POP909: A POP-SONG DATASET FOR MUSIC ARRANGEMENT GENERATION

Ziyu Wang[1]   **Ke Chen**[2]   **Junyan Jiang**[1]   **Yiyi Zhang**[3]
Maoran Xu[4]   **Shuqi Dai**[5]   **Xianbin Gu**[1]   **Gus Xia**[1]

[1] Music X Lab, Computer Science Department, NYU Shanghai
[2] CREL, Music Department, UC San Diego
[3] Center for Data Science, New York University
[4] Department of Statistics, University of Florida
[5] Computer Science Department, Carnegie Mellon University

`{ziyu.wang, jj2731, yz2092, xianbin.gu, gxia}@nyu.edu,`
`knutchen@ucsd.edu, maoranxu@ufl.edu, shuqid@cs.cmu.edu`

ABSTRACT

Music arrangement generation is a subtask of automatic music generation, which involves reconstructing and re-conceptualizing a piece with new compositional techniques. Such a generation process inevitably requires reference from the original melody, chord progression, or other structural information. Despite some promising models for arrangement, they lack more refined data to achieve better evaluations and more practical results. In this paper, we propose POP909, a dataset which contains multiple versions of the piano arrangements of 909 popular songs created by professional musicians. The main body of the dataset contains the vocal melody, the lead instrument melody, and the piano accompaniment for each song in MIDI format, which are aligned to the original audio files. Furthermore, we provide the annotations of tempo, beat, key, and chords, where the tempo curves are hand-labeled and others are done by MIR algorithms. Finally, we conduct several baseline experiments with this dataset using standard deep music generation algorithms.

1. INTRODUCTION

Music arrangement, the process of reconstructing and re-conceptualizing a piece, can refer to various *conditional* music generation tasks, which includes *accompaniment generation* conditioned on a lead sheet (the lead melody with a chord progression) [1–4], transcription and *re-orchestration* conditioned on the original audio [5–7], and *reduction* of a full score so that the piece can be performed by a single (or fewer) instrument(s) [8,9]. As shown in Figure 1, arrangement acts as a bridge, which connects lead sheet, audio and full score. In particular, *piano arrange-*
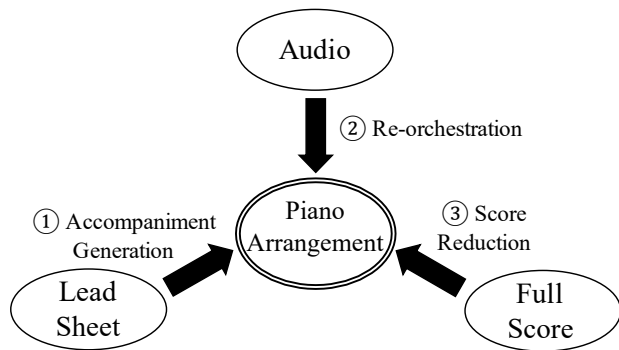

Figure 1: Illustration of the role of piano arrangement in the three forms of music composition, where ① and ② are covered by our POP909 dataset.

ment is one of the most favored form of music arrangement due to its rich musical expression. With the emergence of player pianos [10] and expressive performance techniques [11, 12], we expect the study of piano arrangement to be more meaningful in the future, towards the full automation of piano composition and performance.

In the computer music community, despite several promising generative models for arrangement, the lack of suitable datasets becomes one of the main bottlenecks of this research area (as pointed by [19, 20].) A desired arrangement dataset should have three features. First, the arrangement should be a style-consistent re-orchestration, instead of an arbitrary selection of tracks from the original orchestration. Second, the arrangement should be paired with an original form of music (audio, lead sheet, or full score) with precise time alignment, which serves as a supervision for the learning algorithms. Third, the dataset should provide external labels (e.g., chords, downbeat labels), which are commonly used to improve the controllability of the generation process [21]. Until now, we have not seen such a qualified dataset. Although most existing high-quality datasets (e.g., [13, 15]) contain at least one form of audio, lead melody or full score data, they have less focus on arrangement, lacking accurate alignment and

