MUSE2BACH

Wei-Jaw Lee
National Taiwan University
r12942089

Hsin Ai
National Taiwan University
r12942156

Chih-Pin Tan
National Taiwan University
d12942015

ABSTRACT

We introduce Muse2Bach, an innovative style-transferring system for cover song generation. It accepts pop song audio as input and transforms it into a four-part chorale song in the style of Johann Sebastian Bach. Our model leverages a 2-stages approach, extracting a lead sheet from song audio and generating a Bach-style cover song from this lead sheet, our model not only transforms pop songs into Bach's chorales but also enhances the controllability and explainability of the generation process, relative to existing end-to-end approach. Moreover, we examine the results by conducting a comprehensive data analysis and then provide our interpretations and insights in the discussion.

1. INTRODUCTION

Cover song generation, recreating or rearranging the musical elements from an existing piece, is popular within the music-creative community. Musicians have to craft a cover song with careful consideration of musical components such as melody, chords, rhythm, and performance techniques. A notable approach addressing this challenge is Pop2Piano [1]. They present an end-to-end method which transforms song audio into Mel spectrograms as its input and generates piano cover notes based on the provided input conditions. However, it primarily caters to single-instrument cover song generation, necessitates paired data for training, and suffers from reduced controllability and explainability due to its end-to-end design.

To circumvent the limitations of Pop2Piano, we introduce Muse2Bach, an innovative style-transferring system. Our system accepts pop song audio as input and transforms it into a four-part chorale song in the style of Johann Sebastian Bach. By employing a lead sheet as an intermediary representation, we bridge the gap between pop songs and Bach's stylistic music. By leveraging the 2-stages approach, extracting a lead sheet from song audio and generating a Bach-style cover song from this lead sheet, our model not only transforms pop songs into Bach's chorales but also enhances the controllability and explainability of the generation process.

The core of our model lies the Multitrack Music Transformer (MMT). Recognizing MMT's incapacity to process chord information, we augment its representation accordingly. We train our model on the Bach Doodle Dataset, and assess the results by examining the output samples.

However, despite our efforts, the generated results do not meet our expectation. Thus we conduct a comprehensive data analysis to find out the root causes of the encountered issues. We then provide our interpretations and insights in the discussion.

2. RELATED WORKS

In recent years, symbolic music generation has shown success in both single-track [2–4] and multi-track settings [5,6]. Unlike previous works which generate music by just one step, Compose&Embellish [7] propose a 2-stage generation strategy which generate lead sheet first, a coarse-grained content with structure information, and then fine-grained content such as accompaniment notes. In contrast to prior methodologies that generate music in a single step, Compose&Embellish [7] introduces a two-stage generation strategy. This approach generates the lead sheet first, a coarse-grained content with structural information, followed by fine-grained content, such as accompaniment notes.

Regarding conditional generation, instead of generating music entirely from scratch, certain works aim to address conditional generation, wherein music is generated with human-controlled factors, such as melody, chords, or other musical elements. For instance, Compound Word Transformer [4] demonstrates generating piano performance MIDI notes by conditioning on a given lead sheet. Beyond symbolic-domain generation, there's a burgeoning interest in text-to-music works (audio-domain generation) this year [8–11]. Mustango [10] proposes an approach to control generated content based on musical hints like chord and tempo in the text prompt. Music ControlNet [11] utilizes a different method to generate music with precise musical control.

A special form of conditional generation is style transfer in music. In this context, style transfer refers to altering the contextual texture while preserving representative musical elements, such as chorus melody, chord progression, background percussion, etc. This transformation aims to create a feeling of a change in music style while maintaining the overall musical structure. Groove2Groove [12] proposes an end-to-end method to change song style by feeding its model with different style priors. MuseMorphose [13] changes the song style by controlling note density and rhythmic intensity through disentangling these attributes

with a Variational Auto Encoder. Furthermore, cover song generation is one facet of style transfer. Pop2Piano [1] is an end-to-end system that extracts musical features from song audio and generates piano notes for a cover song.

