Model	HE	SN	MS	Other
High-Resolution [11]	3.2%	1.5%	1.2%	5.6%
OnsetsFrames [29]	4.2%	2.4%	0.1%	6.7%
Seq2Seq [12]	8.1%	2.9%	0.3%	7.9%

Table 2. Transcription note error rate (aligned with symbolic score) on the Mazurka dataset.

2. **SN**: Segmented notes: One continuous note being transcribed into two segments with a small (<10ms) gap between offset and onset. This error might come from amplitude modulation [32].
3. **MS**: Mis-touched short notes: The spurious, short notes (<16ms) that appear randomly in transcription.

In Table 2, we quantitatively evaluate the presence of these error types on the Mazurka dataset. Given that no performance ground truth exists for this data, we rely on the MusicXML score and the assumption that the performances match the same score version. Among the three most recent deep-learning based transcription models [11, 12, 29], High-Resolution model [11] makes the fewest errors overall, but generates more short notes compared to the other models.

Besides notewise errors, another factor of concern for transcribed MIDI is the note duration. From the inference output [11], the notes' duration are elongated to achieve a sustain effect that is usually implemented by sustain pedal in reality. Whilst such elongation doesn't make an auditory difference in MIDI rendering software, accurate end positions of each note are required in piano performance analysis.

3.2 Joint Note-Pedal Training

With the goal of reconciling the sustain effect from both pedal and keys, as well as achieving more accurate note offsets, we modify the original High-Resolution model [11] with joint note-pedal training. As the first piano transcription model that incorporated the sustain pedal into training, the High-Resolution model trained separate networks for key activity and sustain pedal (with binary velocity), while extending the note offset to match the pedal off timestamp. In joint note-pedal training, 88 keys and 3 pedal channels are combined to output 91 prediction classes with velocity for each channel, and the extension of note offsets is removed. In this case, during training the sustained effect would be conditioned on both key-down duration as well as pedal controls.

As shown in Figure 1, what we model is the key action from the pianist instead of the string damping time of the note (that can either come from the sustain pedal or key action), which deviates from the traditional transcription task. With other training parameters unchanged, the onset F1 score ($tol = 50\,ms$) achieved is 92.1% after 300k iterations, and onsets and offsets evaluation achieved 68.2%. Note that the evaluation results are much lower than the original results, as we are attempting to learn patterns of behaviour that are not present in the audio.

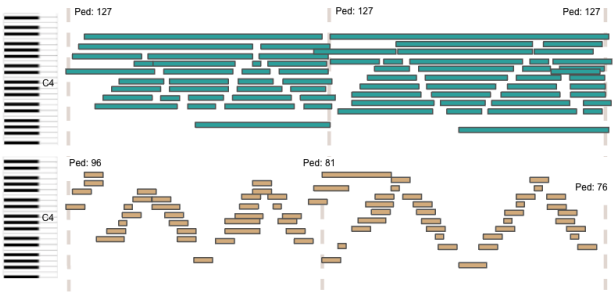

Figure 1. Output pianoroll comparison of the original High-Resolution model (top) and joint note-pedal version (bottom). Dashed lines represent pedal-on messages with velocity.

3.3 Score-Alignment and Correction

As described in Section 3.1, a score-performance alignment algorithm [30] is employed to automatically find transcription errors with reference to a score. As a post-processing step, we correct the differences according to the alignment. Extra notes (those in transcription but not in score) are deleted, mismatching notes (aligned with pitch error) are corrected to the pitch given in the score, and missing notes (those in score but not in transcription) are interpolated and written back to MIDI according to the following rule:

$$g(i) = g(i - \Delta_p) + (g(i + \Delta_n) - g(i - \Delta_p))\frac{\Delta_p}{\Delta_p + \Delta_n}, \quad (1)$$

where $g(i)$ is the onset or offset timestamp of the missing note at beat i, and Δ_p, Δ_n represent the beat distances between the missing note and the previous or next existing notes, respectively.

3.4 Listening Evaluation

In order to evaluate the perceptual quality of transcribed piano music, we perform a subjective listening test where participants rate the similarity of reproduced MIDI and the reference recording. In the test, we compare the ground truth with 4 transcribed MIDI renderings: the original High-Resolution transcription system proposed in [11](**C1**), the joint note-pedal model described in Section 3.2 (**C2**), a score-corrected version of C2 as described in Section 3.3 (**C3**), and the language-model transcription system proposed in [12] (**C4**). All MIDI performances were rendered using a KAWAI CA49 electric piano and recorded using Zoom H4n Pro Recorder. The recordings are then processed with basic noise-reduction in Audacity.