Dataset	Size	Paired Property				Annotation			Modality
		Polyphony	Lead Melody	Audio	Time-alignment	Beat	Key	Chord	
Lakh MIDI [13]	170k	✓		✓	Δ	✓	Δ	Δ	score, perf
JSB Chorales [14]	350+	✓			N/A	✓	✓		score
Maestro [15]	1k	✓		✓					perf
CrestMuse [16]	100	✓		✓	✓	✓	✓		score, perf
RWC-POP [17]	100	✓	✓	✓	Δ	✓	✓	✓	score, perf
Nottingham [18]	1k		N/A		N/A	✓	✓	✓	score
POP909	1k	✓	✓	✓	✓	✓	✓	✓	score, perf

Table 1: A summary of existing datasets.

labels.

To this end, we propose POP909 dataset.[1] It contains 909 popular songs, each with multiple versions of piano arrangements created by professional musicians. The arrangements are in MIDI format, aligned to the lead melody (also in MIDI format) and the original audios. Furthermore, each song are provided with manually labeled tempo curves and machine-extracted beat, key and chord labels using music information retrieval algorithms. We hope our dataset can help with future research in automated music arrangement, especially task ① and ② indicated in Figure 1:

Task 1: Piano accompaniment generation conditioned on paired melody and auxiliary annotation. This task involves learning the intrinsic relations between melody and accompaniment, including the selection of accompaniment figure, the creation of counterparts and secondary melody, etc.

Task 2: Re-orchestration from audio, i.e., the generation of piano accompaniment based on the audio of a full orchestra.

Besides those main tasks, our dataset can also be used for unconditional symbolic music generation, expressive performance rendering, etc.

2. RELATED WORK

In this section, we begin with a discussion of different modalities of music data in Section 2.1. We then review some existing composition-related datasets in Section 2.2 and summarize the requirements of a qualified arrangement dataset in Section 2.3. Again, our focus is piano arrangement and this dataset is designed for task ① and ② indicated in Figure 1, i.e., *piano accompaniment generation* based on the lead melody or the original audio.

2.1 Modalities of Music Generation

As discussed in [22], music data is intrinsically multi-modal and most generative models focus on one modality. In specific, music generation can refer to: 1) *score generation* [23–26], which deals with the very abstract symbolic representation, 2) *performance rendering* [4, 19, 20], which regards music as a sequence of controls and usually involves timing and dynamics nuances, and 3) *audio synthesis* [27, 28], which considers music as a waveform or

spectrogram. The POP909 dataset is targeted for arrangement generation in the modality of score and performance.

2.2 Existing Datasets

Table 1 summarizes the existing music datasets which are the potential resources for the piano arrangement generation tasks. The first column shows the dataset name, and the other columns show some important properties of each dataset.

Lakh MIDI [13] is one of the most popular datasets in symbolic format, containing 176,581 songs in MIDI format from a broad range of genres. Most songs have multiple tracks, most of which are aligned to the original audio. However, the dataset does not mark the lead melody track or the piano accompaniment track and therefore cannot be directly used for piano arrangement.

Maestro [15] and E-piano [29] contains classical piano performances in time-aligned MIDI and audio formats. However, the boundary between the melody and accompaniment is usually ambiguous for classical compositions. Consequently, the dataset is not suitable for the arrangement task ①. Moreover, the MIDI files are transcription rather than re-harmonization of the audio, which makes it inappropriate for the arrangement task ② either.

Nottingham Database [18] is a high-quality resource of British and Irish folk songs. The database contains MIDI files and ABC notations. One drawback of the dataset is that it only contains monophonic melody without polyphonic texture.

RWC-POP [17], CrestMuse [16], and JSB-Chorale [14] all contain polyphonic music pieces with rich annotations. However, the sizes of these three datasets are relatively small for training most deep generative models.

