symbolic music for possible computational musicology. In polyphonic music, there are multiple notes and they may appear at the same time. The combined output of multiple notes faces the problem of dimension explosion, which makes it difficult to achieve accurate transcription. To overcome the above challenge, a deep learning model based on graph convolution network (CR-GCN) is proposed to cope with the problem of dimension explosion by exploring the interdependence between musical notes. The model is divided into two parts, i.e., feature learning and label learning. Feature learning is composed of serial convolutional neural network (CNN) and recurrent neural network (RNN), which aims to extract temporal and spatial features from the input music signals. Label learning consists of graph convolutional network (GCN), which is adopted to model the interdependence between notes. The CR-GCN model can be end-to-end trainable through joint training feature networks and label networks. Experiments on public polyphonic music data sets show that the proposed method is able to mine more co-existing notes, and it is superior to existing methods in both frame-level and note-level indexes. Moreover, visual analysis shows that the learned interdependence between notes has good explainability in music theory.

1. Introduction

Automatic music transcription (AMT) is a major task in the field of music information retrieval (MIR), and the automatic transcription of polyphonic music is the most challenging task of AMT [1]. It is committed to automatically converting a piece of audio into symbolic music representation, such as MIDI. It has a wide range of applications, including music automatic search and annotation, musicology analysis, auditory scene analysis, music topic classification, etc.

Polyphonic music transcription is a very difficult challenge even for professional musicians due to the complexity of polyphonic music. As shown in Fig. 1, polyphonic music transcription aims to extract note events from the playing audio (Fig. 1 (a)) and finally obtain the piano-roll representation of the score (Fig. 1 (c)).

From the perspective of signal processing, polyphonic music signals have strong coupling in both the time domain and frequency domain [2]. Multiple notes are triggered simultaneously in polyphonic music, making them overlap each other in the time domain. Unlike voice signals, the frequency distribution of the musical signal conforms to the twelve-tone equal temperament [3], and the fundamental frequency among the notes of different octaves is strongly coupled with the harmonics. Therefore, it is difficult to distinguish multiple notes which are occurring simultaneously, especially when there is an octave relationship between the notes.

From the perspective of machine learning, the following three issues need to be considered in order to complete the accurate and efficient transcription of polyphonic music:

1. Timing. The music signal presents a strong timing relationship, with continuity on a short-term scale (frame-level) and certain musical structure on a long-term scale (note-level) [4];
2. Dimension explosion. The possible combination of notes leads to the explosion of output space dimension. Taking piano transcription as an example, the output space dimension of 88 keys is 2^{88} per frame;
3. Sparsity. There are so many possibilities for combinations of multiple notes from 88 notes, the dataset cannot cover all possible note combination, i.e., the sparsity of sample space.

The existing approaches to solving the AMT task are divided into four different types, which are *feature-based, statistical model-based,spectral decomposition-based*, and *deep learning-based*. The features-based models are committed to analyzing the charac-

* Corresponding author.
E-mail address: chenxin@cug.edu.cn (X. Chen).

https://doi.org/10.1016/j.sigpro.2023.109134
0165-1684/

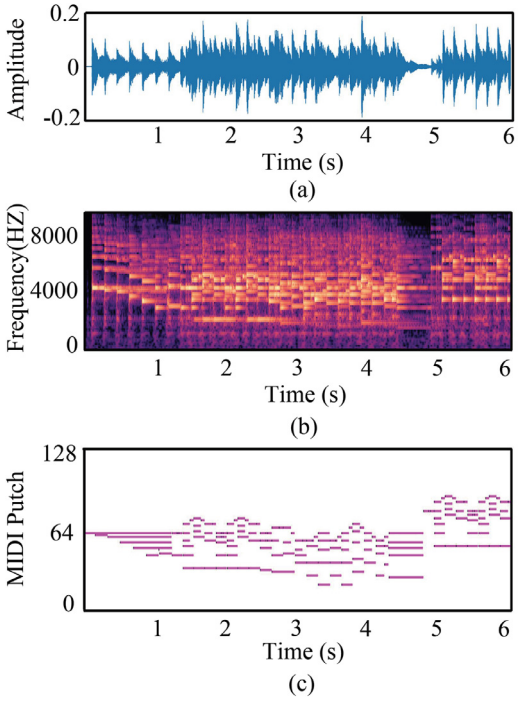

Fig. 1. Data transformation in AMT system: (a) input audio waveform, (b) audio time-frequency representation, and (c) output piano-roll representation.

teristics of music signals. By combining the time domain and frequency domain, it designs excellent classification features to achieve multi-note classification. The core idea of the statistical model is to establish the model of music signals, and the maximum posteriori probability is used to estimate the model parameters and iterate out the most significant multiple. Model based on spectral decomposition tries to decompose the original audio spectrum into a linear combination of notes, in which the monophonic spectrum is obtained through supervised learning. The deep learning-based model constructs a multi-layer network to realize end-to-end training. Usually, the music spectrum is used as input, and the transcription information can be obtained directly by training model parameters with a large number of samples.

The above four models use note periodicity and sparsity constraints to realize AMT and improve transcription accuracy. However, the problem of output space explosion is handled by limiting the maximum polyphony or giving the known number of polyphony in music, which is unreasonable in practical application. Current four models treat each note as an isolated object and predict the probability of occurrence of each note independently. Ignoring the interdependence between notes make these four models inherently limited. According to the music theory, notes are implicitly dependent on each other. Notes that form a fixed chord are more likely to occur together rather than occur independently.

To solve the problem of output space explosion, it is necessary to consider the interdependence between notes in the learning model. Inspired by the success of graph convolutional network (GCN) in unstructured data modeling [5], we use GCN to model the interdependence between notes. Each note is regarded as a node in the graph, while the topological connection between notes is represented by the edges between nodes to express the implicit dependence between notes. The proposed model firstly uses a series of convolutional neural network (CNN) and recurrent neural network (RNN) to extract the effective features of music signal. Then, the learning results of GCN network are mapped to a set of interdependent classifiers. Through joint training feature network and label network, the network can be trainable end-to-end. The contributions of this paper are as follows:

1. To solve the problem of output space explosion in polyphonic transcription, GCN is first introduced to model the interdependence between notes to limit the size of note combinations, and the visual experiments show that GCN can effectively learn the correlation between musical notes;
2. A novel CR-GCN hybrid model is proposed, which integrates a music language model based on GCN with an acoustic model for polyphonic music transcription. The hybrid model can make full use of the dependence between notes to effectively improve the transcription performance of polyphonic music;
3. Experimental results on several datasets show that our model can mine inter-frame co-occurrence notes well. Excellent performance has been achieved in both frame-level and note-level, especially for music with high polyphony, which is more competitive than previous methods.

The remaining part of this article is organized as follows. The second part introduces the relevant work of AMT in detail. The third part introduces the structure of the proposed model and the concrete realization of each part. The fourth section gives the experimental results and discussion, and finally gives some conclusions in the fifth chapter.

2. Related works

The *feature-based* model was originally proposed to achieve polyphonic music transcription. The type of model combined the human understanding of music. By introducing the prior knowledge of music signal [6], good time-frequency representation features were designed to improve the precision of musical note recognition.

The pitch of a note is directly related to the fundamental frequency of the signal, and most features were calculated from the frequency domain in earlier studies [7], such as the amplitude spectrum. However, the frequency domain features ignored the periodicity of the music signal [8]. Autocorrelation function (ACF) and logarithmic cepstrum [9] have been developed to extract periodic features, and have been widely used in monophonic detection [10].Su and Yang [11] devised a set of criteria combining time domain and frequency domain simultaneously. It detected pitches based on the consistency of harmonic orders in the frequency domain and subharmonic orders in the quefrency domain. Wang [12] investigated the application of robust harmonic features for classification-based pitch estimation. Based on energy intensity and spectral envelope shape, five types of robust harmonic features were proposed to reflect the harmonic structure related to pitch.

