Guan *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2022) 2022:7
https://doi.org/10.1186/s13636-022-00237-8

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Estimation of playable piano fingering by pitch-difference fingering match model

Xin Guan[1], Haoyue Zhao[1] and Qiang Li[1]*

Abstract

Most existing statistical models used to predict piano fingering apply explicit constraints among fingers and between fingers and notes; however, they disregard the relationship among notes. Furthermore, the state transfer matrix of HMM often makes the fingering of notes in compact scales unplayable without moving the hands. The direct adoption of notes interferes with mapping between specific pitches and the corresponding fingering. Inspired by human annotation and the note span constraints used in rule-based methods (in which fingering knowledge is acquired from span), we developed a model by which to match pitch difference and finger sequences (PdF). Playable fingering is achieved by combining learned finger-transfer knowledge with priori finger-transfer knowledge. The playability of the model was evaluated using a novel index, referred to as the irrational fingering rate (IFR). Experiment results demonstrate that the proposed model outperforms the third-order hidden Markov finger annotation model in terms of average match rate (by 4.06%) and highest match rate (by 2.87%). The proposed scheme also resolves the unplayable-without-hand-movement problem in compact scales.

Keywords: Piano playable fingering, Learned finger transfer knowledge, BI-LSTM, Pitch difference, Irrational fingering rate

1 Introduction

Musical performance on keyboards is largely influenced by hand movements, and especially the choice of finger. The ability to determine appropriate fingering is a necessary skill; however, it can be challenging for the novice and virtuoso alike. The annotation used to describe fingering is often derived via trial-and-error and therefore beyond the ability of individuals lacking extensive experience. Automating the process of fingering estimation is an emerging topic in the field of musical symbol processing. An understanding of musical symbol information has led to the development of instrument practice assistants[1], education systems [2], and music content arrangements [3].

Individuals differ in terms of the fingering strategies they employ. As a result, no single fingering strategy can be deemed optimal for every individual, regardless of hand size and shape. Musicians also differ in terms of the skills they emphasize, including precision, speed, phrasing, and dynamic articulation. The fact that sheet music can be performed using any number of finger sequences makes it fundamentally from the ground truth labels sought in many estimation tasks. Our objective in the current was to develop a reasonable fingering scheme (without unfeasible fingering) for presentation in piano textbooks.

Most automatic fingering estimation strategies are divided into rule-based and data-driven models. Rule-based methods adopt anatomic constraints and the difficulty in performing finger pairs as a cost function of finger sequences in order to identify an optimal sequence [1, 4–8]. This approach is prone to conflict among the various rules for fingering, and setting weights by which to measure constraints is not a trivial matter. Data-driven schemes learn the parameters of note sequences and corresponding fingering through the use of statistical models [9–12]. Note that this depends heavily on the quality and size of the dataset and degree to which the model reflects the relationship between the musical context and finger sequence. The fact that statistical models can also account

*Correspondence: liqiang@tju.edu.cn
[1] School of Microelectronics, Tianjin University, Tianjin, China

for anatomic constraints means that with sufficient data, they should be able to outperform rule-based schemes in terms of annotation accuracy [9].

Finger selection depends mainly on the interval between neighboring pitch pairs [4]. This approach is meant to reduce the degree of variation in musical symbol sequences, and thereby reducing model complexity, lessening the constraints on dataset size, and improving matching accuracy.

Finger transfer can also be constrained when the note sequence is ascending or descending without hand movement. Take for example, two descending notes on the right hand, where it is possible to transition from finger 1 to 2; however, it is not possible without moving the hand to transition from finger 2 to 3. In this paper, we refer to impossible sequences as unplayable. When dealing with unplayable fingering sequences, it is not possible to link finger pairs using the conventional long short-term memory (LSTM) networks. Nonetheless, LSTM networks can include long-range pitch interval information as well as the relationship between the pitch interval and the fingering. In the current study, we sought to improve model performance by adding a new layer that uses finger transfer knowledge.

To overcome the limitations imposed by small datasets, we developed a data augmentation scheme based on pitch-difference fingering (PdF) statistics from the existing dataset. In evaluating fingering sequences, we also emphasized the importance of playability to ensure that the results are indeed practical, as determined using a novel metric referred to as fingering unplayability. Experiment results demonstrated that our model improves the match rate of the fingering estimation.

Our main contributions are as follows:

1. We developed a pitch-difference fingering (PdF) model. Compared with the pitch sequence and fingering (PF), the PdF is more conducive to network fitting in establishing the relationship between notes and fingering.

2. We developed a novel approach to augment datasets by portraying data through the use of a hidden Markov model.

3. We introduce a novel metric by which to evaluate fingering annotation results in terms of fingering playability.

The remainder of this paper is outlined as follows. Section 2 presents a review of existing piano fingering annotation methods. Section 3 outlines the proposed method for piano fingering estimation. Evaluation results are presented in Section 4, and a summary is presented in Section 5.

2 Related works

Since 1997, researchers have addressed the problem of fingering estimation using methods based on rules [1, 4–8] or data [9–12].

Parncutt et al. [4]developed a model for note segments of finite length played by the right hand. They established a rule-based approach to fingering logic with a focus on maintaining a comfortable span between the fingers and notes. For longer sequences, they employed dynamic programming with a rule-based cost function by which to derive the path with the lowest fingering cost. Note that their approach requires the manual adjustment of rule weights to achieve suitable results. Jacobs et al. [5] improved that model by replacing semitone measurements with the physical distance to reduce the likelihood of erroneous rankings. Nellåker E et al. [6] introduced an additional pause-based rule based on the 12 rules established by Parncutt et al. Lin CC et al. [1] designed a sliced fingering generation (SFG) fingering generation algorithm, in which the score is split into cross-content segments. They employed the rules suggested by Parncutt et al. [4] to define the cost function in conjunction with dynamic programming to generate piano fingering in real time. Hart et al. [7] developed a dynamic programming method for the right-hand segment as a state transition constraint. Al Kasimi et al. [8] defined the horizontal cost of adjacent notes and the vertical cost of chords, wherein the grid graph generated by music fragments is used to find a path. The effectiveness of their approach has been demonstrated; however, their model has compatibility issues and lacks criteria by which to perform quantitative evaluations. Rule-based methods based on note spans rather than pitch inspired us to replace pitch with pitch difference.

The use of rules or costs facilitates a logical understanding of fingering; however, the parameters must be modified for every musical score.

Yonebayashi et al. [10] were the first to employ statistical models of pitch and fingering, in which a hidden Markov model (HMM) was used to model the fingering sequence. The probability of a given fingering state occurring depends on the fingering state in the previous time and the note output in the current [10]. Note however that independent observations do not allow for the inclusion of adjacent pitches for use in constraining the probability of finger transitions. Nakamura et al. [11] proposed a “merged HMM” to automate the separation and labeling of undifferentiated left-hand and right-hand fractions. Li Qiang et al. [12]combined fingering rules with a judgment function to improve the optimization rules used in the Viterbi algorithm. Nakamura et al. [9] constructed first public piano fingering dataset and two hidden Markov models with higher-order extensions.

Conventional statistical models enable the optimization of parameters based on patterns observed in the data; however, they focus exclusively on the local fingering constraints of continuous notes, with the result that much of the important information (e.g., long-range fingering relationships) is disregarded. Furthermore, highly adaptability models are prone to higher error rates when special fingerings appear.