2.3 Requirements of Datasets for Piano Arrangement

We list the requirements of a music dataset suitable for the study of piano arrangement. The design objective of POP909 is to create a reliable, rich dataset that satisfies the following requirements.

- **A style-consistent piano track**: The piano track can either be an re-orchestration of the original audio or an accompaniment of the lead melody.

- **Lead melody or audio**: the necessary information for the arrangement task ① and ②, respectively.

---

[1] The dataset is available at https://github.com/music-x-lab/POP909-Dataset

- **Sufficient annotations** including key, beat, and chord labels. The annotations not only provide structured information for more controllable music generation, but also offer a flexible conversion between score and expressive performance.

- **Time alignment** among the piano accompaniment tracks, the lead melody or audio, and the annotations.

- **A considerable size**: while traditional machine learning models can be trained on a relatively small dataset, deep learning models usually require a larger sample size (expected 50 hours in total duration).

3. DATASET DESCRIPTION

POP909 consists of piano arrangements of 909 popular songs. The arrangements are time-aligned to the corresponding audios and maintain the original style and texture. Extra annotation includes beat, chord, and key information.

3.1 Data Collection Process

We hire professional musicians to create piano arrangements. In order to maintain a high-quality standard of the arrangements, we divide the musicians into two teams: the *arranger team* and the *reviewer team*. The collection is finalized through an iterative procedure between two teams. For each song, each iteration goes through three steps:

1. Arrangement: the arranger team creates an arrangement from scratch, or revise the previous version of arrangement.

2. Review: the reviewer team decides whether the current version is qualified and comments on how to improve the arrangement in case further revisions are required.

3. Discussion: musicians from both teams catch up with the progress, discuss and improve details of arrangement standards.

We start the process from a list of 1000 popular songs and finally select 909 songs with high arrangement quality. We not only present the last revision (i.e., the qualified version) of each song but also provide the unqualified versions of each song created during the iterative process. This multi-version feature may potentially offer a broader application scenario of the dataset.

3.2 Data Content and Format

In POP909, the total duration of 909 arrangements is about 60 hours. The songs are composed by 462 artists. The release of all songs spans around 60 years (from the earliest in 1950s to the latest around 2010).

Each piano arrangement is stored in MIDI format with three tracks. Figure 2 shows an example of a three-track MIDI file, in which different tracks are labeled with different colors. The three tracks are:

- MELODY: the lead (vocal) melody transcription.

Figure 2: An example of the MIDI file in a piano roll view. Different colors denote different tracks (red for MELODY, yellow for BRIDGE, and green for PIANO).

- BRIDGE: the arrangement of secondary melodies or lead instruments.

- PIANO: the arrangement of main body of the accompaniment, including broken chords, arpeggios, and many other textures.

Here, the combination of BRIDGE and PIANO track forms the piano accompaniment arrangement of the original song. Each MIDI file is aligned with the original audio by manually labeled tempo curve. Moreover, each note event contains expressive dynamics (i.e., detailed velocity control) based on the original audio.

Beat, chord, and key annotations are provided in five separate text files for each song. Annotations for beat and chord have both MIDI and audio versions while key changes annotations are merely extracted from audios.[2] The relevant music information retrieval algorithms are discussed in Section 4.

3.3 Data Folder Structure

Figure 3 demonstrates the folder structure of POP909. In the root directory, there are 909 folders, corresponding to 909 songs. In each folder, we provide the MIDI format arrangement, text format annotations, and a folder of all arrangement versions produced during the iterative processes.

The annotation files contain beat, chord and key annotations in plain text format. Table 2 shows the partial annotations of the song `003` in table format for better illustration purposes. For the beat annotation, `beat_audio` and `beat_midi` are the annotation files extracted from audio and MIDI, respectively. The source of chord and key annotations are indicated in a similar way.

Finally, we provide an index file in the root directory containing the song name, artist name, number of modified times and other useful metadata of the dataset.