The *statistical model* adopted statistical framework to describe polyphonic music transcription problems. Given the observation frames and all fundamental frequency combination sets, multi-pitch estimation was regarded as the maximum posterior probability estimation problem of the given frame information [13]. Typical systems is harmonic timing sequence structure clustering [14]. If there is no prior information, this task could be modeled as a maximum likelihood estimation problem. Typical methods include spectral peak and non-peak region modeling [15], overtone spectral envelope modeling [16] and non-parametric Bayes modeling [17].

In the past decade, the most rapidly developing methods has been those *spectral decomposition-based*. By constructing dictionary matrix and activation matrix, Benetos and Weyde [18] decomposed the spectrum of music signal into linear combinations of multiple concurrent notes and their corresponding intensity or probability. In order to limit the number of notes that can be pronounced at the same time, regularization was introduced into the activation

matrix in [19] to achieve sparsity of pitch. Other extensions focused on dictionary matrix, where a clear template for each note could be pre-established by supervised learning [20]. Furthermore, multiple templates were built for a single note [21] to cope with the time-varying spectral characteristics of the note (the beginning of a note (or attack phase) might have entirely different spectral properties than the central part (decay phase)). Other typical spectral decomposition-based include probabilistic latent component analysis (PLCA) [22] and various improved methods [23].

Deep learning has been successfully applied in the field of image recognition, and *deep learning-based* method have also been tried in music transcription and music signal processing [24]. The musical signal has a strong timing, and the spectral properties of the notes evolve across input frames. LSTM [25] was the most commonly used acoustic network model because of its ability to model sequences in a compact manner. Bock [26] proposed the first successful system, the core idea of which is to use two spectrograms as inputs, which were processed by one (or more) long and short term storage (LSTM) layers. In addition to accurate acoustic models, other scholars have explored music language models to capture the temporal structure of music. Raczynski et al. [27] proposed a dynamic Bayesian network to estimate the prior probability of note combinations. Similarly, in [28], a music language model based on recursive neural network (RNN) was applied to estimate the prior probability of musical note sequences. In [29], a sequence transduction framework was proposed, in which the acoustic model and the music language model was combined in a single RNN. However, it is still a very challenging problem to model the large output space of polyphonic music. Recently, the current state-of-the-art model has been proposed [30], which combined two networks, with one network detecting the beginning of a note and the other one detecting the duration of a note frame. In the latest work, An arbitrary resolution transcription system was proposed in [31], and [32] proposed the HPPnet network with smaller parameters to achieve state-of-the-art performance.

The above four models achieve polyphonic music transcription from different perspectives, but there are still some areas worth improving. Feature-based model rely on music theory and signal processing background knowledge, which has high requirements for developers. Statistical model-based method need to estimate a large number of parameters due to the complexity of polyphony combinations. The spectral decomposition-based method is difficult to decompose the clear audio sequence, since the gradual attenuation effect of musical note signal in time cannot be considered. Deep learning-based constructs end-to-end network to estimate joint probability of simultaneously triggered notes. However, it is hard to learn the distribution of joint probability due to the large space of polyphony combinations.

There are other efforts focused on musical language models to model musical timing and correlations between pitch combinations. The language model of the AMT, arguably the most challenging task among MIR, have suffered because of the complexity of the musical language. [33] studied a probabilistic model based on a distribution estimator, which is based on a recurrent neural network and can detect the time dependence in high-dimensional sequences. Sigtia et al. [34] developed an end-to-end hybrid model, including an acoustic model and a musical language model. Acoustic models are used to estimate the probability of tones in an audio frame whereas language models model correlations between tone combinations. Wang et al. [35] proposed a musical language model that can simulate the temporal correlation between notes in musical sequences. Ycart et al. [36] developed a music language model based on LSTM, which can be trained on any MIDI data. An autoregressive model is proposed in [37], which can effectively learn the inter-state dependence of musical notes.

Our work is an improvement on [30]. Although the network in [30] decomposes the complex joint probability distribution into the probability distribution of start and frame activation, there is still a lot of notes combination to be pronounced during the estimation of each note frame. We believe that there is a musical principle between notes that are sounded at the same time. For example, multiple notes in a chord are more likely to be triggered at the same time, whereas certain notes will never be pressed at the same time due to the harmony of the music. There are complex interdependencies between the notes that are triggered. In this paper, we use graph structure to explore the interdependence between musical notes. With the successful application of GCN in unstructured data modeling [5], we use GCN to model the coupling structure between musical notes. We propose a new network model, CR-GCN, to improve the performance of AMT by reducing the space size of musical note combination.

3. Approach

In this section, we introduce the proposed CR-GCN model for polyphonic music transcription in detail. Firstly, the general design idea of the proposed model is explained. Secondly, basic knowledge of GCN network and how to apply it to AMT field are introduced. Finally, the detailed structure of each sub-network is described.

3.1. Overall framework of CR-GCN model

The difficulty of polyphonic music transcription lies in the dimensional explosion of output space. How to limit reasonably the size of combination space to improve transcription accuracy is an important research branch for improving the performance of AMT. In this paper, we use graph to model the interdependencies between notes in note space. Instead of independently estimating the probability of a single note in each frame, our model greatly compresses the space of note combinations by taking into account the implicit dependencies between notes.

As shown in Fig. 2, our model is mainly composed of two parts: feature learning network and label learning network. Feature learning network is an acoustic model composed of serialized CNNs and RNNs, whose specific details are consistent with the model in [30]. The feature network predicts the probability of multiple pitches in each frame. Label learning network is a music language model consisting of three-layer of stacked GCN, where each note is represented as a node in the graph. We design an adjacency matrix for node feature updating by counting the co-occurrence pattern of notes in the database. This prior knowledge is used to model correlations between note combinations.

The input of the feature network is the spectrum image of each frame, and its output is the feature map. The input of the label network is node feature of each note, which is obtained by one-hot encoding, and its output is the node feature encoded by GCN, which is mapped directly to a group of interdependent classifiers. In the training process of network, the feature network parameters and label network parameters are updated synchronously through the frame-level loss of audio. Since the mapping parameters of the classifier are shared among all notes, the classifier can retain the structure that is close to each other in the note feature space. That is, our model no longer estimates individual notes independently, but considers the implicit dependence of note combinations. During the test, the learned classifier is fixed, and the dot product of the feature map and classifier is calculated directly to estimate the pitch combination in each frame.

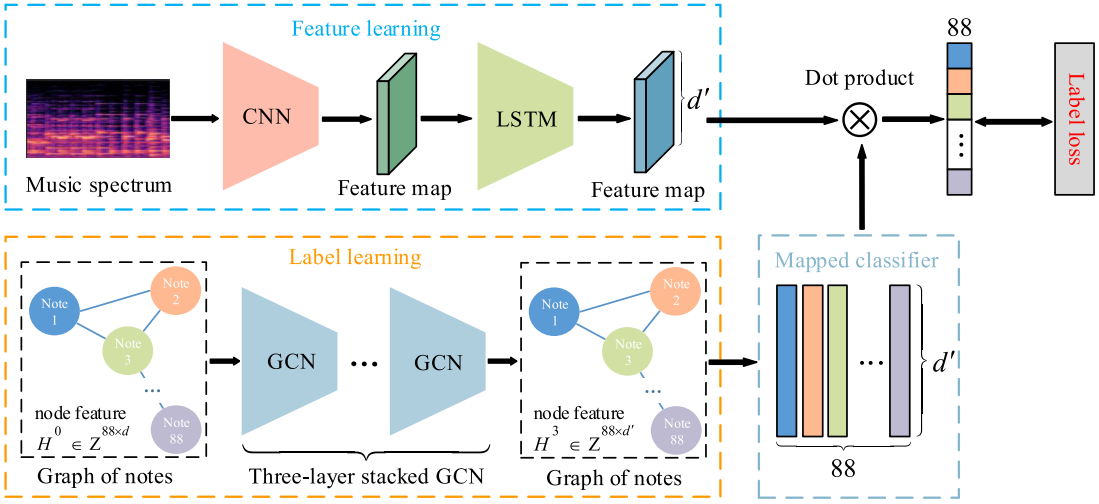

Fig. 2. Framework of CR-GCN model.