Nakamura et al. [9] constructed two deep neural networks (DNNs), using pitch sequence and fingering (PF). The input is a sequence of integer pitches and the output is the corresponding fingering numbers. They employed a feedforward (FF) network and a long short-term memory (LSTM) network to estimate piano fingerings. The accuracy of this approach is slightly lower than that of statistical models, due to the fact that their FF and LSTM networks did not have constraints between output layer units. Thus, their model disregards the degree of dependency between fingerings, and there is no way to distinguish between monophonic and polyphonic notes, leading to finger reusing and finger crossing of chords.

3 Methods

As shown in Fig. 1, the proposed PdF annotation network has a recursive structure with three layers. The first layer converts notes into pitch differences. The middle layer takes pitch differences and chord information as inputs for the BI-LSTM like network. The third layer implements finger transfer rules. After training, priori knowledge pertaining to fingering can be used eliminate impossible fingering paths, avoid cross fingering in chords, and ensure playability. The parenthesized superscripts in Fig. 1 indicate their time step.

3.1 Pitch difference representation to differentiate between monophony from polyphony

For most people, the choice of fingering tends to focus more on changes in pitch rather than the pitches themselves. The original pitches are first converted into pitch differences. Note however that the conventional approach to measuring the intervals (i.e., pitch distance) between two notes gives no clue as to whether the notes are monophonic or polyphonic, which can have a profound effect on the choice of fingering. Thus, we sought to develop methods by which to express the same interval in terms of melody or harmony.

From a piano score with fingering label, we extracted pitch information pertaining to each note as well as the start time, offset time, hand information, and fingering. After separating left-hand data from right-hand data, we combine the pitch start and offset time and convert the pitch into a MIDI number $x^{(t)}$. The pitches are then expanded in chronological order to form a sequence, wherein the order of harmonic notes proceeds from low pitch to high (Fig. 2). Finally, we calculate the pitch difference $d^{(t)}$ as follows:

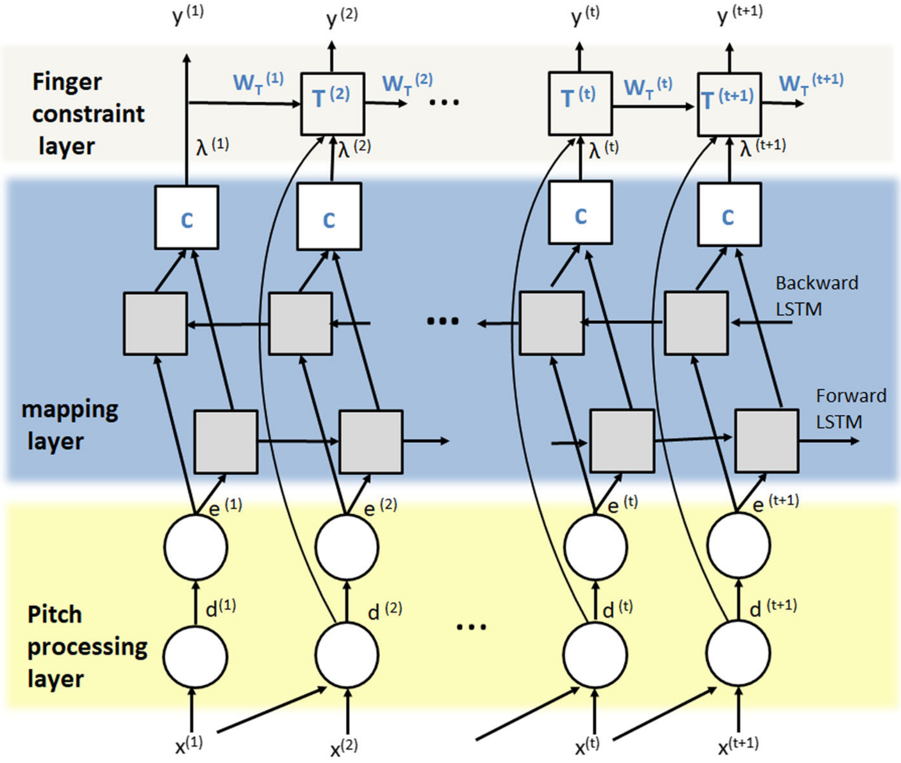

Fig. 1 Pitch-difference fingering annotation network

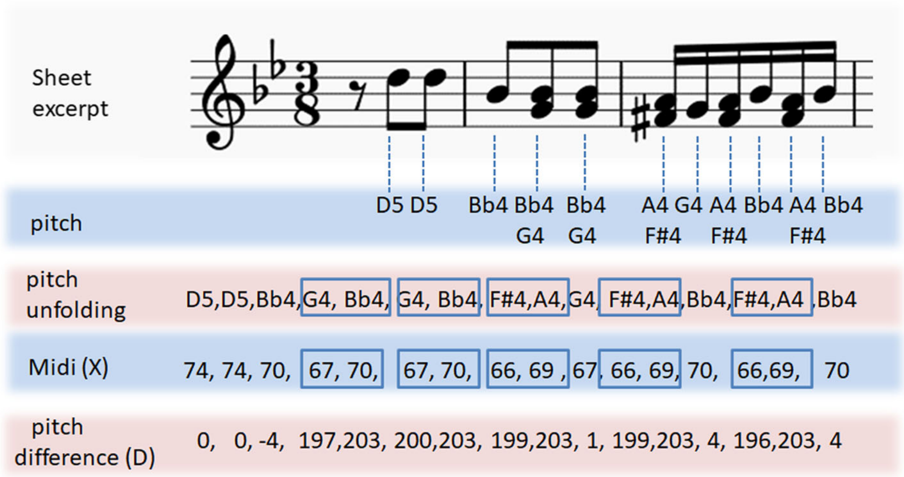

Fig. 2 Computing of pitch difference sequence excerpted from a score

$$d^{(t)} = \begin{cases} 100n & t = 1 \\ x^{(t)} - x^{(t-1)} + 100n & \left| x^{(t)} - x^{(t-1)} \right| < 12, t > 1 \\ 80sgn\left(x^{(t)} - x^{(t-1)}\right) & \left| x^{(t)} - x^{(t-1)} \right| \geq 12, t > 1 \end{cases} \tag{1}$$

where n represents the number of notes with the same onset. For a single tone, n equals 0.

In the first time step, if the initial sound is a single tone, then $dt = 0$; however, if it is polyphonous, then dt is the number of notes in the polyphony multiplied by 100.

When the interval exceeds one octave (i.e., the MIDI difference is equal to or greater than 12), then the fingering is relatively simple. The specific difference of MIDI numbers has little effect on fingering estimation. In this situation, $d^{(t)}$ represents the ascending or descending scale, and no longer represents the specific MIDI difference, and is uniformly represented as 80. Before this is input into the model, $d^{(t)}$ is integer encoded into a vector $e^{(t)}$.

An example of conversion is presented in 2.