Participants in the listening test are presented with five *20 s* classical piano excerpts with varying style (Q1-Liszt, Q2-Debussy, Q3-Bach, Q4-Rachmaninov, Q5-Mozart; see appendix for music passages). The test is conducted using the MUSHRA protocol [33], each with 5 recordings (reference plus 4 stimuli). Participants are asked which transcribed stimulus sounded closer to the reference on a 100-point scale. During the test, we explicitly ask participants to ignore the timbral or acoustic differences but make

	Q1	Q2	Q3	Q4	Q5	Overall
Reference	4.42±0.24	4.17±0.29	4.24±0.31	4.28±0.31	4.46±0.27	4.30±0.12
C1	**4.12±0.38**	3.52±0.37	3.88±0.32	3.60±0.41	3.88±0.4	3.81±0.16
C2	3.83±0.42	**3.86±0.39**	**4.28±0.31**	**3.97±0.42**	**4.06±0.45**	**4.01±0.17**
C3	3.44±0.37	2.96±0.36	3.44±0.35	3.32 ±0.42	3.76±0.41	3.38±0.17
C4	3.88±0.34	2.32±0.47	3.60±0.29	3.84±0.33	3.68±0.37	3.46±0.18

Table 3. Results of listening test. The mean opinion scores (MOS) and 95% confidence intervals are reported.

judgements based on the expressive differences between the stimuli such as dynamics and timing.

We collected 1075 ratings from 43 listeners. Half of our listeners reported over 5 years of piano performing experience. Table 3 shows the mean opinion scores (converted to a 5-point scale) from the ratings. According to the Wilcoxon signed-rank test ($p < 0.05$), all stimulus groups differ significantly from the reference. Among the stimuli, C2 is preferred by the listeners, while C3 and C4 have significantly lower ratings compared to the other two groups of stimuli. In free text responses, the score-corrected transcription received negative comments such as *unnatural* and *abrupt*. Consequently, we use the C2, note-pedal jointly trained model for transcribing our dataset. The result also shows a perceptual difference of transcription quality across music styles, demonstrating the bias of transcription models: transcribed fast, arpeggio-heavy passages are rated with lower perceptual quality. But for slow, sparse textures, good transcriptions sound much closer to the reference.

4. DATASET OVERVIEW

4.1 Data Collection and Curation

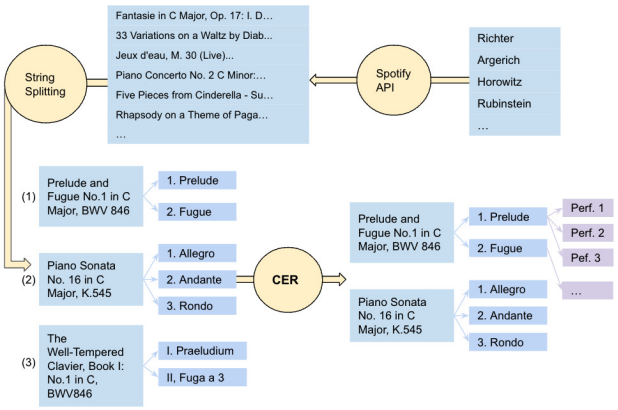

Figure 2. Data curation pipeline.

Our data collection pipeline is presented in Figure 2. Starting with 49 world-renowned pianists, metadata (including composer, performer, album, title and track duration) of their discography was obtained using the Spotify API [3]. After filtering out non-solo keywords such as *concerto* or *trio*, a composition-movement hierarchy was built.

As discussed in Section 2.1, the next challenging step was achieving *Composition Entity Resolution (CER)*, defined as finding out which tracks correspond to the same piece of music, given the variety of naming conventions in classical music. For example, compositions (1) and (3) in Figure 2 actually correspond to the same work, while their title differs so much that a simple string similarity match is not sufficient to resolve them as identical.