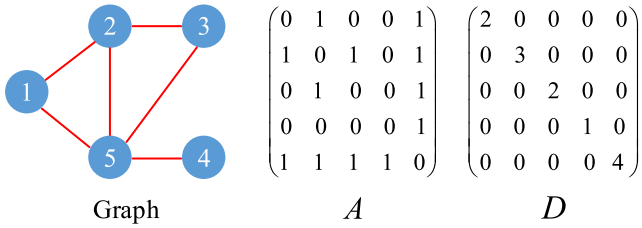

Fig. 3. Graph, adjacency matrix and degree matrix.

3.2. CR-GCN model for AMT

Graph convolutional networks are well known for their use in semi-supervised learning [38]. Similar to CNN, its essence is a feature extractor. The difference is that it deals with graph data instead of Euclidean data. GCN has cleverly designed a method to extract features from graph data, so that these features can be used for node classification, graph classification, edge prediction of graph data, and the embedded representation of graph can be obtained at the same time.

A graph can be described as $G = (V, E)$, where V represents the nodes in the graph, and E represents the connected edges between nodes. GCN has two inputs, one is the feature representation of each node $H \in \mathbb{Z}^{n\times d}$, and the other is the adjacency matrix between nodes $A \in \mathbb{Z}^{n\times n}$ (where n denotes the number of nodes, and d indicates the node feature dimension). Therefore, each layer of the GCN can be written as

$$H^{(l+1)} = \phi\,(H^l, A) \tag{1}$$

where $\phi\,(\cdot\cdot)$ is a nonlinear mapping function, that is, an information dissemination rule. The Laplace matrix of the graph is defined as

$$L = D - A \tag{2}$$

where L is the Laplace matrix, D is the degree matrix of the vertices (diagonal matrix), that is, the elements on the diagonal are the degrees of each vertex, and A is the adjacency matrix of the graph. An example is shown in Fig. 3. After applying the Symmetric Normalized Laplacian, $\phi\,(\cdot\cdot)$ can be represented by

$$H^{(l+1)} = h\,(D^{-1/2}\tilde{A}D^{-1/2}H^l\mathrm{w}^l) \tag{3}$$

where $\tilde{A} = A + I$, I is the identity matrix, that is, an adjacency matrix adding a self-join. $\mathrm{w}^l \in \mathbb{Z}^{d\times d'}$ is the weight matrix to be learned. $h(\cdot)$ a nonlinear activation function. GCN utilizes shared adjacency matrix A to transfer information between nodes, and integrates feature information and structural information to update node features of the next layer. In this way, node features are gradually abstracted in successive layers.

Our CR-GCN model is based on the stackable GCN. The original node output of GCN is the feature of each node after merging the information of other nodes. Different from that, we map the final output of GCN node to the classifier corresponding to music notes, and obtain the joint expression of features and labels. In addition, the graph structure (correlation matrix) is the basis for exchanging information in GCN network, which requirements are known, while the correlation between music notes is not clear. Thus, we need to devise a strategy for learning proper connections between notes from polyphony. CR-GCN model framework is described in Fig. 2, which mainly consists of feature learning module and label learning module.

3.2.1. Feature learning

The original input of audio signal is spectrum image representation, and we use CNN to extract potential notes due to its excellent feature extraction ability in two-dimensional images. In order to capture the timing of music signals, we further build the LSTM model. The image features output by CNN are the input of LSTM, and the rule of feature sequence in time sequence is learned on the basis of image features. Finally, an effective high-dimensional classification feature containing temporal and spatial information is obtained. we use *Librosa* to compute the Mel-scale representation of the original signal with 229 logarithmically-spaced frequency bins, a hop length of 512, an FFT window of 2048, and a sample rate of 16kHz. If an audio contains 480 frames, it will be represented as an image with resolution of 480×229. After the application of CNN network, we can obtain $96 \times 480 \times 57$ feature maps. Then, feature maps are flattened into one-dimensional, followed by a fully connected layer, and output image-level feature X with 480×768 resolution.

$$X = f_{fc}(f_{cnn}(I; \theta_{cnn})) \in \mathbb{Z}^{d'} \tag{4}$$

where θ_{cnn} indicates model parameters and $d' = 768$.

The output of CNN network is followed by an LSTM network layer with 128 units in both the forward and backward directions. Finally, there is a fully connected sigmoid layer with the same feature dimensions as X.

$$x = f_{sigmoid}(f_{lstm}(I; \theta_{lstm})) \in \mathbb{Z}^{d'} \tag{5}$$

3.2.2. Label learning

We use GCN to model the interdependence between notes and map it to the corresponding note classifier $W \in \mathbb{Z}^{c\times d'}$. Specifically, we applied three layers of stacked GCN, with one node for each note. Take the piano, the instrument with the widest range, for example, there are 88 notes in total, that is, 88 nodes ($c = 88$) are established in the graph. GCN takes the node representation of the previous layer H^l as input and updates the node representation of the next layer H^{l+1}.

For the first layer, the node feature vector is initialized as a identity matrix $H^0 \in \mathbb{Z}^{c\times d}$, where $d = c = 88$. That is, each node is assumed to be independent of each other.

For the last layer, the output is $H^3 \in \mathbb{Z}^{c\times d'}$, where d' is the feature dimension of the output node, which is consistent with the dimension after feature learning of the image. Finally, node features are mapped as classifiers $W \in \mathbb{Z}^{c\times d'}$, which directly act on the results of image feature learning x, and we can get the prediction result y of each note in the current frame.

$$y = Wx \tag{6}$$

Assuming that the actual label of the image is $\tilde{y}$, where $y^i = \{0, 1\}$ denotes whether note i appears in the current frame or not. There can be multiple notes in the same frame, so the whole network is trained using the traditional multi-label classification loss as follows

$$\ell = \sum_{i=1}^{c} \tilde{y}^i \log(\sigma(y^i)) + (1 - \tilde{y}^i) \log(1 - \sigma(y^i)) \tag{7}$$

where $\sigma(\cdot)$ is the sigmoid function.

3.3. Adjacency matrix of CR-GCN

The essential principle of GCN is to disseminate information between nodes based on adjacency matrix, which describes the graph relation between samples in probability graph. Therefore, the construction of reasonable adjacency matrix is the key problem of GCN. In most applications, the graph structure is known, i.e. the adjacency matrix A is known. However, as for music, an abstract and emotional work of art, its creation is quite random. There is no clear topology between notes.

In this paper, we use data-driven method to establish correlation matrix between notes. Usually, The machine learning approach implements classification or recognition tasks by exploring correlations between attributes and labels of samples in a data set. However, we believe that this is not enough. The information of label space in the dataset is further mined. We define the adjacency matrix by counting the co-occurrence patterns of notes within the dataset.

Specifically, we count the occurrence times of note pairs in each frame of audio and construct the co-occurrence matrix M. M_{ij} represents the times that note i and note j appear in the same frame. Both M_{ij} and M_{ji} are counted once the notes i and j are present together. Therefore, the co-occurrence matrix M is a symmetric matrix ($M_{ij} = M_{ji}$).