3.2 Mapping of pitch differences to fingering

This layer constitutes a BI-LSTM-like network, in which the two LSTM parts capture the relationship between pitch differences and fingerings, and the remaining part is used to estimate the probabilities of given fingerings. In the forward and backward LSTM network, the basic unit is the LSTM cell, the internal gates of which can account for relationships among different pitches over a long-range to inform the process of estimating fingerings. The long-range context is critical in situations involving cross-fingering in ascending or descending scales or cases where fingers must be changed to enable the rapid repetition of notes.

3.2.1 BI-LSTM

Forward and backward LSTMs are used to learn note-finger relationships. For specific notes, fingering depends on the current note and previous note as well as the notes that follow [4]; therefore, we adopted a bi-directional network to determine the relative positions of fingers.

The input of the LSTM cell is a splicing of the hidden state in the previous time-step $h^{(t-1)}$ and the current pitch difference $e^{(t)}$. The output is its hidden state $h^{(t)}$. For the basic unit of the LSTM, as shown in Fig. 3. The three gates used to realize the function of the LSTM are the forget gate, input gate and output gate [13]. The internal function of fingering estimation can be embodied as follows. The forget gate is used to control the degree to which the previous difference between pitch and fingering affects the current fingering estimate. The input gate controls the degree to which the current difference in pitch affects fingering. Cell state $C^{(t)}$, also called long-term memory, is a combination of processed information by forget and input gate and previous memory $C^{(t-1)}$. The output gate retains some input information as short-term memory. The hidden state is then updated in accordance with long-term memory and short-term memory.

BI-LSTM combines forward and backward LSTM, where backward LSTM refers to the forward LSTM with flipped inputs. The output of BI-LSTM $\left[h^{(t)}_{(\text{forward})}, h^{(t)}_{(\text{backward})}\right]$is a concatenation of the forward and backward hidden vectors with the same size hidden_size.

3.2.2 Feature linear compression

To facilitate characterization of fingering estimation results, the hidden state vector of 2 × *hidden_size* is mapped to k dimension using linear layer c, where k refers

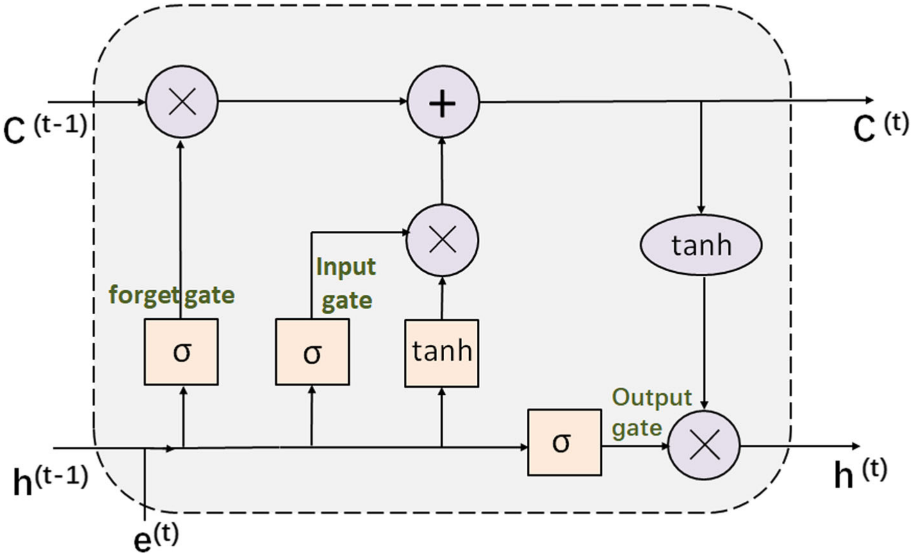

Fig. 3 Structure of the LSTM cell

to the number of fingering labels and $\lambda^{(t)}$ indicates the probability of each fingering label occurring in the current input. In our method, the left and right hands are independent, and k is 5.

$$\lambda^{(t)} = c\left[h^{(t)}_{(\text{forward})}, h^{(t)}_{(\text{backward})}\right] \tag{2}$$

$$\lambda^{(t)} = \left(\lambda^{(t)}_1, \lambda^{(t)}_2, \lambda^{(t)}_3, \ldots, \lambda^{(t)}_k\right)^T \tag{3}$$

$\lambda^{(t)}_j$ indicates the probability of finger j occurring at time t, as estimated using the BI-LSTM model.

3.3 Learned fingering transfer knowledge

This layer is added to the output of the BI-LSTM. Note that the BI-LSTM model is able to process long-range contextual information pertaining to note sequences and learn the correspondence between musical notes and fingering; however, it cannot be used to represent ergonomic constraints among fingers. We therefore introduced learned fingering transfer knowledge and a priori knowledge of fingering to constrain two adjacent fingerings in the output of BI-LSTM.

Finger transitions are related to the ascending and descending of neighboring monophonies; therefore, we introduced various fingering transfer matrices to the BI-LSTM output based on the input type, as follows:

$$W_T = \frac{1}{2}\left[\left(W_{T\uparrow} + W_{T\downarrow}\right) + sgn\left(d^{(t)}\right) \cdot \left(W_{T\uparrow} - W_{T\downarrow}\right)\right] \tag{4}$$

where $W_{T\uparrow}$ and $W_{T\downarrow}$ are the finger transition probability matrices of ascending and descending pitches, respectively. Note that $W_{T\uparrow}$ and $W_{T\downarrow}$ restrict only the fingering used for the adjacent single tone. In the constraint matrix W_T in Fig. 4, element P_{ij} indicates the probability of transferring from finger f_i to f_j.

Following the addition of W_T, output Y is expressed as follows:

$$y^{(t)} = W_T \cdot y^{(t-1)} + \lambda^{(t)} \tag{5}$$

$$Y = \left(y^{(1)}, y^{(2)}, \ldots y^{(n)}\right) \tag{6}$$

Note that $y^{(1)}_i = \lambda^{(1)}_i = \Lambda(1, i)$, $y^{(t)}$ can also be expressed as the follows:

$$y^{(t)} = \left(y^{(t)}_1, y^{(t)}_2, y^{(t)}_3, \ldots, y^{(t)}_k\right)^T \tag{7}$$

where $y^{(t)}_i$ indicates the occurrence probability of the fingering label with index i in the tth position of the score. At time t, the most likely fingering $\phi^{(t)}$ for the current pitch is derived as follows:

$$\phi^{(t)} = \arg\max_i \left[y^{(t)}_i\right] \tag{8}$$

3.4 Priori knowledge of finger transfer

Under the assumption that fingers (other than the thumb) cannot cross over each other, then the kinematic characteristic of the left-hand and right-hand fingers makes some fingerings impossible without movement, as shown in 1.

Table 1 lists the possible (✓) and impossible (x) finger transitions for ascending scale by the left-hand and descending scales by the right-hand when playing monophonic note pair where the hand does not move. Note that the descending scale (left-hand) and ascending scale (right-hand) are diagonally mirror-symmetric.