We address CER using three steps: **1)** Language-specific mapping. We manually compile a dictionary for interchangeable terms, such as *Prelude* ↔ *Praeludium*. **2)** Unique identifier extraction. Unique information such as key and catalogue number (*Opus*, *BWV*, *K.*, *D.*, etc.) are extracted from the title string. **3)** Fuzzy string matching. For both composition title and movement title, we use normalized Levenshtein distance [34] to compute similarity scores. Note that such string matching is not always reliable, as generic names like *Piano Sonata* are extremely frequent in our discography. Combining the three steps, our composition entity resolution is described in Algorithm 1, where inputs include composition title C and movement title M as well as duration D. Based on the organized metadata, we download each track from a corresponding open source audio at YouTube Music, while the online metadata is again validated by the same CER algorithm.

Algorithm 1 Composition Entity Resolution

```
# UniqueInfo extracts canonical key and composer-specific catalogue number.
for k1, k2 in UniqueInfo(C1, C2) do
    if k1 ≠ k2 then return False
    end if
end for
S_c ← 1 − (Levenshtein(C1, C2))/ max(|C1|, |C2|)
S_m ← 1 − (Levenshtein(C1, C2))/ max(|M1|, |M2|)
S_d ← abs(D1 − D2) / max(D1, D2)
S ← (S_c + S_m)/2 − S_d
return S ≥ 0.6
```

4.2 Audio Matching by Chroma Features

Besides metadata linking using CER, we also match tracks by downloaded audio content to ensure the same piece of

[3] https://developer.spotify.com/documentation/web-api/

music is being performed. Within each group of performances, we apply Chen et al.'s cover song detection algorithm [35] to compare each performance with a reference. We first extract the Harmonic Pitch Class Profile (HPCP) [36] from the reference and the target performance audio. Next, we use the Q_{max} measure, which represents the maximum value of a cumulative matrix computed from the HPCP descriptors of two performances [37]. The similarity between two performances is defined by Eq. 2. We only retain performances whose similarity is larger than 0.9 to build the ATEPP dataset.

$$Sim = 1 - Q_{max}, \quad 0 < Q_{max} < 1 \tag{2}$$

4.3 Applause Filtering with CNN

We also need to filter out any sound that might not be part of a solo piano performance. The most prominent ones are applause and speech from live recordings, which would be transcribed as random pitch. We train a deep learning model based on the Musicnn [38] architecture to filter out any non-solo-piano segment. For each 1 s segment, the probability of non-piano sound is predicted using a binary classifier. In training, a subset of AudioSet [39] with various environmental sounds is used as negative examples, and solo piano recordings are used as positive examples. With a binary tag-gram inferred from the audio, our post-processing step searches for the timestamp of the longest continuous non-solo segment at the beginning and the end to remove from the audio file.

Among the 11742 tracks, 567 of them are detected with starting or ending applause, and were subsequently cleaned. This was followed by a manual verification process by listening to ensure the audio split was accurate.

4.4 MusicXML Score

Given that the musical score is also important for music research, we collect scores in MusicXML format that correspond to our performance data. 228 files are drawn from the ASAP dataset [24], and 90 files from the MuseScore [4] online library, crowd-sourced by the users of MuseScore software. This results in a total of 319 movements, corresponding to 5124 tracks in our dataset (43% of all tracks). The score-performance correspondence is first determined automatically by name matching followed by manual correction.

4.5 Content and Statistics

The ATEPP dataset contains 11742 tracks of 1580 movements. The tracks overlap only 0.2% of the GiantMidi-Piano dataset [23]. Figure 3 shows a breakdown of the movement-performance distribution. Among 1580 movements, 44% have more than 5 performances, providing us with rich data for studying different interpretations of the same piece of music.

In addition, we show a distribution of the top 25 pianists in our dataset in Figure 4, where Sviatoslav Richter contributes the most data in ATEPP. Composer-wise, solo piano works from 25 Western classical composers are included in our dataset, ranging from the Baroque to the Modern era (a full composer breakdown is given in the appendix).

[4] https://musescore.com/sheetmusic

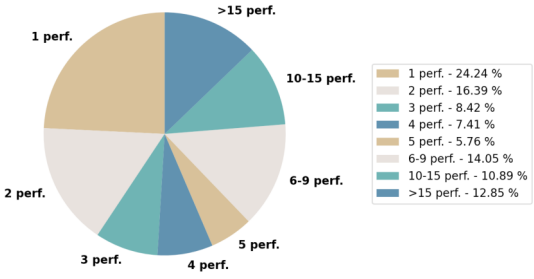

Figure 3. Distribution of movements by number of performances. E.g. 12% of our data have more than 15 performances.