---

[2] For annotations from MIDI files, the qualified (final) version of arrangements is used.
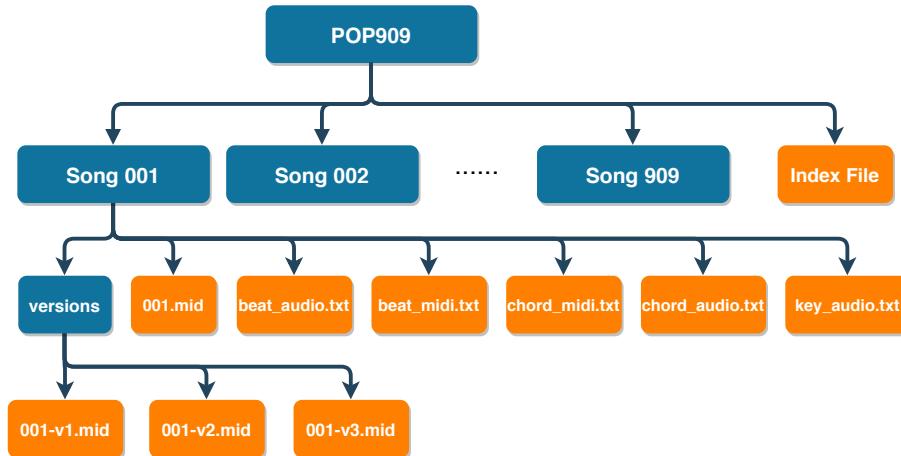
Figure 3: The folder structure of POP909. The blue boxes denote the folder and the orange boxes denote the file.

4. ANNOTATION METHODS

In this section, we discuss how we annotate the beat, chord and key information. For each of the three tasks, different algorithms are applied to extract information from MIDI or audio.

4.1 Beat & Downbeat Estimation

We first extract beat information from MIDI files by taking advantage of two features of the MIDI performance: (1) human-annotated tempo curves, and (2) the accompaniment figure of arrangements which shows a significant sign of beat and downbeat attacks.

Our method can be seen as a modification of the beat-tracking algorithms used in [30, 31]. First, we estimate the initial beat position and use the tempo curve to deduce subsequent beat positions. Second, we estimate the number of beats in a measure by calculating the auto-correlation of the extracted beat features (MIDI onset and velocity), assuming time signature is in general consistent within one song except for some infrequent phase changes. Finally, we search among all the possible phase shifts and find the optimal beat track that has the highest correlation with the extracted features.

We also provide the beat and downbeat annotations extracted from the audio using the algorithm introduced in [32] and compare them with the annotations extracted from MIDI.

For beat position estimation, the two algorithms have more than 90% consistency when the maximum error tolerance is 100 ms, which is acceptable in the data collection process. For downbeat estimation, the two algorithms have 80% agreement. We provide both extraction results in our annotation files.

4.2 Chord Label Extraction

We also provide the chord labels extracted from both MIDI and audio files. For the audio chord recognition, we adopt a large-vocabulary chord transcription algorithm by [33]. As chord changes in popular music are most likely to happen

at beat positions, we post-process the chord boundaries by aligning them to beats to produce the final chord labels.

file	beat time	downbeat_1	downbeat_2
	0.02	1.0	0.0
	0.75	0.0	0.0
	1.49	1.0	1.0
beat_midi	2.22	0.0	0.0
	2.95	1.0	0.0
	3.68	0.0	0.0
	...	...	...

file	beat time	beat order	
	1.46	1.0	
	2.18	2.0	
beat_audio	2.92	3.0	
	3.66	4.0	
	...	...	

file	start time	end time	chord
	0.02	0.75	N
	0.75	1.49	N
chord_midi	1.49	4.41	G:min7
	4.41	7.34	Eb:sus2
	...	...	...

file	start time	end time	chord
	0.00	1.46	N
	2.46	4.39	G:min7
chord_audio	4.39	7.31	Eb:maj(9)
	...	...	...

file	start time	end time	key
key_audio	1.46	226.00	Bb:maj

Table 2: The first several lines of the annotation files for song 003. "downbeat_1" and "downbeat_2" in beat_midi are the two downbeat extractors under simple meter and compound meter assumptions, respectively.