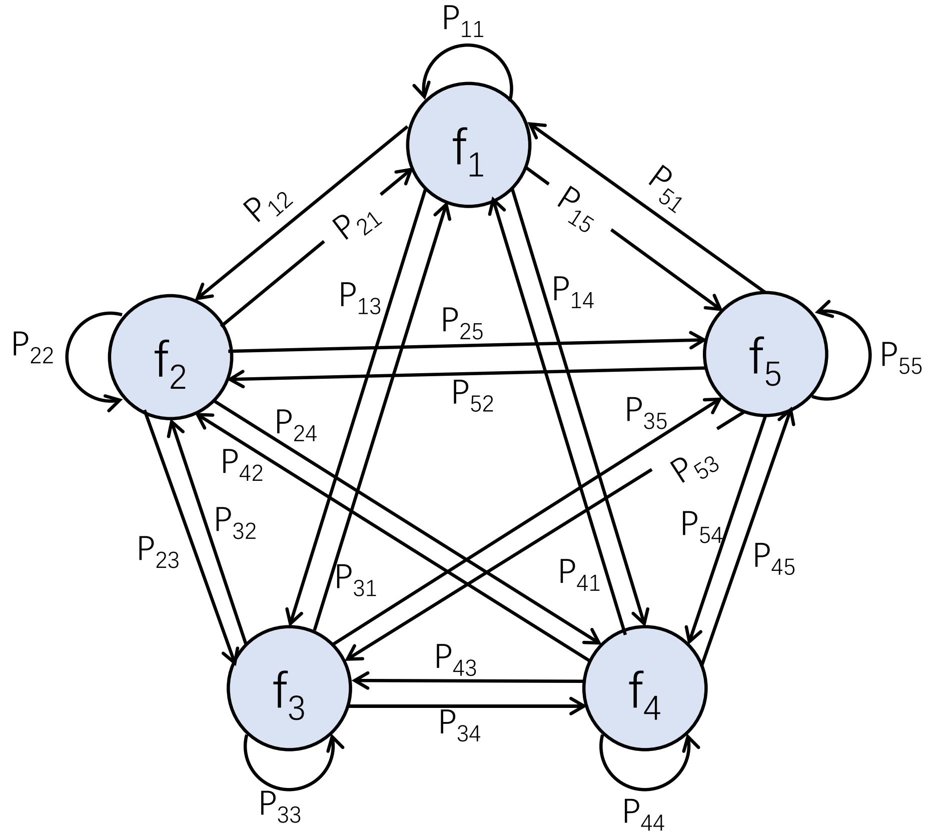

Fig. 4 Schematic diagram showing all possible finger transfers

Decision function T is based on the data in Table 1, as shown in (9). This also imposes restrictions on the fingering of chords, due to the prohibition on finger crossing and repetition. The right-hand maximum span [4] of 5 limits the scope of T in the monophonic type. Due to the mirror-symmetry in the physiological structure of the two hands, we can assume that the maximum span also applies to the left hand.

In cases where the current fingering is impossible, the probability of transition is set to 0; otherwise, we retain the current output, as follows:

$$T=\begin{cases} \frac{1}{2}\left\{sgn\left[5-f^{(t-1)}\cdot f^{(t)}\right]+1\right\} & \begin{array}{l} -6<d^{(t)}<6 \quad \text{and} \\ \text{left}: d^{(t)}\cdot\left(f^{(t)}-f^{(t-1)}\right)>0 \\ \text{right}: d^{(t)}\cdot\left(f^{(t)}-f^{(t-1)}\right)<0 \end{array} \\ 0 & \begin{array}{l} \text{left}-\text{hand} \quad \text{chord}: \\ \qquad f^{(t)}\leq f^{(t+1)} \\ \text{right}-\text{hand} \quad \text{chord}: \\ \qquad f^{(t)}\geq f^{(t+1)} \end{array} \\ 1 & \text{other} \end{cases} \tag{9}$$

As shown in Fig. 5, after adding T, some fingering transfer paths are trimmed.

In the prediction phase, a priori knowledge is used to ensure that all proposed fingerings are indeed playable. The score assigned to the candidate fingering $y^{(t)}$ is derived as follows:

$$y^{(t)}=T\cdot W_T\cdot y^{(t-1)}+\lambda^{(t)} \tag{10}$$

At time t, we select the fingering with the highest probability as the result for the current pitch as follows:

$$\phi^{(t)}=\arg\max_i\left[y^{(t)}\right] \tag{11}$$

Table 1 Possible (✓)and impossible(x) finger transitions for ascending scale by the left-hand and descending scales by the right-hand in which the hand does not move

$f^{(t-1)}/f^{(t)}$	1	2	3	4	5
1	✓	✓	✓	✓	x
2	✓	✓	x	x	x
3	✓	✓	✓	x	x
4	✓	✓	✓	✓	x
5	✓	✓	✓	✓	✓

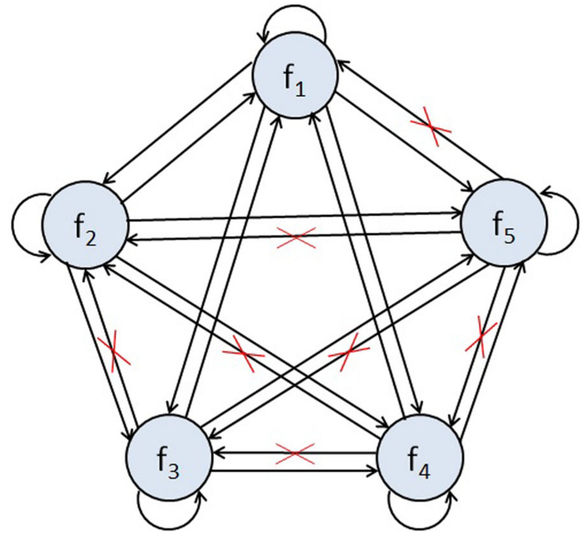

Fig. 5 Schematic diagram showing fingering transfer path when *T* is added

3.5 Data augmentation

Dataset augmentation is based on the HMM model in [16]. As in the HMM-based piano fingering estimation method, we treat fingerings as hidden states and pitch differences as observation. Initial state π is the probability of the initial fingering. The transition probability matrix $A = A\left[a_{i,j}\right]$ statistically describes the dependence between neighboring fingers. $a_{i,j}$ is the element of A, where $a_{i,j} = P\left(f_j|f_i\right)$ represents the probability of finger f_j, given the preceding finger f_i. The emission probability matrix $B = B\left[b_i(pd_m)\right]$ refers to the relationship between fingering and pitch difference. $b_i(pd_m)$ is an element of B, where $b_i(pd_m) = P\left(pd_m|f_i\right)$ indicates the probability of pitch difference pd_m given finger f_i. To ensure the playability of the generated chord fingerings, the fingerings of chords in the original data are calculated as a whole. When the frequency of f_i to pd_m is greater than 5%, it is counted as $b_i(pd_m)$; otherwise, $b_i(pd_m)$ is 0. The statistical content does not include fingerings with long-term connections, such as homophonic swapping. We also restrict the continuous appearance of the same fingering. In other words, when three consecutive single-tone fingerings are generated, the initial fingering will be re-selected.

Figure 6 presents an example of dataset augmentation. Figure 6 illustrates the fingering generation process for the left-hand when the current finger is 3. The process is as follows: First, select the next fingering in accordance with the finger transition matrix. The next fingering is selected as a polyphony fingering 1–5. Note that in the figure, the 6 × 6 white grid represents the current and next fingering numbers, where C represents all polyphonic fingering combinations in the dataset. There are many such combinations, the collection of which is denoted as C. Note that these combinations exist as separate indexes in the actual finger transformation matrix. In accordance with the emission matrix, the selector adds corresponding pitch difference information to the extended set sequence. This process is repeated until the length of the fingering sequence reaches a specified length.