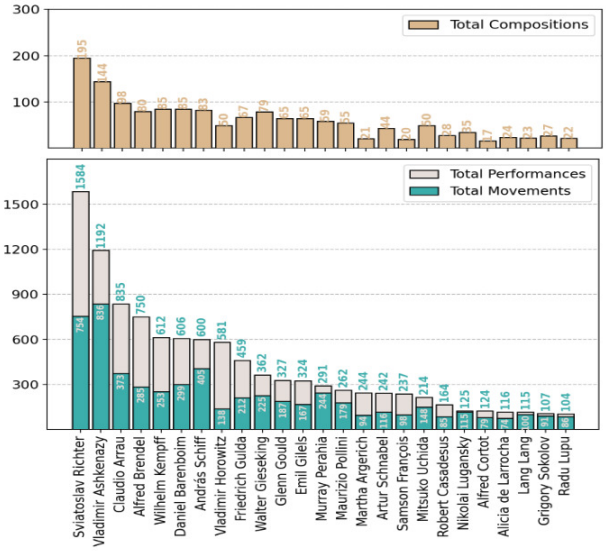

Figure 4. Distribution of the top 25 pianists' performances in the ATEPP dataset.

5. DATASET APPLICATIONS

5.1 Performer Identification

Distinguishing virtuoso performers using computational models has long been studied by researchers who focus on expressive parameters of music performances. Data-driven approaches such as traditional machine learning methods and feature distribution comparison have been applied to this task [8, 9, 40]. However, none of the existing studies have applied deep learning methods to performer identification, due to the lack of large-scale datasets with overlapping performances by different performers.

Using the ATEPP dataset, we are now able to train deep neural networks to identify different performers. We choose four subsets from the ATEPP dataset, only considering performers with over 100 performances. For the

Mixture subset, we only consider compositions that have more than 15 different performances. We also create three composer-specific subsets (L. Beethoven, F. Chopin, and J. S. Bach) to remove the bias of performer-composer correlation. The train-validate-test split is 8:1:1.

Subset	Pianists	Size	Acc.	F1-score
Mixture	16	4676	0.47	0.45
Beethoven	12	3078	0.48	0.46
Chopin	5	973	0.55	0.54
Bach	5	1019	0.59	0.55

Table 4. Performer identification results.

We extracted note-level features including onset time, offset time, velocity, and pitch number from the MIDI files without any expression-related preprocessing. We stacked the sequences of those features and input them into a 1D convolution neural network (see Appendix for the network details). The model was trained on four subsets at a learning rate of 10^{-4} with a decay weight of 10^{-8}.

With all subsets achieving over 0.45 F1 score in Table 4, our baseline model demonstrates the capability of learning individual performing style if given enough data, as well as generalizing across different compositions.

Moreover, the results from composer-specific datasets show that we can achieve comparable results even when performer-style correlation (e.g. Horowitz's repertoire concentrates on Romantic styles while Gould almost exclusively plays Bach) was removed. Confusion matrices are provided in the appendix as well as precision, recall, and F1-score for each performer.

5.2 Performance Rendering with VirtuosoNet

There has been a growing research interest in quantifying and modelling expressive performance using computational models [1], including understanding of how humans perform [41], as well as automatically generating expressive performances [42], and rendering expressive performances from a score [3]. Again, ATEPP contributes to such tasks by providing expressive performance data on a large scale.

In this study, we show the utility of our dataset using the performance rendering model VirtuosoNet [3]. The model consists of an RNN with a hierarchical attention network to model note, beat and measure level hierarchy of music and a conditional variational autoencoder (CVAE) to model the expressive performance. We selected two pianists' Beethoven performances with over 300 tracks from the ATEPP dataset, and altered the model slightly to render a performance in the style of a particular performer by concatenating a performer label vector to the latent vector. During training, we use the same hyper-parameter setting as the the original paper.

The trained model takes a series of note-level score features extracted from MusicXML and a performer name as input, and predicts the corresponding note-level performance features of that performer. Figure 5 shows the real and predicted (pianist1-generated & pianist2-generated) performances of an out-of-sample music, Beethoven's Piano Sonata No. 3 in C Major, Op. 2 No. 3, Mvt. II, by pianist 1 and 2 in terms of dynamics. The flat line represents the default dynamic level of a non-expressive, mechanical rendition of the score.