For MIDI chord recognition, we adopt a method similar to the one proposed in [34]. We made two minor changes based on the original algorithm. First, the chord segmentation is performed on the beat level. Second, we alter the chord templates to include more chord qualities

used by pop songs: (1) triads (`maj`, `min`, `dim`, `aug`) with inversions, (2) basic sevenths (`maj7`, `min7`, `7`, `dim7`, `hdim7`) with inversions, (3) suspended chords (`sus2`, `sus4`, `sus4(b7)`), and (4) sixth chords (`maj6`, `min6`).

Note that the arrangement and its original audio may have different chord progressions. For example, a `C:maj` chord may be arranged into `C:sus2`, if necessary. Therefore, both annotations are reasonable and they are not necessarily consistent with each other. To compare the extraction accuracy, we compute the matching rate of the root notes of the chords extracted from both methods. Results show that the matching degree of more than 800 songs in POP909 are above 75%. On the other hand, there are still a few songs whose matching degrees are below 40%. The main reasons are: (1) some of these audio recordings are slightly out of tune, and (2) some parts of the audio have complicated sound effects, in which case our teams decide to re-arrange the chord progression.

4.3 Key Signature Extraction

We also provide key signature annotation from the audio files. We adopt an algorithm very similar to [35]. The original algorithm performs the key classification for a whole song based on the averaged frame-wise feature. In our modified algorithm, we also allow key changes in the middle of the song using a median filter to post-process the frame-level labels.

5. EXPERIMENTS

In this section, we conduct two baseline experiments on music (score-modality) generation with the POP909 dataset: 1) polyphonic music generation (without melody condition), and 2) piano arrangement generation conditioned on melody. For both tasks, we use the Transformer architecture [36] for its advantages in capturing long-term dependencies on time-series data.

5.1 Polyphonic Music Generation

We use a transformer encoder with relative positional encoding [19, 37] to model the distribution of polyphonic music. We adopt a MIDI-like event-based representation slightly modified from [19, 38] to encode the polyphonic music. Each piece of music is represented as a series of events, including note onsets, offsets, velocity changes, and time shifts. We further quantize time shifts tokens under the resolution of $\frac{1}{4}$ beat. In total, we use 16 time-shift events, ranging from $\frac{1}{4}$ beat to 4 beats. Longer notes or rests can be represented by multiple time-shift tokens in a sequence. Table 3 shows the details of our data representation.

We split the dataset into 3 subsets: 90% for training, 5% for validation, and 5% for testing. We set the maximum sequence length $L = 2048$, transformer hidden size $H = 512$, the number of attention heads $h = 6$, and the number of attention layers $N = 6$. Cross Entropy loss is used as the loss function and early stopping is applied.

Event type	Tokenization
Note-On	0-127 (MELODY & BRIDGE track) 256-383 (PIANO track)
Note-Off	128-255 (MELODY & BRIDGE) 384-511 (PIANO track)
Time-Shift	512-527
Velocity	528-560

Table 3: The tokenization of the modified MIDI-like event sequence representation.

GPT-2-based transformer in POP909			
Train Loss	Train Acc.	Test Loss	Test Acc.
2.08978	0.62021	2.38122	0.54529

Table 4: The report of training and test loss and prediction accuracy of MIDI event tokens.

We use Adam optimizer [39] with hyperparameters $\beta_1 = 0.9, \beta_2 = 0.998$. We further adopt the warm-up schedule to control the learning rate. Formally, at the i-th warm-up step, the learning rate

$$lr = \frac{1}{\sqrt{H}} \times \min(\frac{1}{\sqrt{i}}, \frac{i}{S\sqrt{S}}), \tag{1}$$

where $S = 4000$ is a hyperparameter controlling the number of warm-up steps. The training result is presented in Table 4.