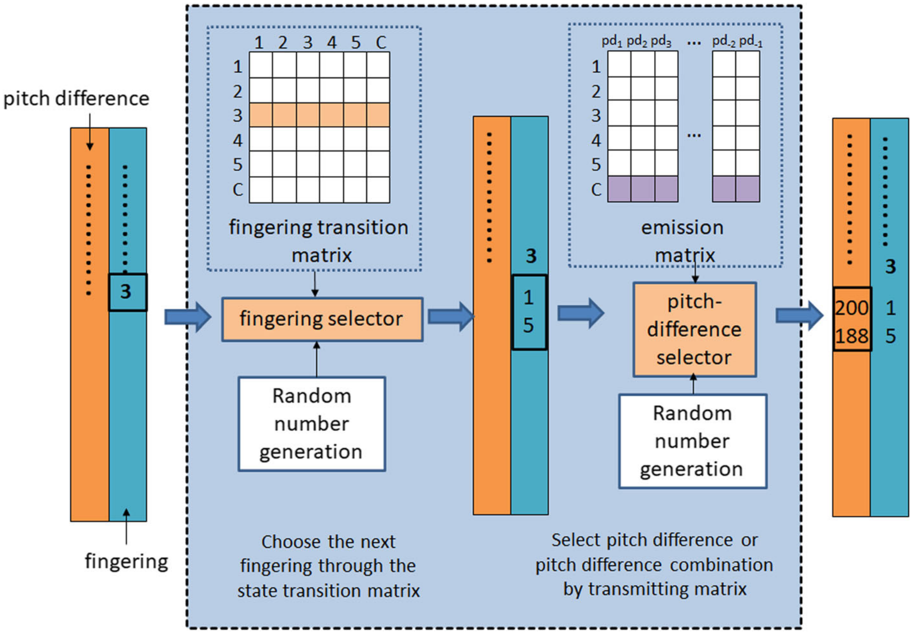
pitch difference
3
fingering
1 2 3 4 5 C
1
2
3
4
5
C
fingering transition matrix
fingering selector
Random number generation
Choose the next fingering through the state transition matrix
3
1
5
pd$_1$ pd$_2$ pd$_3$ ... pd$_{-2}$ pd$_{-1}$
1
2
3
4
5
C
...
emission matrix
pitch-difference selector
Random number generation
Select pitch difference or pitch difference combination by transmitting matrix
200
188
3
1
5

Fig. 6 Principle of the dataset augmentation

4 Evaluation

4.1 Setup

Experiment data was obtained from the Piano Fingering (PIG) Dataset [9], which includes 150 scores and 309 music fingering data. We selected a 2-layer LSTM network and a BI-LSTM network. Their hidden layer sizes are both 32. The Adam optimizer was used for training [14]. In order to reflect the comparability of the results, we used the same test set as [9]. To facilitate comparisons of the results, the dataset was divided in accordance with the methods outlined by Nakamura et al. [10]. We used a miscellaneous subset comprising 120 pieces by 24 composers as the training set, and the remaining 30 scores were used as the test set.

We also used a proprietary dataset to evaluate the generalizability of learned fingering transfer knowledge. This data set included 28 pieces by Bach and 7 pieces from the Chinese Conservatory of Music Social Art Level Examination Level 1–3. There was no overlap between the two datasets.

The proposed data augmentation method presented in the previous section was used to generate 50 data points with sequence lengths between 150 and 300. Unless otherwise stated, all results marked with "our model" were used the model in Fig. 1 and training data include the augmented set and miscellaneous subset.

4.2 Evaluation measures

4.2.1 Match rate

Theoretically, the effectiveness of an algorithm depends on the likelihood of coincidence between the labeling result and label. The formula used to calculate the match rate is as follows:

$$\alpha = 1 - \left| L_h \wedge L_a + R_h \wedge R_a \right| / n \tag{12}$$

where n indicates the total length of the music score, L_h and R_h respectively indicate the manual annotation of the left hand and right hand, L_a and R_a respectively indicate the algorithm annotation of the left hand and right hand, and symbol $\wedge$ is an exclusive OR operation. The effectiveness of the proposed method was evaluated using the method in [9]. In PIG dataset, a given score may be associated with more than one fingering; therefore, we obtain the average match rate between the model result and each actual fingering, which is expressed as the general match rate M_{gen}. We can also focus on the ground truth closest to the estimation and define the highest match rate M_{high}. The score data size is N, and after adding different fingering tags, the training data is extended to N_{gen}. The general match rate and the highest match rate are calculated as follows:

$$M_{\text{gen}} = \frac{\sum_{i,j} \alpha_{i,j}}{N_{\text{gen}}} \tag{13}$$

$$M_{\text{high}} = \frac{\sum_i \max_j \alpha_{i,j}}{N} \tag{14}$$

where $\alpha_{i,j}$ indicates the match rate of a prediction to the jth fingering ground truth in the ith score.

4.2.2 Irrational fingering rate

The percentage of fingerings that cannot be performed is referred to as the irrational fingering rate (IFR), which is calculated as follows:

$$\text{IFR} = \left(\sum_i \frac{\sum_{t=2}^{n_i} \psi\left(d^{(t)}, f^{(t)}, f^{(t-1)}\right)}{n_i} \right) \Big/ N \tag{15}$$

$$\psi\left(d^{(t)}, f^{(t)}, f^{(t-1)}\right) = \begin{cases} \frac{1}{2}\left(sgn\left[f^{(t-1)} \cdot f^{(t)} - 5.5\right] + 1\right) & \begin{array}{l} -6 < d^{(t)} < 6 \quad \text{and} \\ \text{left}: d^{(t)} \cdot \left(f^{(t)} - f^{(t-1)}\right) > 0 \\ \text{right}: d^{(t)} \cdot \left(f^{(t)} - f^{(t-1)}\right) < 0 \end{array} \\ 1 & \begin{array}{l} \text{left} - \text{hand} \quad \text{chord}: \\ \qquad f^{(t)} \leq f^{(t+1)} \\ \text{right} - \text{hand} \quad \text{chord}: \\ \qquad f^{(t)} \geq f^{(t+1)} \end{array} \\ 0 & \text{other} \end{cases} \tag{16}$$

where ψ is derived from Table 1 use in calculating erroneous transfer fingerings.

4.3 Results and discussion

4.3.1 Evaluation on Pitch difference modeling strategy

We compared the pitch difference modeling strategy with the note modeling strategy, the results of which are presented in Fig. 7. In note modeling, right-hand data from the PIG dataset [9] included 555 types of monophony or polyphony, whereas the left-hand data included 564 types. In pitch difference modeling, the right-hand data included 108 types of monophony or polyphony, and the left-hand data included 101 types. In the case where the matching network is LSTM, the M_{gen} of the PdF (B in Fig. 7) modeling strategy was 12.76% higher than that of PF (A), M_{high} was 12.98% higher, and IFR was 4.03% lower. In the case where the matching network is BI-LSTM, compared with PF(C) the M_{gen} of the PdF(D) modeling strategy is increased by 11.33%, M_{high} is increased by 12.46%, and the IFR is reduced by 2.54%. Pitch difference information is more directly related to fingering than is pitch information, as is the case when manually determining fingering.