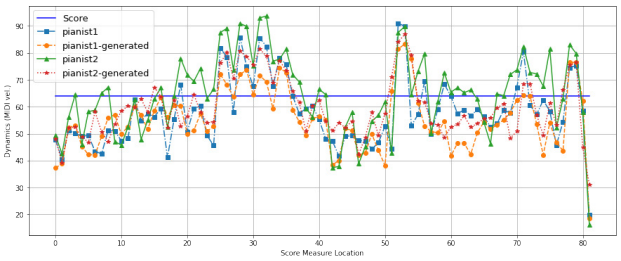

Figure 5. Measure level dynamic variation of Beethoven's Piano sonata No. 3 in C, Op. 2 No. 3, Mvt. II, as performed and generated in the styles of pianists 1 and 2. The flat line depicts the default dynamic level provided in the score file.

We can observe that both actual performances and generated performances (in the style of pianist1 and pianist2) tend to deviate from the default interpretation in a similar way, this can be a demonstration of common performance practice. For individual interpretations, we calculated the cross-correlation of the dynamic variation between the actual performance and the generated version using Pearson's correlation coefficient (r). We found that both actual performances from pianist1 and pianist2 are highly correlated ($r > 0.75$) with their generated counterparts, respectively. In addition, we computed the correlation coefficient between the actual performance of pianist1 and the generated performance of pianist2 and vice-versa. Both of them provide a lower correlation coefficient ($r<0.6$). This demonstrates that deep learning architectures are capable of learning some of the expressive techniques of each individual pianist from respective training data.

6. CONCLUSION AND FUTURE WORK

This paper presents ATEPP, a large-scale dataset of 11,742 expressive piano performances by 49 virtuoso pianists. Nearly half of the compositions are provided with scores in MusicXML format. All of the performances were transcribed by piano transcription model which was trained jointly with pedals and keys. We performed a listening test to evaluate the reliability of the transcription algorithm. To our knowledge, our dataset is the largest dataset of expressive piano performance MIDI with robust metadata of classical music, derived via Composition Entity Resolution (CER). We presented our baseline experiments for performer identification and performer-oriented expressive performance rendering. The results demonstrate that the ATEPP dataset enables us to study expressive features of individual performing styles with deep learning methods.

In the future, we will consider a hierarchical database system that comprehensively links performances with performer, composer, and composition entities through an au-

tomatic CER process. A more balanced dataset is planned as well as an extended version with more variety across the skill level of performers.

7. ACKNOWLEDGMENTS

This work is supported by the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, funded by UK Research and Innovation [grant number EP/S022694/1]. Jingjing is also funded by China Scholarship Council (CSC) [grant number 202008440382].

8. REFERENCES

[1] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational models of expressive music performance: A comprehensive and critical review," *Frontiers in Digital Humanities*, vol. 5, no. October, pp. 1–23, 2018.

[2] A. Lerch, C. Arthur, A. Pati, and S. Gururani, "Music performance analysis: A survey," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[3] D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam, "Virtuosonet: A hierarchical RNN-based system for modeling expressive piano performance," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 908–915.

[4] A. Maezawa, K. Yamamoto, and T. Fujishima, "Rendering music performance with interpretation variations using conditional variational RNN," *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 855–861, 2019.

[5] P. Ramoneda, M. Miron, and X. Serra, "Piano fingering with reinforcement learning," 2021.

[6] H.-W. Dong, C. Zhou, T. Berg-Kirkpatrick, and J. Mcauley, "Deep performer: Score-to-audio music performance synthesis," in *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[7] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, and J. Engel, "MIDI-DDSP: Detailed control of musical performance via hierarchical modeling," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=UseMOjWENv

[8] S. R. M. Rafee, G. Fazekas, and G. A. Wiggins, "Performer identification from symbolic representation of music using statistical models," in *Proceedings of the International Computer Music Conference (ICMC)*, 2021.

[9] E. Stamatatos and G. Widmer, "Automatic identification of music performers with learning ensembles," *Artificial Intelligence*, vol. 165, pp. 37–56, 2005.

[10] A. Tobudic and G. Widmer, "Learning to play like the great pianists," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005, pp. 871–876.

[11] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[12] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[13] M. Bernays and C. Traube, "Expressive production of piano timbre: Touch and playing techniques for timbre control in piano performance," *Proceedings of the Sound and Music Computing Conference*, no. August 2013, pp. 341–346, 2013.