However, the frequency of occurrence of different notes varies greatly, by an order of magnitude. The co-occurrence matrix M is normalized to obtain the conditional probability matrix P.

$$P_{ij} = M_{ij} / \sum_{m=1}^{c} M_{im} \tag{8}$$

where c is the total number of notes. $\sum_{m=1}^{c} M_{im}$ denotes the occurrence times of note i in the dataset. P_{ij} denotes the probability of occurrence of note j when note i appears.

Even so, there are still shortcomings in using probability matrix to model note correlation. (1) Co-occurrence patterns of notes in different compositions may have different distributions, which may contain some random noises. (2) If the adjacency matrix overfits the training set, the generalization performance of the model will be reduced. Therefore, a threshold τ is adopted to binarize the probability matrix P. The two notes are related if P_{ij} is greater than the threshold, otherwise the correlation is considered false. Finally, note adjacency matrix A can be written as

$$A_{ij} = \begin{cases} 0, & \text{if } P_{ij} < \tau \\ 1, & \text{if } P_{ij} \geqslant \tau \end{cases}, \tag{9}$$

where A is the binary adjacency matrix.

In addition, threshold τ value can also be used to adjust the aggregation information ratio of nodes in the network. When τ approaches 0, the neighborhood node information in the network will be considered more. When the τ approaches 1, the feature of a node itself will account for more weight of the network.

4. Experiments

In this section, we first describe the implementation details of our experiment, and then give the evaluation indicators of polyphonic music transcription, and related description of the dataset. The experimental results of proposed method as well as comparison method are reported in detail. Finally, the visual analysis results are displayed.

4.1. Implementation details

Without otherwise stated, the original input audio sampling rate is 16kHz, and the signal is represented as a mel-scaled spectrograms with 229 logarithmically-spaced frequency bins, a hop length of 512 and an FFT window of 2048, and was fed into the feature learning layer. ReLU was adopted as a nonlinear activation function during feature learning, and the original feature of each frame of audio was mapped to feature representation in high-dimensional feature space ($d' = 768$). Raw audio is cut into 20 second segments for LSTM training, which makes network training consume less memory and can be learned more quickly.

There are two inputs for the label learning layer, one is the one-hot encoded unit vector of the node itself, and the other is the adjacency matrix A. The 88 nodes in the network correspond to the notes A0-C8 on the piano, which are finally mapped to 88 classifiers. Our CR-GCN is composed of three GCN layers, and the final node feature output dimension is 768. The threshold τ of adjacency matrix is set to 0.7. Adam Optimizer is used as the optimizer for network optimization, with an initial learning rate of 0.006, which attenuated to 0.98 times of the original value every 10,000 cycles, and the network is trained for 50,000 steps, for MAPS dataset and 150,000 steps for MAESTRO dataset. The network was implemented based on PyTorch and is trained on the RTX5000 GPU.

4.2. Evaluation metrics

Most AMT methods are evaluated using a set of metrics proposed for the MIREX multiple-F0 Estimation and Record Tracking Common Evaluation Task[1]. Three types of indicators are included: frame-level, note-level and stream-level transcription. We used the *mir_eval* library[2] to evaluate system performance based on frame-level and note-level metrics. Frame-based evaluation is done by

[1] http://www.music-ir.org/MIREX/

[2] https://craffel.github.io/mir_eval/

comparing transcribed binary output and MIDI truth frame by frame. Frame-based scores, $F-score$, *Precision* and *recall*, are calculated using standard metrics defined as follow.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1\_score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives.

For note-level evaluation, the system returns a list of notes, along with the corresponding pitch, onset, and offset times. Following [31], we calculate the note metrics, which requires the onset be within 50ms of ground truth and the offset resulting in note duration to be within 20% of ground truth or within 50ms, whichever is larger.

4.3. Dataset and reference methods

We evaluate our model on two different datasets, MAPS[3] and MAESTRO [4]. The training dataset and testing dataset of the two experiments were based on MAPS and MAESTRO respectively.

MAPS provides recordings with CD quality (16-bit, 44-kHz sampled stereo audio) and the related aligned MIDI files as ground truth. The overall size of the database is about 40GB. *i.e.* about 65 hours of audio recordings. The dataset consists of audio and corresponding notes for isolated sounds, chords, and complete piano music fragments. In our experiment, we only used complete music clips to train and test the CR-GCN neural network model. The dataset consists of 270 pieces of classical music and MIDI notes. According to different piano types and recording conditions, there are 9 kinds of recordings, each of which contains 30 pieces. 7 categories audio was produced by a software piano synthesizer, while 2 sets of recordings were obtained from yamaha Disklavier upright pianos. Thus, the dataset consists of 210 synthetic recordings and 60 real recordings. Like other scholars [3,30], we used synthetic music as the training set and real piano recordings as the test set to verify the practicability of the algorithm.

MAESTRO dataset V2.0.0 is a large-scale dataset containing paired audio recording and MIDI files to train and evaluate our proposed piano transcription system. The MAESTRO dataset contains piano recordings from the International Piano-e-Competition. Pianists performed on Yamaha Disklaviers concert-quality acoustic grand pianos integrated with high-precision MIDI capture and playback system. The MAESTRO dataset contains over 200 hours of solo piano recordings. Those audio recordings and MIDI files are aligned. Each music recording contains meta-information, including the composer, title, and year of the performance. MAESTRO dataset consists of training, validation and testing subsets.

In order to compare with the state-of-the-art algorithms, we select the most classical algorithms from four different solutions, including the summing harmonic amplitudes SHA[4], the spectral peaks and non-peak regions modeling SPNRM[5], the sound space-based spectral factorization (S3F, the source codes can be obtained from the authors) and the convolutional neural network and bidirectional long short-term memory Networks CBLSTM[6]. For the *feature-based* approach, we choose the SHA [6] as the reference method. And SPNRM [15] is the most typical *statistical model-based* method. S3F [18] is currently the most advanced *spectral decomposition-based* solution. CBLSTM [30] is the basis of our algorithm, a typical *deep learning-based* model. Further, three recent works, S2S[7] [29] and HRT[8][31] are reproduced. For HPPnet [32], we directly referenced the data based on the MAESTRO V2.0.0 dataset because there is no open source code to do a fair reproduction. As reference methods, the above 7 methods are tested together with our proposed model on the same dataset.

4.4. Results and discussions

In this subsection, The evaluation results and data analysis of each method are provided in detail. we calculated *Precision*, *Recall* and $F1$ of frame-level and note-level respectively.

4.4.1. Evaluation results for MAPS

The experimental results based on MAPS dataset are shown in Table 1. As far as the frame-level metrics are concerned, it can be seen that the S2S method achieves the highest score of 83.75% on $F1$ metric, followed by 83.51% of our CR-GCN method. All four deep learning-based methods scored highly. However, the overall score of the other three methods is relatively low, about 15% lower than the deep network. In terms of index *Recall*, the proposed CR-GCN model achieved the best result, reaching 77.92%. Another deep learning method, S2S, achieved 76.33%, and our method still improved by 1.6%. This is because our model takes the correlation between notes into account, which can mine more information about relevant notes when estimating the notes in each frame of audio. Therefore, our method can detect more real notes while maintaining accuracy.

As far as note-level indicators are concerned, the HRT achieves the highest score of 85.03% on Onset $F1$ and the S2S achieves the best performance when offset is considered. On both measures, our approach gets the second-best results, at 84.48% and 54.98%, respectively. By contrast, the overall result of considering the offset is not satisfactory. Once a note is triggered, its energy decays over time, making duration of notes difficult to detect. In general, compared with the other three methods, the deep learning model still has a great advantage in dealing with polyphonic piano transcription.