4.3.2 Contribution of Bidirectional network

Figure 7 presents the contribution of the bidirectional network to M_{gen} and M_{high} (A and C or B and D). When

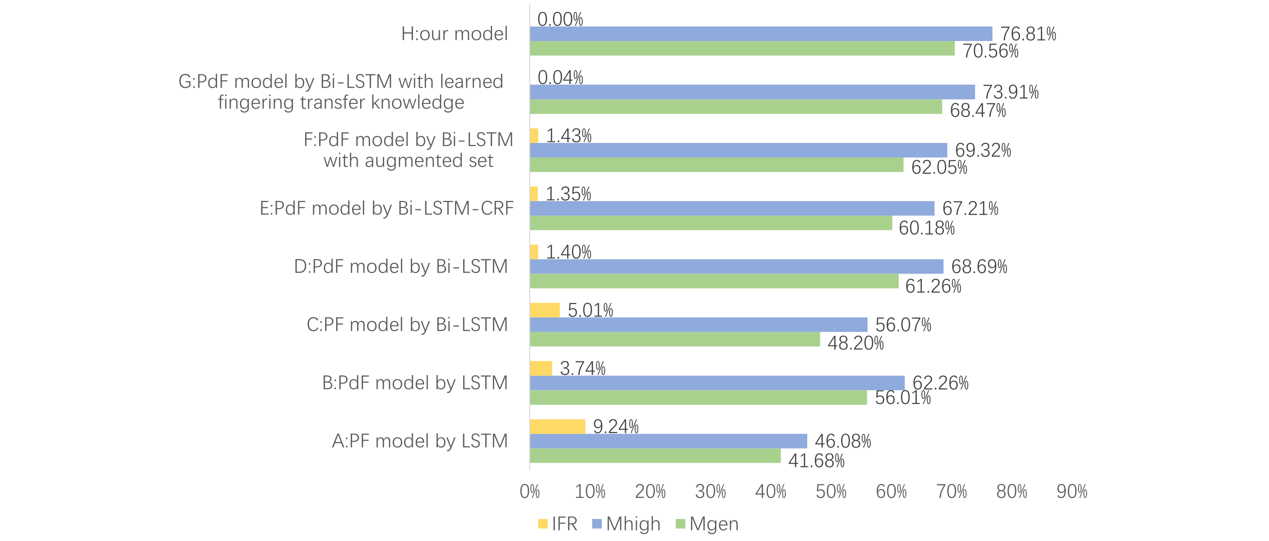

Fig. 7 Results of the ablation study

humans determine fingerings, they must consider the current note as well as the notes well as those that precede and follow it. The IFR values demonstrate that by considering the context, it is possible to mediate the generation of unplayable fingerings, thereby allowing the model to better understand the correspondence between notes and fingering.

4.3.3 *Contribution of adding learned fingering transfer knowledge*

The contribution of finger relationship knowledge was assessed by comparing the performance of BI-LSTM with learned transfer constraint and BI-LSTM-CRF (see Fig. 7G, E). In BI-LSTM-CRF, placing the CRF at the backend of BI-LSTM proved effective in part-of-speech tagging [15]. CRF fits the logical relationship between tags in order to establish the connection between BI-LSTM network outputs. However, in fingering estimation task, it does not make sense to consider the relationship between fingering labels while ignoring specific notes (i.e., the type of input). As shown in Table 1, the logical relationship of fingering is related to whether the pitch is ascending or descending. Experiments demonstrated that the fingering transfer layer is more effective than adding a CRF.

Furthermore, the generalizability of the learned fingering transfer layer was evaluated by estimating the fingerings in the proprietary dataset using the PIG as a training set. The match rate of the BI-LSTM network was 51.06%, whereas the match rate of BI-LSTM with the fingering transfer knowledge layer was 52.26%. These results demonstrated that transfer knowledge learned from PIG data is applicable to other datasets.

4.3.4 *Contribution of data augmentation*

Deep learning-based models depend heavily on the quantity and quality of training data. Extracting key factors from training data while retaining as much information as possible can help to reveal relationships between the original notes and fingerings. One effective strategy to alleviate model overfitting is to augment the available data.

Augmentation respectively increased M_{gen} and M_{high} by 0.79% and 0.63% as shown in Fig. 7D, F. These results confirmed our assertion that increasing the amount of available data would make it easier for the network to establish relationship between fingering and pitch.

4.3.5 *Contribution of priori knowledge*

As shown in experiment G in Fig. 7, the inclusion of fingering transfer knowledge enhanced the practical value of the results; however, we still encountered a number of fingerings that would require additional hand shifts or render the piece unplayable. As shown in experiment H in Fig. 7, the addition of priori knowledge reduced IFR to 0. As long as there is enough data for training, the model can learn the correct fingering transfer knowledge by itself, such that the priori knowledge and learned transfer knowledge can be combined. Limited to data size at present, ensuring that the annotation results are playable requires the addition of constraints.

4.3.6 *Results of ablation study*

The PdF strategy, the inclusion of learned and priori finger transfer knowledge, and dataset augmentation all contributed to improving the performance of the

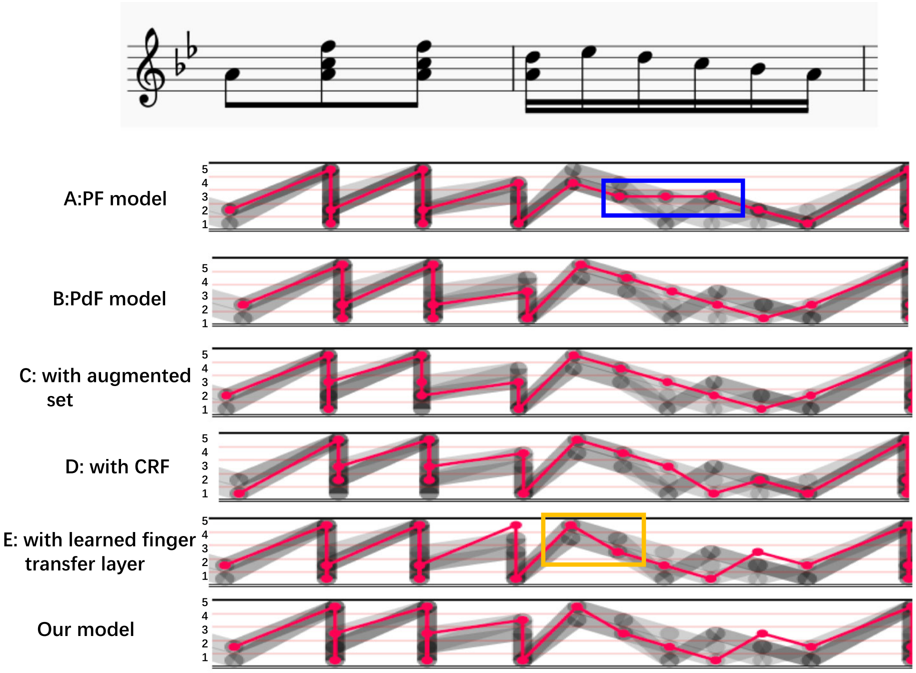

Fig. 8 Comparison of annotation results under a descending scale of the right-hand (English Suite No. 3 – Prelude)

fingering estimation system. The modeling strategy made the largest contribution, increasing accuracy by more than 10% (see Fig. 7C, D). Accordingly, the learning of finger transfer regulations improved model performance by roughly 5% (see Fig. 7D, G). Dataset augmentation also improves model performance (see Fig. 7D, F).