5.2 Piano Arrangement Generation

In the second experiment, we design an automatic piano arrangement task: piano accompaniment generation conditioned on the melody. In the data processing step, we first merge the MELODY track and the BRIDGE track into the *main melody* and regard PIANO track the *piano accompaniment*.

We use the same (trained) model in Section 5.1 to model the joint distribution of the main melody and piano accompaniment. During the inference, we force the generated melody to match the given melody condition, generating the most likely accompaniment conditioned on the melody. (A similar conditional generation method has been used in [20].)

5.3 Experiment Results

Figure 4 shows several examples generated by the trained model. In each subfigure, the top piano roll shows the polyphonic music generation (introduced in Section 5.1) result and the bottom piano roll shows the piano arrangement generation (introduced in Section 5.2) result conditioned on the main melody (the blue track). In both cases, the first 500 MIDI-event tokens are given as the context; the red line separates the given context and the generated outputs. We see that the generated pieces capture basic harmonic relationships between the melody and accompaniment and contain consistent rhythmic patterns. Although the quality is still far from the music generated by state-of-the-art
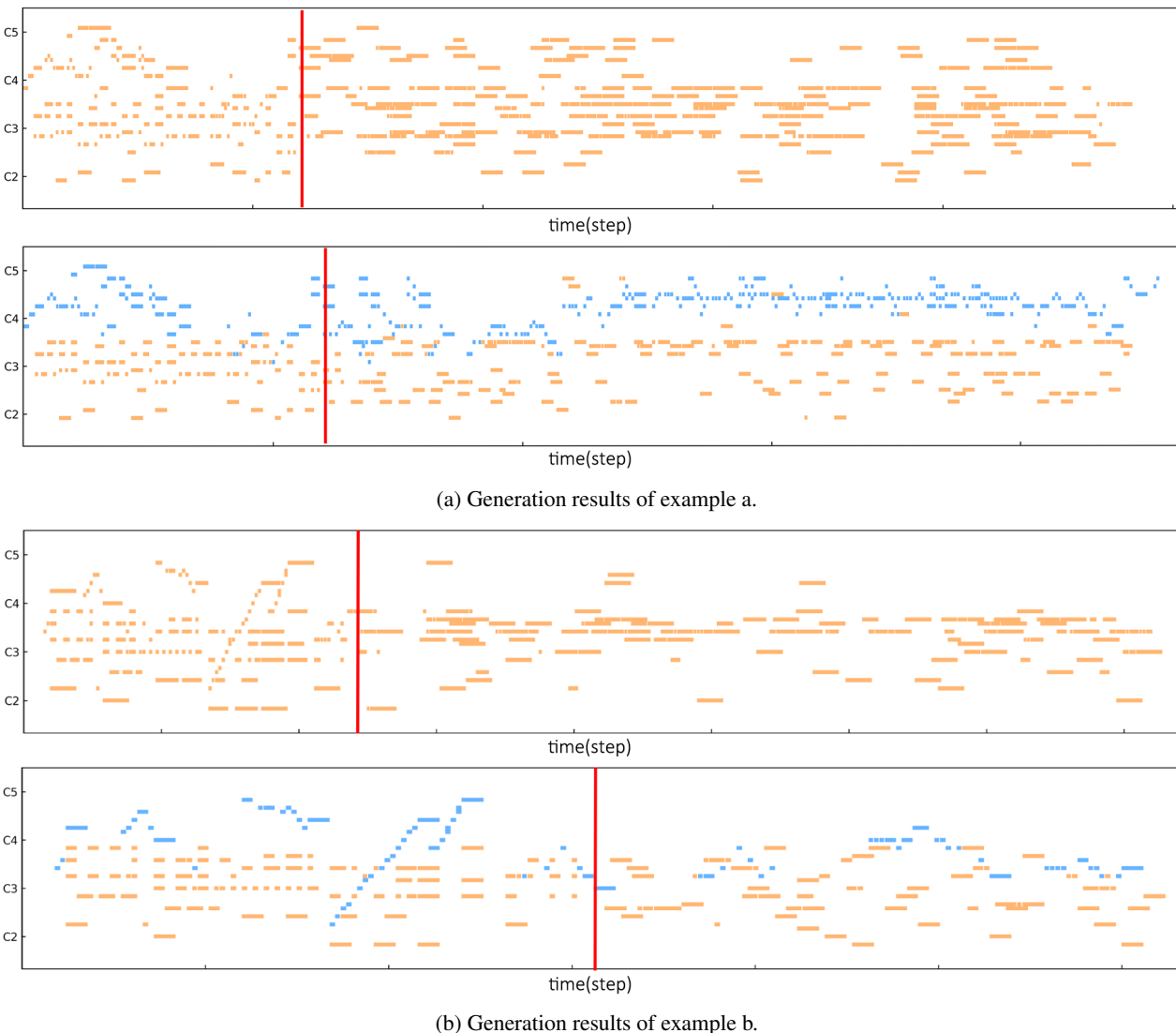
(a) Generation results of example a.