To compare the robustness of the proposed methods, boxplots of the seven methods on three frame_level indicators are drawn to provide a more intuitive representation, as shown in Fig. 4. As can be seen from the boxplot, except for CBLSTM and HRT, the boxes of the remaining 2 deep learning methods are flat, which means that the transcription results are relatively concentrated and those two models, S2S and CR-GCN, can achieve good stability. As for *Recall* indicators, CR-GCN has the highest average recall rate, in addition, its case coverage is the lowest, indicating the best robustness. In general, among the four deep learning methods, although S2S has achieved the highest $F1$ score, the CR-GCN proposed by us has the lowest variance and the best algorithm stability.

4.4.2. Evaluation results for MAESTRO

The experimental results based on MAESTRO dataset are shown in Table 2. Due to the great distance between deep learning models and traditional methods on MAPS dataset, we only list the experimental results of 5 deep learning algorithms on MAESTRO dataset. Although HPPnet achieved the best results. It is important to note that we did not reproduce HPPnet, but directly quoted the original data. Therefore, a perfectly fair comparison cannot be achieved.

[3] http://www.tsi.telecom-paristech.fr/aao/en/category/database/
[4] https://github.com/tiendung/mutiple-f0-estimation
[5] http://www2.ece.rochester.edu/projects/air/publications.html
[6] https://github.com/magenta/magenta
[7] https://goo.gl/magenta/seq2seq-piano-transcription-code
[8] https://github.com/bytedance/piano_transcription

Table 1
Transcription results evaluated on the MAPS dataset .

Method	Frame			Note			Note w/ offset		
	P	R	F1	P	R	F1	P	R	F1
SHA [6]	53.27%	71.65%	62.38%	64.75%	72.21%	68.53%	36.32%	41.65%	39.44%
SPNRM [15]	67.83%	59.81%	63.52%	70.47%	68.29%	69.38%	42.62%	40.27%	41.34%
S3F [18]	78.33%	52.26%	65.74%	76.78%	67.24%	72.06%	47.78%	41.26%	44.53%
CBLSTM [30]	88.23%	70.39%	77.84%	83.92%	80.16%	81.78%	51.09%	48.77%	50.34%
S2S [29]	**91.18%**	76.33%	**83.75%**	84.91%	83.75%	84.34%	**57.28%**	53.76%	**55.52%**
HRT [31]	87.64%	75.97%	81.85%	**86.25%**	83.82%	**85.03%**	56.64%	52.28%	54.46%
CR-GCN	90.42%	**77.92%**	83.51%	84.30%	**84.65%**	84.48%	55.17%	**54.82%**	54.98%

Table 2
Transcription results evaluated on the MAESTRO dataset .

Method	Frame			Note			Note w/ offset		
	P	R	F1	P	R	F1	P	R	F1
CBLSTM [30]	91.13%	88.76%	89.19%	97.42%	92.37%	94.84%	81.84%	77.66%	79.67%
S2S [29]	91.61%	93.29%	92.43%	98.11%	95.89%	96.95%	84.57%	83.23%	83.82%
HRT [31]	88.91%	90.28%	89.51%	**98.43%**	94.81%	96.61%	83.81%	80.7%	82.26%
HPPnet [32]	**92.36%**	93.46%	**92.86%**	98.31%	96.18%	**97.21%**	**85.36%**	83.54%	**84.41%**
CR-GCN	91.49%	**94.03%**	92.77%	97.38%	**96.21%**	96.88%	82.41%	**84.22%**	83.18%

Compared with the other three methods, our method achieves the best $F1$ score in both frame_level and note_level index.

Results from the MAESTRO dataset generally improved by 12% compared to the MAPS dataset. This is because the MAESTRO dataset is three times larger, and models trained on a large dataset have significant advantages. Our proposed method performs particularly well on three *recall* indices. Similar to the results on the MAPS dataset, our method detects as many real musical notes as possible, which is consistent with the original intention of our model design.

The boxplots of the 4 algorithms on the three indexes are drawn as shown in Fig. 5. As can be seen from the boxplot, as far as $F1$ is concerned, the average scores of the four deep-learning based algorithms are all very ideal. However, among the four algorithms, CBLSTM and HRT have the greatest difference in transcription results, followed by S2S. Our algorithm achieves the most stable transcription results. Therefore, the proposed CR-GCN method is very competitive.

4.4.3. Transcription results

We present the transcription results of a music fragment of 1500 frames (about 50s) in the MAPS dataset. More transcribed audio and source code is available[9]. As shown in Fig. 6, the abscissa of the graph represents the audio frame, and the ordinate represents which notes are triggered under the current audio frame, represented by MIDI digital coding, with a total of 88 notes.

As can be seen from the Fig. 6, the SHA algorithm transcribes a large number of notes, but most of them should not exist. In order to detect real notes as much as possible, the transcription sound like it contains a lot of noise. S3F focuses more on the detection of real notes, as shown in Fig. 6(c). Compared with ground truth, detection results miss a large number of real notes, and only a small number are detected successful. Transcribing audio can be acoustically monophonic, making it difficult to recognize chord hierarchies of notes. The result of SPNRM algorithm is like a compromise between SHA and S3F, with fewer false notes but missing some real notes. On the whole, four deep learning algorithms, CBLSTM, S2S, HRT and CR-GCN, get better results. Both results recognized almost all the real notes, and did not introduce many false notes. The transcription results are consistent with the musical hierarchy of the original audio.

Table 3
Ablation study .

Method		CR (without GCN)	CR-GCN
Frame	P	87.94%	**90.42%**
	R	70.02%	**77.92%**
	F1	77.27%	**83.51%**
Note	P	83.46%	**84.30%**
	R	79.97%	**84.65%**
	F1	81.57%	**84.48%**

In terms of frame-level transcription details, our method contains fewer octave errors. For example, at frame 350 and 740, the result of CBLSTM introduced the wrong note 76. HTR barely recognized the note 40 from 650 to 900 frames. S2S incorrectly transcribed note 40 as note 52 (octave error) between 600 to 1000 frames. The transcription results of CR-GCN method are closer to the ground truth. This proves that the extended GCN network can dig out more correlations between notes, so it contains more transcription details.

4.4.4. Ablation study

To verify the effectiveness of the GCN module in our model, we conducted an ablation experiment. We directly compare whether the performance is improved before and after the addition of GCN network. Hence, during the training of the ablation experiment, we froze the parameters of the GCN network. In this way, the network only contains the feature learning layer composed of CNN and RNN (CR without GCN model, as shown in the upper part of Fig. 2). Under the same initial conditions, the CR and CR-GCN models are trained based on MAPS dataset for 50,000 steps respectively, and the experimental results are shown in Table 3.