4.4 Example result and error analysis

We discuss some example results to demonstrate the capabilities and limitations of our method. Figures 8, 9, and 10 respectively demonstrate the estimation performance of models with different refining factors in terms of descending scales, chords, and repeated notes. The

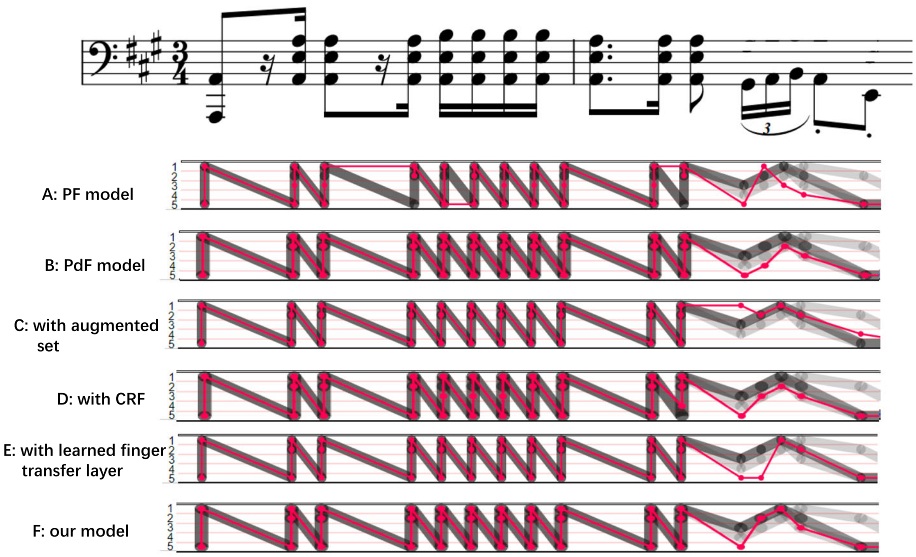

Fig. 9 Comparison of annotation results from a chord of a left-hand fragment in Polonaise Op. 40 No. 1

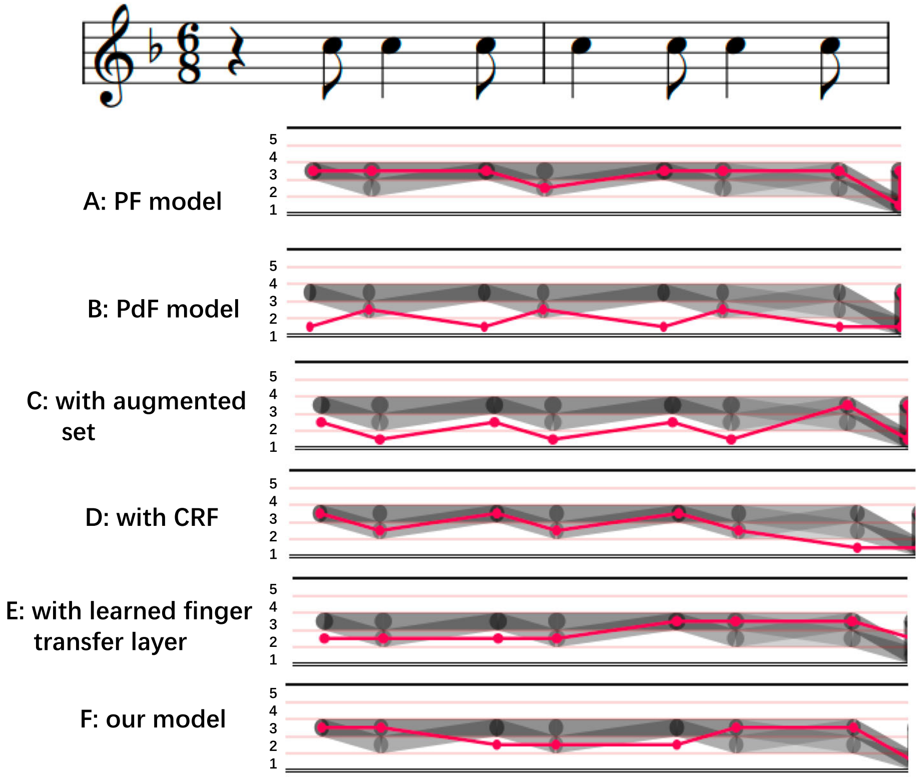

Fig. 10 Annotation result comparison for repeated notes

red circles indicate estimated fingerings and the half-transparent gray circles indicate the ground truth fingerings by multiple players. Note that the results in the examples are based on BI-LSTM.

Figure 8 presents the annotation results for a descending scale. The continuous use of finger 3 (see the blue box in the results of the PF model), could affect the fluency of performance. Here, the learned fingering transfer knowledge prioritized finger 5 to finger 3 instead of the 5 to 4, as seen in the orange box, which also affected the final result. The final estimation results differed somewhat from the ground truth; however, this fingering may still appear in the actual performance.

Figure 9 presents the example results of chord annotation. The PF model continually caused chord estimation errors due to a lack of time information; however, the chord estimates obtained using the PdF strategy were reasonable. Unnecessary finger-crossing appeared among the single notes behind the chords after adding the augmented dataset, due perhaps to the learning of unnecessary features. The inclusion of a fingering transfer layer produced unreasonable finger translations. The proposed model mediated the erroneous transfer of fingers and unplayable chords; however, it failed to eliminate unnecessary finger-crossings or finger repeats.

Figure 10 illustrates a performance of the right-hand score excerpted from Ballade No. 2, in which one tone appears repeatedly. In cases where many fingering combinations are available for repeated notes, performers can choose fingerings based entirely on their playing habits and convenience; thus, there is a wider choice of fingerings than another excerpt. Theoretically, the PdF model discards the actual pitch and represents the continuously repeated tones as 0. This unified representation allows the network to efficiently and quickly obtain the fingering features of repeated notes and at the same time eliminates the interference of different pitches on continuous fingering. Nonetheless, our model did not perform well in learning fingering annotation for repeated pitches, and the estimates obtained using CRF were the closest to the ground truth. The results show that our model does not fully learn the fingering features of this type of segment, for example, it should reduce or avoid reusing the same finger. Despite the differences between the results of our model and the ground truth, the results are generally playable.