[14] A. Robertson, "Decoding tempo and timing variations in music recordings from beat annotations," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, 2012.

[15] B. H. Repp, "Acoustics, perception, and production of legato articulation on a digital piano," *The Journal of the Acoustical Society of America*, vol. 97, 1995.

[16] C. S. Sapp, "Comparative analysis of multiple performances," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2007.

[17] M. Grachten, W. Goebl, S. Flossmann, and G. Widmer, "Phase-plane representation and visualization of gestural structure in expressive timing," *Journal of New Music Research*, vol. 38, no. 2, pp. 183–195, Jun. 2009.

[18] Z. Shi, C. S. Sapp, K. Arul, J. McBride, and J. O. Smith, "SUPRA: Digitaizing the stanford university piano roll archive," in *Proceeding of the 20th International Society on Music Information Retrieval (ISMIR)*, 2019.

[19] V. Konz, W. Bogler, and V. Arifi-M, "Saarland music data," *Late-Breaking and Demo Session of the International Conference on Music Information Retrieval (ISMIR)*, 2011.

[20] K. Kosta, O. F. Bandtlow, and E. Chew, "MazurkaBL: Score-aligned loudness, beat, and expressive markings data for 2000 chopin mazurka recordings," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, 2018, pp. 85–94.

[21] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling

and generation with the Maestro dataset," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019, pp. 1–12.

[22] M. Hashida, E. Nakamura, and H. Katayose, "Crest-MusePEDB 2nd edition: Music performance database with phrase information," in *Proceedings of the 15th Sound and Music Computing (SMC) Conference*, 2018.

[23] Q. Kong, B. Li, J. Chen, and Y. Wang, "GiantMIDI-Piano : A large-scale midi dataset for classical piano music," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 87–98, 2022.

[24] F. Foscarin, A. Mcleod, P. Rigaux, F. Jacquemard, and M. Sakai, "ASAP : a dataset of aligned scores and performances for piano transcription," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[25] W. Goebl and S. Dixon, "Analysis of tempo classes in performances of mozart sonatas," in *Proceedings of the VII International Symposium on Systematic and Comparative Musicology and III International Conference on Cognitive Musicology*, 2001, pp. 65–76.

[26] A. Tobudic and G. Widmer, "Relational IBL in classical music," *Machine Learning*, vol. 64, no. 1-3, pp. 5–24, 2006.

[27] B. H. Repp, "The dynamics of expressive piano performance: Schumann's "Traumerei" revisited," *The Journal of the Acoustical Society of America*, pp. 641–50, 1996.

[28] G. Widmer, S. Dixon, W. Goebl, E. Pampalk, and A. Tobudic, "In search of the Horowitz factor," *AI Magazine*, vol. 24, no. 3, pp. 111–130, 2003.

[29] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 50–57, 2018.

[30] E. Nakamura, K. Yoshii, and H. Katayose, "Performance error detection and post-processing for fast and accurate symbolic music alignment," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[31] E. Nakamura, N. Ono, Y. Saito, and S. Sagayama, "Merged-output hidden markov model for score following of midi performance with ornaments, desynchronized voices, repeats and skips," in *Proceedings of the 11st Sound and Music Computing Conference 2017, SMC*, 2017.

[32] T. Cheng, S. Dixon, and M. Mauch, "Modelling the decay of piano sounds," in *Proceedings of International Conference on Acoustic, Speech and Signal Processsing (ICASSP)*. IEEE, 2015.

[33] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

[34] W. Heeringa, "Measuring dialect pronunciation differences using levenshtein distance," Ph.D. dissertation, University of Groningen, 2005.

[35] N. Chen, W. Li, and H. Xiao, "Fusing similarity functions for cover song identification," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 2629–2652, 2018.

[36] E. Gomez, "Tonal description of music audio signals," Ph.D. dissertation, University of Pompeu Fabra, 2006.

[37] J. Serrà, X. Serra, and R. G. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics*, vol. 11, 2009.

[38] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 637–644, 2018.

[39] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. Channing Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[40] R. Ramirez, E. Maestre, and X. Serra, "Automatic performer identification in commercial monophonic jazz performances," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1514–1523, 2010.

[41] C. E. Cancino-Chacón, T. Gadermaier, G. Widmer, and M. Grachten, "An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music," *Machine Learning*, vol. 106, no. 6, pp. 887–909, 2017.