It is obvious from the Table 3 that transcription *recall* is significantly improved after GCN is added. Among the 6 evaluation indexes, all of them were improved after the addition of GCN module. Ablation experiments demonstrated that GCN can indeed im-

[9] https://github.com/XiaoZhecug/CR-GCN

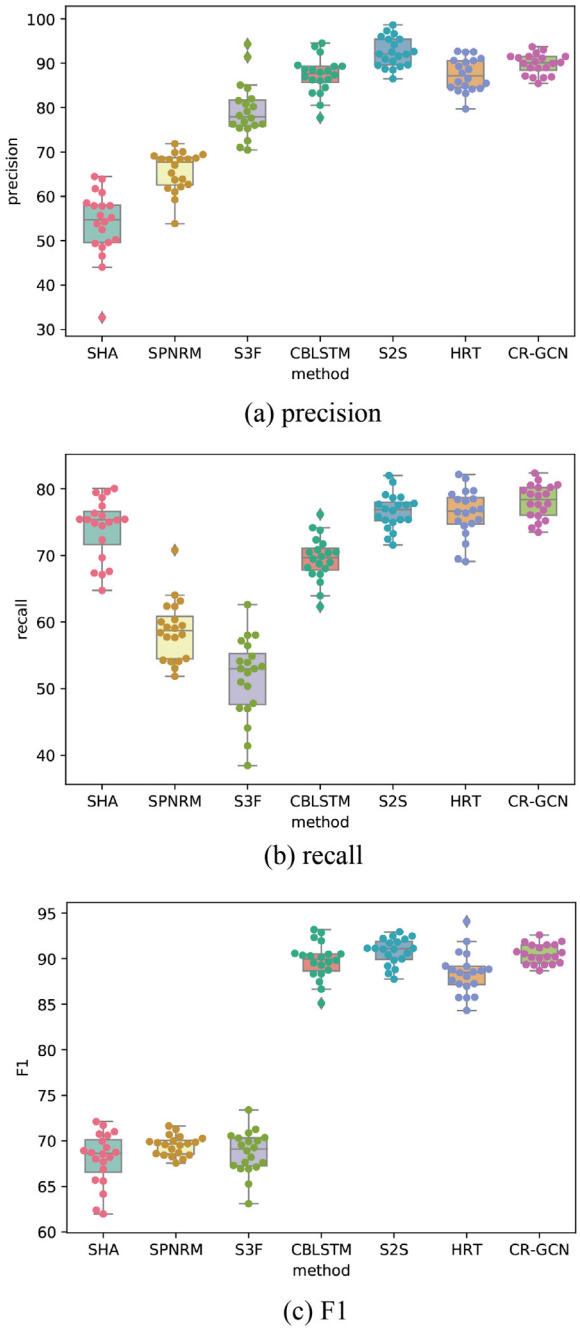

Fig. 4. Performance comparison of different methods for the MAPS dataset.

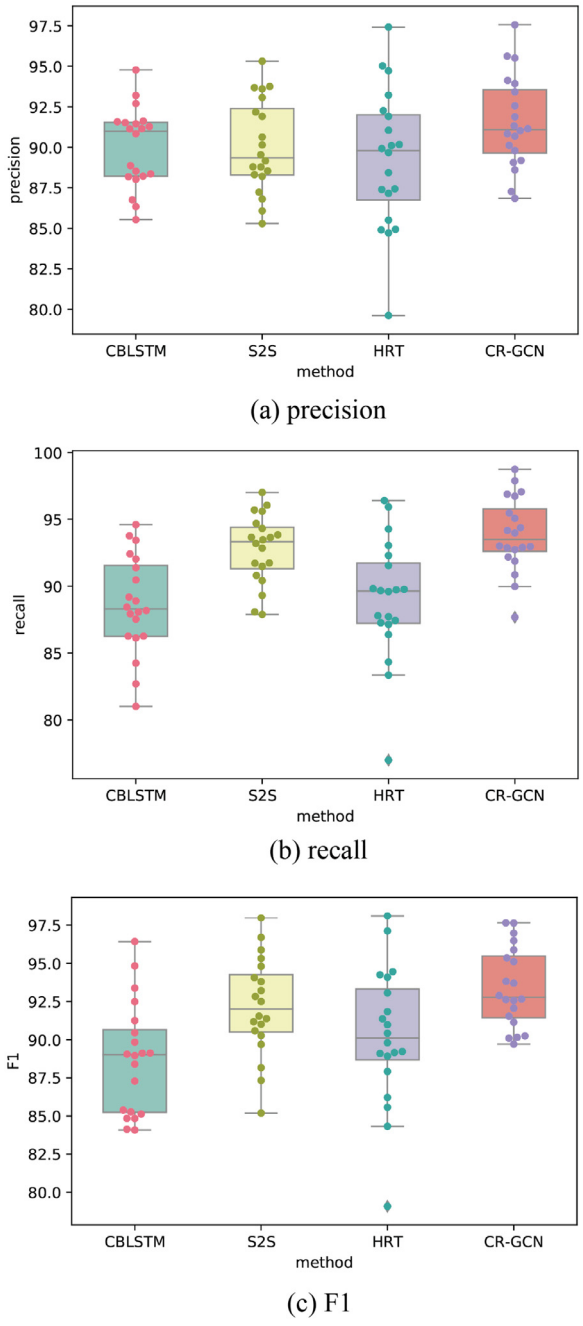

Fig. 5. Performance comparison of different methods for the MAESTRO dataset.

prove model performance by modeling the correlation between musical notes.

4.4.5. Effects of different threshold values τ

In order to explore the effect of the hyper parameter τ in Eq. (9), we change the threshold value τ to filter the correlation matrix. The experiment was conducted based on MAESTRO dataset and the experimental results are shown in the Fig. 7.

When τ gradually increased, $F1$ score of frame_level changed from 80% to 92%, showing unimodal change overall, first increasing and then decreasing. As mentioned before, when τ is set to a small value, there are too many adjacent nodes in the graph structure (the model will consider too much other node information, including adjacent nodes caused by statistical noise), resulting in low precision. When τ is set to a larger value, adjacent nodes will also be filtered out, which will also lead to performance degradation. The optimal value of τ is 0.7.

4.4.6. AMT for random chords

For the proposed CR-GCN model, GCN is used as a music language model to model the correlation between musical combinations. GCN draws on a priori knowledge of the adjacency between notes. However, the training dataset MAPS and MAESTRO mainly contain classical piano pieces. In other words, the adjacency between notes may only work for classical music. In order to explore the transcription effects of GCN on different genres of music, we completed transcription experiments on random chords.

The MAPS dataset contains a random chord dataset, the *RAND* set. This dataset provides chords composed of randomly selected notes. The pitches range from 21 to 108, polyphony levels vary

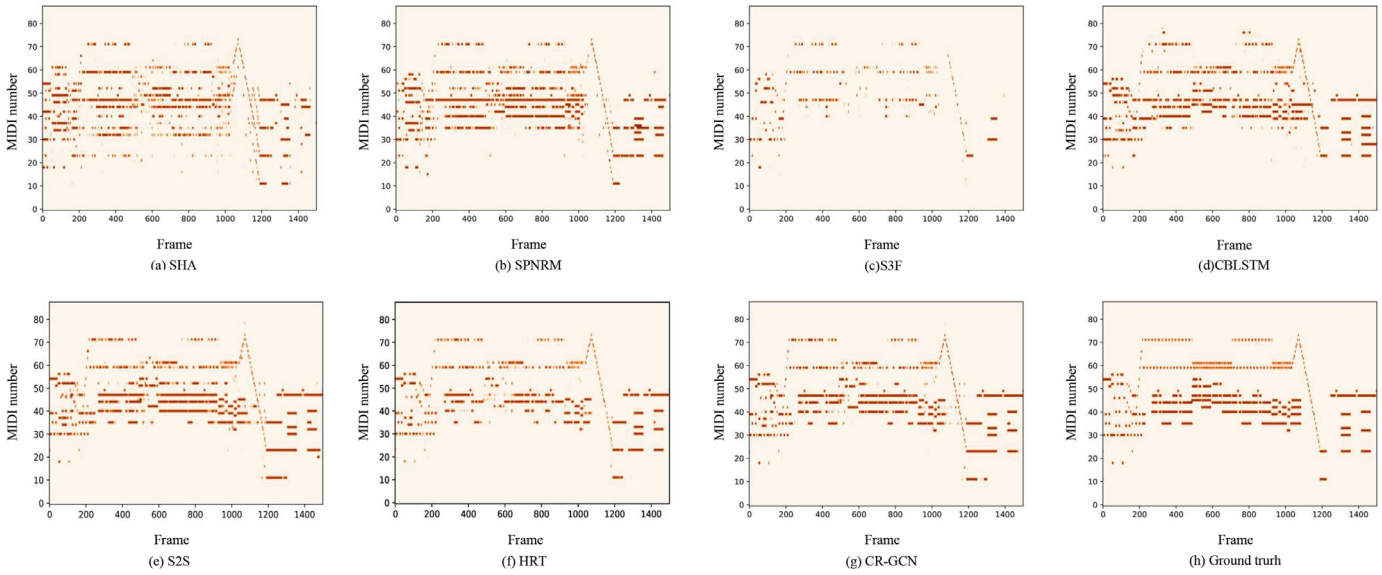

Fig. 6. Ground truth and the estimated multiple pitches by the compared methods for an example.