4.5 Comparison of existing models in terms of accuracy

Comparisons were conducted between the proposed model and the state-of-the-art 3rd HMM, LSTM, and FF [9] in Fig. 11; the results of which are presented in

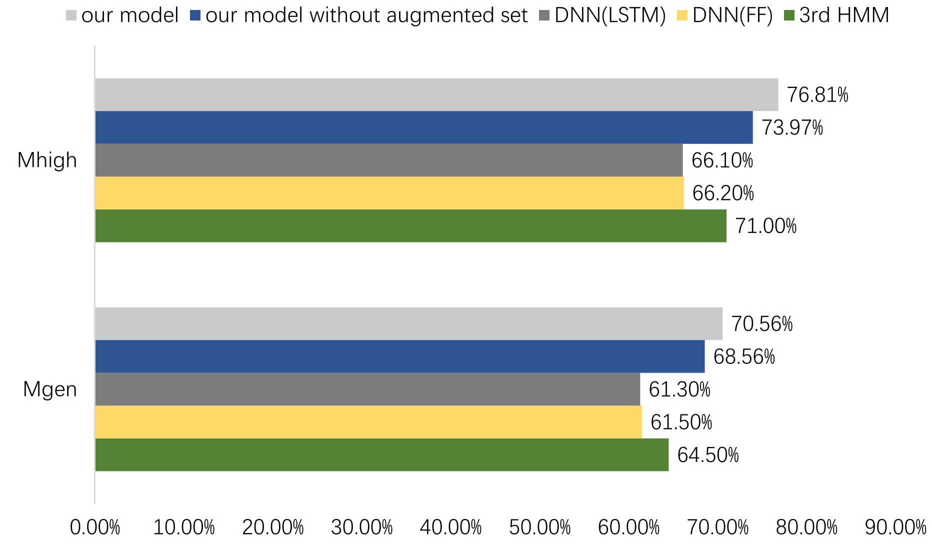

Fig. 11 Comparison of labeling results from the various algorithms

Fig. 11. Overall, the proposed model outperformed the other methods in terms of M_{gen} and M_{high}. Compared with the current best-performing 3rd HMM, our method improves M_{gen} by 6.06% and M_{high} by 5.81%. Without the augmented set, the match rates were increased by 4.06% and 2.97%, respectively.

5 Conclusion

This paper presents a piano fingering estimation method emphasizing pitch differences in conjunction with finger transfer knowledge. In the recognition process, priori knowledge strengthens constraints on finger transfers. In this way, it is possible to maximize the likelihood that the estimated fingerings will be playable from a practical perspective. We also generated an augmented dataset based on the distribution of the training data with the aim of alleviating the problem of overfitting due to a small training set.

Experiment results show that pitch difference with time and interval information is more conducive to fingering estimation than is note-related information. The BI-LSTM network with fingering transfer constraints provided fingering results that were very close to those obtained manually. The use of fingering constraints also helped to ensure the playability of the resulting fingerings. In addition, data augmentation helped the network to capture the note and finger relationship more effectively.

The proposed method outperformed the third-order hidden Markov method in terms of matching rate as well as playability. Nonetheless, our method also has a number of limitations as follows:

(a) In the future, long-range finger transfer information will have to be included to reflect the complex relationships among fingers.

(b) Our model does not consider pauses or hand cooperation, such that decision function T may eliminate some reasonable fingering paths in these situations with the result that match rate is sacrificed for playable fingerings.

(c) The polymorphic aspect of fingering leads to somewhat arbitrary finger sequences, regardless of whether the sequences are derived by a human of computational algorithm. The proposed method is unable to provide a variety of fingering choices for the same score.

(d) Despite data augmentation, the size of our dataset was still insufficient to reveal the relationship between notes and finger sequences.

These problems will have to be addressed in future research.

Abbreviations

PdF: Pitch difference sequence and fingering; IFR: Irrational fingering rate; PF: Pitch sequence and fingering; BI-LSTM: Bidirectional long short-term memory; SFG: Sliced fingering generation; HMM: Hidden Markov model; DNN: Deep neural network; FF: Feedforward; LSTM: Long short-term memory; PIG: Piano fingering; CRF: Conditional random field

Acknowledgements

This work was supported partly by the Natural Science Foundation of Tianjin Municipal Science and Technology Commission (16JCZDJC31100) and National Natural Science Foundation of China (61471263).

Authors' contributions

X G and H Z worked on the experiments and data analysis. Q L implemented the algorithms. H Z wrote the manuscript text and prepared the figures. X G edited the manuscript. All authors reviewed the manuscript. The authors read and approved the final manuscript.

Funding

Natural Science Foundation of Tianjin Municipal Science and Technology Commission (16JCZDJC31100).
Natural science foundation of China (61471263).

Availability of data and materials

The dataset is the open source for academic research on piano fingering, PIG[9], and can be found under the following link: http://beam.kisarazu.ac.jp/research/PianoFingeringDataset/.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 20 July 2021 Accepted: 31 January 2022
Published online: 11 April 2022

References

1. C. C. Lin, D.S.M. Liu, *An intelligent virtual piano tutor, Proceedings of the 2006 ACM International Conference on virtual reality continuum and its applications*. (ACM, Hong Kong, pp. 353–356
2. M. Miura, I. Hirota, N. Hama, et al., Constructing a system for finger-position determination and tablature generation for playing melodies on guitars. Syst. Comput. Jpn. **35**(6), 10–19 (2010)
3. D. R. Tuohy, W. D. Potter, in *Paper presented at the 31st International Computer Music Conference*. A genetic algorithm for the automatic generation of playable guitar tablature, (Barcelona, 2005), pp. 5–9
4. R. Parncutt, J. A. Sloboda, E. F. Clarke, M. Raekallio, P. Desain, An ergonomic model of keyboard fingering for melodic fragments. Music. Percept. **14**(4), 341–382 (1997). https://doi.org/10.2307/40285730
5. J. P. Jacobs, Refinements to the ergonomic model for keyboard fingering of Parncutt, Sloboda, Clarke, Raekallio, and Desain. Music. Percept. **18**(4), 505–511 (2001)
6. E. Nellåker, X. Lu, *Dissertation. Optimal piano fingering for simple melodies*. (School of Computer Science & Communication, 2014)
7. M. Hart, R. Bosch, E. Tsai, Finding optimal piano fingerings. UMAP J. **21**(2), 167–177 (2000)
8. A. Al Kasimi, E. Nichols, C. Raphael, in *Paper presented at the 8th International Conference on Music Information Retrieval, ICMIR 2007*. A simple algorithm for automatic generation of polyphonic piano fingerings, (Vienna, 2007), pp. 23–27
9. E. Nakamura, Y. Saito, K. Yoshii, Statistical learning and estimation of piano fingering. Inf. Sci. **517**, 68–85 (2020)
10. Y. Yonebayashi, H. Kameoka, S. Sagayama, in *Paper presented at the 20th International Joint Conference on Artificial Intelligence*. Automatic decision of piano fingering based on hidden Markov models, (Hyderabad, 2007), pp. 6–12
11. E. Nakamura, N. Ono, S. Sagayama, in *Paper presented at the 15th International Society for Music Information Retrieval Conference, ISMIR 2014*. Merged-output HMM for piano fingering of both hands, (Taipei, 2014), pp. 27–31
12. L. Qiang, L. Chenxi, G. Xin, Automatic fingering annotation for piano score via judgement-HMM and improved viterbi. J Tianjin Univ. (Sci. Technol.) **53**(08), 2020
13. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. D. Kingma, J. Ba, in *Paper presented at the 3rd International Conference on Learning Representations, ICLR 2015*. Adam: a method for stochastic optimization, (San Diego, 2015)
15. Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging. ArXiv (2015). https://doi.org/abs/1508.01991
16. L. R. Rabiner, in *Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286*. A tutorial on hidden Markov models and selected applications in speech recognition, (1989). https://doi.org/10.1109/5.18626

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.