(b) Generation results of example b.

Figure 4: Generation examples with POP909 dataset. Unconditioned polyphonic music generation and piano arrangement generation (blue for the melody, orange for the accompaniment) of the two selected examples are displayed.

algorithms [19, 40], they serve as a baseline to illustrate our dataset usage. We believe that the model can produce better and more structured results with the development of deep generative models.

6. CONCLUSION

In conclusion, we contributed POP909, a tailored dataset for music arrangement. It contains multiple versions of professional piano arrangements in MIDI format of 909 popular songs, together with precise tempo curve aligned to the original audio recordings. We also provide annotations of tempo, beat, downbeat, key, and chord labels. To guarantee a high data quality, the dataset was collected via the collaboration of two groups of professional musicians, arrangers and reviewers, in an interactive process. Apart from the arrangement problem, the POP909 dataset serves as a high-quality resource for structural music generation and cross-modal music generation.

8. REFERENCES

[1] I. Simon, D. Morris, and S. Basu, "Mysong: automatic accompaniment generation for vocal melodies," in *Proceedings of the 2008 Conference on Human Factors in Computing Systems (CHI)*.   Florence, Italy: ACM, 2008, pp. 725–734.

[2] A. Elowsson and A. Friberg, "Algorithmic composition of popular music," in *the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*, 2012, pp. 276–285.

[3] Z. Wang and G. Xia, "A framework for automated pop-song melody generation with piano accompaniment arrangement," *arXiv preprint arXiv:1812.10906*, 2018.

[4] H. Dong, W. Hsiao, L. Yang, and Y. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 34–41.

[5] H. Takamori, T. Nakatsuka, S. Fukayama, M. Goto, and S. Morishima, "Audio-based automatic generation of a piano reduction score by considering the musical structure," in *International Conference on Multimedia Modeling (ICMM)*. Springer, 2019, pp. 169–181.

[6] G. Percival, S. Fukayama, and M. Goto, "Song2quartet: A system for generating string quartet cover songs from polyphonic audio of popular music." in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 114–120.

[7] Y.-N. Hung, I. Chiang, Y.-A. Chen, Y.-H. Yang *et al.*, "Musical composition style transfer via disentangled timbre representations," *arXiv preprint arXiv:1905.13567*, 2019.

[8] E. Nakamura and S. Sagayama, "Automatic piano reduction from ensemble scores based on merged-output hidden markov model," in *Proceedings of the 41st International Computer Music Conference (ICMC)*, 2015.

[9] J.-L. Huang, S.-C. Chiu, and M.-K. Shan, "Towards an automatic music arrangement framework using score reduction," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 8, no. 1, pp. 1–23, 2012.