[42] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: learning expressive musical performance," in *Neural Computing and Applications*, vol. 32, no. 4, 2018, pp. 955–967.

PDAUGMENT: DATA AUGMENTATION BY PITCH AND DURATION ADJUSTMENTS FOR AUTOMATIC LYRICS TRANSCRIPTION

Chen Zhang[1] **Jiaxing Yu**[1] **LuChin Chang**[1] **Xu Tan**[2] **Jiawei Chen**[3] **Tao Qin**[2] **Kejun Zhang**[1]
[1] Department of Computer Science, Zhejiang University, China
[2] Microsoft Research Asia
[3] South China University of Technology
zc99@zju.edu.cn, yujxzju@gmail.com, changluchin@gmail.com, xuta@microsoft.com,
csjiaweichen@mail.scut.edu.cn, taoqin@microsoft.com, zhangkejun@zju.edu.cn

ABSTRACT

Automatic lyrics transcription (ALT), which can be regarded as automatic speech recognition (ASR) on singing voice, is an interesting and practical topic in academia and industry. ALT has not been well developed mainly due to the dearth of the paired singing voice and lyrics datasets for model training. Considering that there is a large amount of ASR training data, a straightforward method is to leverage ASR data to enhance ALT training. However, the improvement is marginal when training the ALT system directly with ASR data, because of the gap between the singing voice and standard speech data which is rooted in music-specific acoustic characteristics in singing voice. In this paper, we propose PDAugment, a data augmentation method that adjusts pitch and duration of speech at syllable level under the guidance of music scores to help ALT training. Specifically, we adjust the pitch and duration of each syllable in natural speech to those of the corresponding note extracted from music scores, to narrow the gap between natural speech and singing voice. Experiments on DSing30 and Dali corpus show that the ALT system equipped with our PDAugment outperforms previous state-of-the-art systems by 5.9% and 18.1% WERs respectively, demonstrating the effectiveness of PDAugment for ALT.

1. INTRODUCTION

Automatic lyrics transcription (ALT), which recognizes lyrics from singing voice, is useful in many applications, such as lyrics-to-music alignment, query-by-singing, karaoke performance evaluation, keyword spotting, and so on. ALT on singing voice can be regarded as a counterpart of automatic speech recognition (ASR) on natural speech. Although ASR has witnessed rapid progress and brought convenience to people in daily life in recent years [1], there is not an ALT system that has the same level of high accuracy and robustness as the current ASR systems. The main challenge of developing a robust ALT system is the scarcity of available paired singing voice and lyrics datasets that can be used for the ALT model training. Though a straightforward method is to use speech data to enhance the training data of ALT, the performance gain is marginal because there are significant differences between speech and singing voice. For example, the singing voice has some music-specific acoustic characteristics [2,3] – the large variation of syllable duration and highly flexible pitch contours are very common in singing, but rarely seen in speech [4].

Previous works have already made some attempts to artificially generate "song-like" data from speech for model training. Kruspe et al. [5] applied time stretching and pitch shifting to natural speech in a random manner, which enriches the distribution of pitch and duration in "songified" speech data to a certain extent. Nonetheless, the adjustments are random, so there is still a gap between the patterns of "songified" speech data and those of real singing voices. Compared to the work of Kruspe et al. [5], Basak et al. [6] further took advantage of real singing voice data. It transferred natural speech to singing voice domain with the guidance of real opera data. But it only took the pitch contours into account, ignoring duration, another key characteristic. Besides, it directly replaced the pitch contours with those of the real opera data, without considering the alignment of the note and syllable, which may result in the low quality of synthesized audio.

In this paper, we propose PDAugment, a syllable-level data augmentation method by adjusting pitch and duration under the guidance of music scores to generate more consistent training data for ALT training. To narrow the gap between the adjusted speech and singing voice, we adjust the speech at the syllable level to make it more in line with the characteristics of the singing voice. We try to make the speech more closely fit the music score, to achieve the effect of "singing out" the speech content. PDAugment adjusts natural speech data by the following steps: 1) extracts note information from music scores to get the pitch and duration patterns of music; 2) aligns the speech and notes at

syllable level; 3) adjusts the pitch and duration of syllables in natural speech to match those in aligned notes.