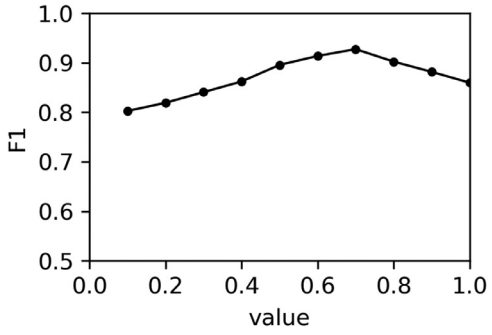

Fig. 7. $F1$ comparisons with different values of τ .

Table 4
The frame_level $F1$ scores for random chords .

Method	CR	CR-GCN
2-notes chords	**98.84%**	98.81%
3-notes chords	**94.34%**	94.02%
4-notes chords	**88.27%**	86.41%
5-notes chords	**79.69%**	75.18%
6-notes chords	**64.54%**	60.19%
7-notes chords	**42.63%**	35.12%

from 2 to 7, with 50 fragments per level, for a total of 300 random chord fragments. We compared experimental results on *RAND* dataset before and after the addition of GCN module. The frame_level $F1$ scores are shown in Table 4.

It can be seen from the Table 4 that, overall, the performance of CR-GCN is inferior to that of CR model. For chords composed of a small number of notes (2-notes chords, 3-notes chords), there was no significant difference in performance between the two models. However, when the polyphony level increases, the performance of CR-GCN model decreases more significantly. As explained in section III-C, when dealing with test sets with different distributions, the overfitting of the adjacency matrix to the data set will lead to the degradation of the generalization performance of the model. This is because the distribution of note combinations in random chords does not conform to the true distribution of the music. Our GCN language model cannot simulate the adjacency of notes in random chords. Therefore, the performance of our model is not satisfactory. The use of a language model does restrict the applicability of the transcription system to certain styles of music.

Table 5
Transcription results on music with a high level of polyphony .

Method		CBLSTM	S2S	HRT	CR-GCN
Frame	P	86.17%	**89.04%**	85.39%	88.28%
	R	66.77%	72.61%	71.34%	**75.92%**
	F1	74.92%	80.28%	78.96%	**81.35%**
Note	P	81.79%	82.88%	**84.25%**	82.53%
	R	76.63%	79.81%	79.97%	**82.17%**
	F1	78.69%	81.83%	82.18%	**82.83%**

4.4.7. Results for music with a high level of polyphony

We conducted experiments to explore the transcription limits of different algorithms on music with a high level of polyphony.

The polyphony of music pieces in the MAPS dataset ranges from 1 to 10, with an average polyphony of 3.65. We filtered out segments with polyphony numbers greater than 4 to obtain our new dataset of highly polyphonic music, with an average polyphony of 4.83. All models are trained based on the MAPS dataset, and the test results are shown in the Table 5.

Compared with Table 1, it is obviously to see from Table 5 that the performance of all algorithms decreases when processing highly polyphonic music transcription. Nonetheless, our algorithm has significant advantages, achieving the best results at both frame-level and note-level $F1$ scores.

Especially in terms of *Recall*, CR-GCN is much stronger than the other three models. This result is consistent with our model motivation. GCN is used to model correlations between notes, allowing our model to detect as many notes as possible within the same frame. In general, CR-GCN can improve music transcription performance well, and is more suitable for dealing with highly polyphonic music than the state-of-the-art models.

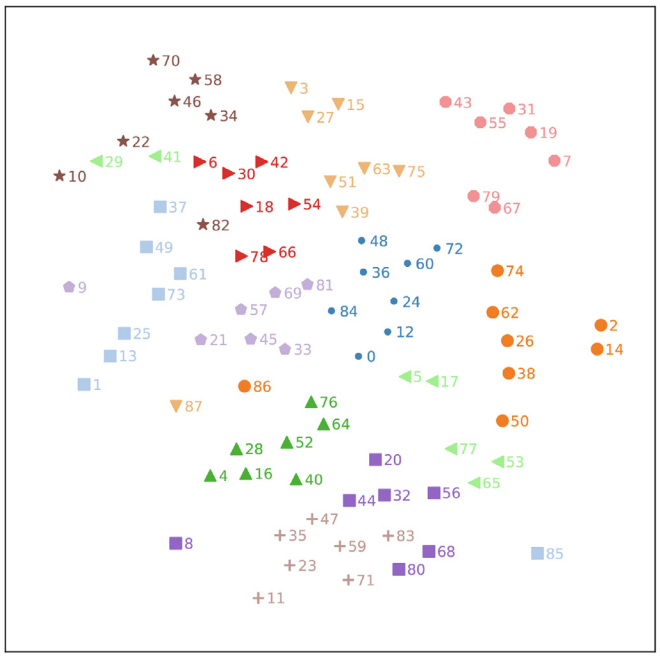

Fig. 8. Visualization of the CR-GCN model based on t-SNE.

4.5. Visualization analysis

To prove that our proposed model really achieves the modeling of note dependencies. The visual tool t-SNE [39] was used to analyze the learning results of GCN. t-SNE is an embedded model that can map data from a high-dimensional space to a low-dimensional space while preserving local properties of datasets. It is mainly used for reduction and visualization of high dimensional data.

We map the node features of 88 notes that are mapped to high-dimensional space after GCN training to two-dimensional space. As shown in Fig. 8, the numbers in the figure represent the notes corresponding to MIDI encoding, and the distance between notes represents the similarity of each note. Node features are initially one-hot encoding, that is, each note considered to be independent. After the 88-dimensional features are mapped to the two-dimensional space, the distances of all musical notes are basically equal. However, after dataset training, the GCN network maps its note features to higher dimensions.

It can be seen from Fig. 8 that there is an obvious interdependence between notes. Notes spaced an octave apart (2, 14, 26, 38, 50, 62, 74) meet closer together in two dimensions. This means that those notes are more similar and have implicit dependencies. The above phenomenon is also consistent with the musical theory that notes spaced an octave apart are important structures in forming chords, and therefore are more likely to occur simultaneously in the same audio frame.

The visualization results show that our model does capture the chord structure between notes and translates it to a set of interdependent classifiers to improve the performance of polyphonic music transcription.

5. Conclusion

Polyphonic piano transcription is the most challenging and difficult problem in AMT. The complex musical structure of polyphonic music causes the space explosion problem of note combination. Unlike previous work, they considered the likelihood of each note appearing to be independent. We propose a novel approach to modeling implicit dependencies between notes to deal with the output space explosions, which, to our best knowledge, has not been attempted for the AMT problem.

A new deep learning model called CR-GCN is proposed in this paper. Feature learning network and label learning network are established for sample and labels respectively. Among them, the feature learning network consists of CNN and RNN. At the same time, the label learning network is built based on three-layer GCN to learn the correlation between notes in the label space.