[10] M. Xu, Z. Wang, and G. Xia, "Transferring piano performance control across environments," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, 2019, pp. 221–225.

[11] G. Xia, "Expressive collaborative music performance via machine learning," Ph.D. dissertation, Carnegie Mellon University.

[12] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Graph neural network for music score data and modeling expressive piano performance," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 3060–3070.

[13] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," *PhD thesis, Columbia University*, 2016.

[14] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*. icml.cc / Omnipress, 2012.

[15] 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 *7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.

[16] M. Hashida, T. Matsui, and H. Katayose, "A new music database describing deviation information of performance expressions," in *Proceedings of 9th International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, PA, USA, 2008, pp. 489–494.

[17] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases," in *Proceedings of 3rd International Conference on Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[18] E. Foxley, "Nottingham database," https://ifdo.ca/~seymour/nottingham/nottingham.html, 2011.

[19] C. A. H. et al., "Music transformer: Generating music with long-term structure," in *7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.

[20] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. J. McAuley, "Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 685–692.

[21] K. Chen, W. Zhang, S. Dubnov, G. Xia, and W. Li, "The effect of explicit structure encoding of deep neural networks for symbolic music generation," in *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*. IEEE, 2019, pp. 77–84.

[22] S. Dai, Z. Zhang, and G. G. Xia, "Music style transfer: A position paper," *Proceeding of International Workshop on Musical Metacreation (MUME)*, 2018.

[23] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation," in *Proceedings of the 34th International Conference on Machine Learning, (ICML)*. Sydney, NSW, Australia: PMLR, 2017, pp. 1362–1371.

[24] A. Pati, A. Lerch, and G. Hadjeres, "Learning to traverse latent spaces for musical score inpainting," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 343–351.

[25] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, "Music transcription modelling and composition using deep learning," in *Proceedings of 1st Conference on Computer Simulation of Musical Creativity (CSMC)*, 2016.

[26] K. Chen, G. Xia, and S. Dubnov, "Continuous melody generation via disentangled short-term representations and structural conditions," in *14th International Conference on Semantic Computing (ICSC)*. San Diego, CA, USA: IEEE, 2020, pp. 128–135.

[27] A. van den Oord et al., "Wavenet: A generative model for raw audio," in *The 9t Speech Synthesis Workshop*. Sunnyvale, CA, USA: ISCA, 2016, p. 125.

[28] P. D. et al., "Jukebox: A generative model for music," *CoRR*, vol. abs/2005.00341, 2020.

[29] "International piano-e-competition," http://www.piano-e-competition.com/.

[30] C. Raffel and D. P. Ellis, "Intuitive analysis, creation and manipulation of midi data with pretty midi," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR), Late Breaking and Demo Papers*, Taipei, Taiwan, 2014, pp. 84–93.

[31] H. Grohganz, M. Clausen, and M. Müller, "Estimating musical time information from performed MIDI files," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 35–40.

[32] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York City, United States, 2016, pp. 255–261.

[33] J. Jiang, K. Chen, W. Li, and G. Xia, "Large-vocabulary chord transcription via chord structure decomposition," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 644–651.

[34] B. Pardo and W. P. Birmingham, "Algorithms for chordal analysis," *Computer Music Journal*, vol. 26, no. 2, pp. 27–49, 2002.

[35] F. Korzeniowski and G. Widmer, "End-to-end musical key estimation using a convolutional neural network," in *25th European Signal Processing Conference (EUSIPCO)*. Kos, Greece: IEEE, 2017, pp. 966–970.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[37] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT (NAACL)(Short Papers)*. New Orleans, Louisiana, USA: Association for Computational Linguistics, 2018, pp. 464–468.

[38] I. Simon and S. Oore, "Performance rnn: Generating music with expressive timing and dynamics," https://magenta.tensorflow.org/performance-rnn, 2017.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceeding of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.

[40] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of 21st International Conference on Music Information Retrieval (ISMIR), virtual conference*, 2020.