Our contributions can be summarized as follows:

- We develop PDAugment, a data augmentation method to enhance the training data for ALT training, by adjusting the pitch and duration of natural speech at syllable level with note information extracted from real music scores.
- We conduct experiments in two singing voice datasets: DSing30 dataset and Dali corpus. The adjusted LibriSpeech corpus is combined with the singing voice corpus for ALT training. The ALT system with PDAugment outperforms the previous state-of-the-art system and Random Aug by 5.9% and 7.8% WERs respectively in DSing30 dataset and 24.9% and 18.1% WERs respectively in Dali corpus. Compared to adding ASR data directly into the training data, our PDAugment has 10.7% and 21.7% WERs reduction in two datasets.
- We analyze the adjusted speech by statistics and visualization and find that PDAugment can significantly compensate the gap between speech and singing voice as well as synthesize relatively high-quality audio.

2. BACKGROUND

2.1 Automatic Lyrics Transcription

The lyrics of a song provide the textual information of singing voice and are important when contributing to the emotional perception of listeners. Automatic lyrics transcription (ALT) aims to recognize lyrics from singing voice, which is of great importance in music information retrieval and analysis. However, ALT is more challenging than ASR – not like speech, in singing voice, the same lyrics with different melodies will have different pitches and durations, which results in the sparsity of training data.

Some works took advantage of the characteristics of music itself: Gupta et al. [7] extended the length of pronounced vowels in output sequences by increasing the probability of a frame with the same phoneme after a certain vowel frame. Kruspe et al. [2] boosted the ALT system by using the newly generated alignment of singing and lyrics. Gupta et al. [8] tried to make use of the background music as extra information to improve recognition accuracy. Ahlback et al. [9] proposed to tag recordings with non-vocal silence and music labels to improve the recognition rate. However, they just designed some hard constraints or added extra information from the music domain, and still did not solve the problem of data scarcity.

Considering the lack of singing voice database, some work aimed at providing a relatively large singing voice dataset: Dabike et al. [10] collected DSing dataset from real-world user information. Demirel et al. [11] built a cascade pipeline with convolutional time-delay neural networks with self-attention based on DSing30 dataset and provided a baseline for ALT task. Some other works leveraged natural speech data for ALT training: they based on pre-trained automatic speech recognition models and then made some adaptations to improve the performance of singing voice: Fujihara et al. [12] built a language model with lyrics but only used the semantic information and ignored the acoustic properties of singing voice. Mesaros et al. [3] used speaker adaptation technique by shifting the GMM components only with global statistics but did not consider the local information.

However, singing voice has some music-specific acoustic characteristics which are not in speech, limiting the performance when training the ALT system directly with natural speech data. Some work tried to synthesize "song-like" data from natural speech to make up for this gap: Kruspe et al. [5] generated "songified" speech data by time stretching, pitch shifting and adding vibrato. However, the degrees of these adjustments are randomly selected within a range, without using the patterns in real music. Basak et al. [6] took use of the F0 contours in real opera data and converted the speech to singing voice through style transfer. Specifically, they decomposed the F0 contours from the real opera data, obtained the spectral envelope and the aperiodic parameter from the natural speech, and then used these parameters to synthesize the singing voice version of the original speech. Nonetheless, in real singing voice, the note and the syllable are often aligned, but they [6] did not perform any alignment of the F0 contours of singing voice with the speech signal. This misalignment may lead to the change of pitch within a consonant phoneme (in normal circumstances, the pitch only changes between two phonemes or in vowels), which further causes distortion of the synthesized audio and limits the performance of ALT system. Besides, they only adjusted the F0 contours, which is not enough to narrow the gap between speech and singing voice.

Some previous works [13, 14] investigated how to convert speech to singing, however, they not only required speech-singing paired data but also a model training process, which made these methods inappropriate for real-time data augmentation.

In this paper, we propose PDAugment, which improves the above adjustment methods by using real music scores and syllable-level alignment to adjust the pitch and duration of natural speech, to solve the problem of insufficient singing voice data.

2.2 Speech vs. Singing Voice

In a sense, singing voice can be considered a special form of speech, but there are still a lot of discrepancies between them [15, 16]. These discrepancies make it inappropriate to transcribe the lyrics from singing voice using a speech recognition model trained on ASR data directly. In order to demonstrate the discrepancies, we randomly select 10K sentences from LibriSpeech [17] for speech corpus and Dali [18] for singing voice dataset, and make some statistics on them. The natural speech and singing voice mainly differ in the following aspects and the analysis results are