We verified the effectiveness of our approach on two public datasets, MAPS and MAESTRO, with 7 state-of-the-art algorithms. A series of experiments are carried out to show that our method can detect more notes that exist at the same time, and is superior to the existing methods in both frame-level and note-level indexes. Experiments on highly polyphonic music data sets prove that our method is more suitable for dealing with music with complex chords, which is consistent with our model motivation. Moreover, visual analysis shows that the interdependence between notes learned by CR-GCN has good interpretability in music theory. The disadvantage is that our method utilizes a prior knowledge of the adjacency between notes, which further limits the performance of our method for transcribing different genres of music. A combined model trained on both pitch classes and absolute pitches would reduce the dimensionality of the training data space, likely improving the performance of the label-learning method. We will continue to improve in the future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Zhe Xiao: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Visualization. **Xin Chen:** Writing – review & editing. **Li Zhou:** Visualization.

Data availability

Data will be made available on request.

Acknowledgment

This work is supported by the Technical Innovation Major Project of Hubei Province, China under Grant 2020AEA010, the Natural Science Foundation of Hubei Province, China under Grant 2020CFA031 and Grant 2019CFB581, and the 111 project under Grant B17040.

References

[1] E. Benetos, S. Dixon, Z. Duan, S. Ewert, Automatic music transcription: an overview, IEEE Signal Process Mag 36 (1) (2019) 20–30, doi:10.1109/MSP.2018.2869928.
[2] J.P. Bello, L. Daudet, M.B. Sandler, Automatic piano transcription using frequency and time-domain information, IEEE Trans Audio Speech Lang Process 14 (6) (2006) 2242–2251.
[3] W. Zhang, Z. Chen, F. Yin, Multi-pitch estimation of polyphonic music based on pseudo two-dimensional spectrum, IEEE/ACM Trans Audio Speech Lang Process 28 (2020) 2095–2108.
[4] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z.A. Huang, S. Dieleman, E. Elsen, J. Engel, D. Eck, Enabling factorized piano music modeling and generation with the maestro dataset, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, United states, 2019.
[5] M. Mesgaran, A. BenHamza, Anisotropic graph convolutional network for semi-supervised learning, IEEE Trans Multimedia (2020).
[6] A. Klapuri, Multiple fundamental frequency estimation by summing harmonic amplitudes, in: ISMIR, 2006, pp. 216–221.
[7] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, A. Klapuri, Automatic music transcription: challenges and future directions, J Intell Inf Syst 41 (3) (2013) 407–434.

[8] A. Klapuri, Multipitch analysis of polyphonic music and speech signals using an auditory model, IEEE Trans Audio Speech Lang Process 16 (2) (2008) 255–266.
[9] S. Kraft, U. Zolzer, Polyphonic pitch detection by iterative analysis of the autocorrelation function, in: DAFx, 2014, pp. 271–278.
[10] G. Peeters, Music pitch representation by periodicity measures based on combined temporal and spectral representations, in: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, volume 5, IEEE, 2006, pp. 1–4.
[11] L. Su, Y.-H. Yang, Combining spectral and temporal representations for multipitch estimation of polyphonic music, IEEE/ACM Trans Audio Speech Lang Process 23 (10) (2015) 1600–1612.
[12] D. Wang, C. Yu, J.H. Hansen, Robust harmonic features for classification-based pitch estimation, IEEE/ACM Trans Audio Speech Lang Process 25 (5) (2017) 952–964.
[13] M. Goto, A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals, Speech Commun 43 (4) (2004) 311–329.
[14] H. Kameoka, T. Nishimoto, S. Sagayama, A multipitch analyzer based on harmonic temporal structured clustering, IEEE Trans Audio Speech Lang Process 15 (3) (2007) 982–994.
[15] Z. Duan, B. Pardo, C. Zhang, Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions, IEEE Trans Audio Speech Lang Process 18 (8) (2010) 2121–2133.
[16] V. Emiya, R. Badeau, B. David, Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle, IEEE Trans Audio Speech Lang Process 18 (6) (2010) 1643–1654.
[17] K. Yoshii, M. Goto, A nonparametric Bayesian multipitch analyzer based on infinite latent harmonic allocation, IEEE Trans Audio Speech Lang Process 20 (3) (2012) 717–730.
[18] E. Benetos, T. Weyde, Multiple-f0 estimation and note tracking for mirex 2015 using a sound state-based spectrogram factorization model, 11th Annual Music Information Retrieval eXchange (MIREX). Malaga (2015) 1–3.
[19] A. Cogliati, Z. Duan, B. Wohlberg, Piano transcription with convolutional sparse lateral inhibition, IEEE Signal Process Lett 24 (4) (2017) 392–396.
[20] E. Vincent, N. Bertin, R. Badeau, Adaptive harmonic spectral decomposition for multiple pitch estimation, IEEE Trans Audio Speech Lang Process 18 (3) (2009) 528–537.
[21] S. Ewert, M. Sandler, Piano transcription in the studio using an extensible alternating directions framework, IEEE/ACM Trans Audio Speech Lang Process 24 (11) (2016) 1983–1997.
[22] E. Benetos, S. Dixon, A shift-invariant latent variable model for automatic music transcription, Comput. Music J. 36 (4) (2012) 81–94.
[23] C.-T. Lee, Y.-H. Yang, H.H. Chen, Multipitch estimation of piano music by exemplar-based sparse representation, IEEE Trans Multimedia 14 (3) (2012) 608–618.
[24] A. Cogliati, Z. Duan, B. Wohlberg, Context-dependent piano music transcription with convolutional sparse coding, IEEE/ACM Trans Audio Speech Lang Process 24 (12) (2016) 2218–2230.
[25] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.
[26] S. Block, M. Schedl, Polyphonic piano note transcription with recurrent neural networks, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2012, pp. 121–124.
[27] S.A. Raczynski, E. Vincent, S. Sagayama, Dynamic bayesian networks for symbolic polyphonic pitch modeling, IEEE Trans Audio Speech Lang Process 21 (9) (2013) 1830–1840.
[28] S. Sigtia, E. Benetos, S. Cherla, T. Weyde, A. Garcez, S. Dixon, Rnn-based music language models for improving automatic music transcription, in: 15th International Society for Music Information Retrieval Conference, 2014.
[29] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, J. Engel, Sequence-to-sequence piano transcription with transformers, arXiv (2021).
[30] C. Hawthorne, E. Elsen, J. Song, A. Roberts, Onsets and frames: Dual-objective piano transcription, in: Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, 2018, 2018, pp. 50–57.
[31] Q. Kong, B. Li, X. Song, Y. Wan, Y. Wang, High-resolution piano transcription with pedals by regressing onset and offset times, IEEE/ACM Trans Audio Speech Lang Process 29 (2021) 3707–3717.
[32] W. Wei, P. Li, Y. Yu, W. Li, HPPNet: modeling the harmonic structure and pitch invariance in piano transcription, arXiv preprint arXiv:2208.14339 (2022).
[33] N. Boulanger-Lewandowski, Y. Bengio, P. Vincent, Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription, arXiv preprint arXiv:1206.6392 (2012).
[34] S. Sigtia, E. Benetos, S. Dixon, An end-to-end neural network for polyphonic piano music transcription, IEEE/ACM Trans Audio Speech Lang Process 24 (5) (2016) 927–939.
[35] Q. Wang, R. Zhou, Y. Yan, Polyphonic piano transcription with a note-based music language model, Applied Sciences 8 (3) (2018) 470.
[36] A. Ycart, A. McLeod, E. Benetos, K. Yoshii, et al., Blending acoustic and language model predictions for automatic music transcription, in: 20th International Society for Music Information Retrieval Conference, 2019.
[37] T. Kwon, D. Jeong, J. Nam, Polyphonic piano transcription using autoregressive multi-state note model, arXiv preprint arXiv:2010.01104 (2020).
[38] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).
[39] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of machine learning research 9 (11) (2008).