3. METHODOLOGY

Taking inspiration from Compose&Embellish [7], we introduce a novel two-stage system designed for music style transfer, in order to generate a pop song in the style of Bach. We operate under the assumption that a music piece is predominantly defined by its lead sheet, encompassing elements such as melody and chord progression. Hence, our model initiates by extracting the lead sheet from the input song audio, and generates polymorphic music notes by utilizing a lead-sheet conditioned decoder which is trained on Bach's style music.

Figure 1 illustrates our model architecture and the data flow during both the training and inference phases. During the training phase, we extract the lead sheet from Bach's music and utilize it as the conditioning input for the cover generation decoder. During the inference phase, we employ a lead sheet transcription tool to extract the lead sheet from the pop song. This extracted lead sheet is then fed into the generation decoder to create a cover version in Bach's style.

The following two subsections elaborate on this process for a more detailed explanation.

3.1 Stage 1: lead sheet extraction

We employ two distinct approaches to acquire the lead sheet from both Bach's music and a pop song. In the case of Bach's music, we heuristically select the highest part (soprano) as the melody and utilize Chorder [1], a chord analysis tool, to obtain chord labels. For the pop song, we utilize SheetSage [14], the state-of-the-art tool for lead sheet transcription, to directly extract the entire lead sheet.

3.2 Stage 2: multitarck music generation

In order to generate the result with same format of Bach's 4-parts chorale, the model must have the capability of (i) generation and instruments control and (ii) indicating multi-track information. Thus the model and the representation selection is very important for our task.

Considering the two key requirements mentioned above, we survey transformer based models for symbolic music generation and provide a comparison in Table 1. According to the survey, we choose MMT [6] as the backbone model of work because MMT and its representation satisfy the requirements.

3.2.1 Representation

The representation of MMT is a sequence of 1-dimension array events, noted as $x = (x_1, x_2, x_3, ..., x_n)$. Each x_k is composed by:

[1] https://github.com/joshuachang2311/chorder

Model	Multitrack	Instrument control	Generative modeling
REMI [3]	✗	✗	✓
MMM [15]	✓	✗	✓
CP [4]	✗	✗	✓
MusicBERT [16]	✓	✗	✗
FIGARO [17]	✓	✗	✓
MMT [6]	✓	✓	✓

Table 1. Comparisons of related transformer-based music models

$$(x^{type}, x^{beat}, x^{position}, x^{pitch}, x^{duration}, x^{instrument})$$

The variable x^{type} determines the type of the event. There are 5 event in MMT task: *start-of-song*, *instrument*, *start-of-notes*, *note*, *end-of-song*. Events of *start-of-song*, *start-of-notes*, *end-of-song* only show once in the beginning of a song, and the instrument event is used to assign the instruments used by this song, which is placed between the events of *start-of-song* and *start-of-notes*. In our task, the number of instruments is always four, to align with the format of Bach's chorale which consists of soprano, alto, tenor, and bass. For the events which don't belong to *note*, their x^{beat}, $x^{position}$, x^{pitch}, $x^{duration}$, and $x^{instrument}$ are set to zero, only $x^{instrument}$ of instrument events can be non-zero number.

x^{beat}, $x^{position}$ and $x^{duration}$ of *note* events provide the information of note onset and note offset. x^{beat} represents the coarse grid define by beat and the $x^{position}$ represent the fine grid defined by subbeat of each beat. The $x^{duration}$, in the same scale of $x^{position}$, is used to represent the duration of a note between its onset and offset. The grid resolution we set is 4 because there is no duration of note less than semiquaver in Bach's style. x^{pitch} represents the pitch information of a note, and $x^{instrument}$ represents which instrument the note is played by.

To fit our task, we need to separate the token by bar to implement condition method, so we add a type token, *sep*, to separate them. Meanwhile, we noticed that MMT's representation is lack of chord information to improve the performance of the transformer-based symbolic music generation model. Hence, we extract the chord from the midi data and add the information into the representation. Hence, we augment MMT's representation to account for chord information. The new representation becomes:

$$(x^{type}, x^{beat}, x^{position}, x^{pitch}, x^{duration}, x^{chord}, x^{instrument}),$$

where x^{chord} is the placeholder for chord information.

In our approach, we utilize Chorder to obtain chord labels by combining a root note with a specific chord quality. The root of the chord is determined by its denotation of the pitch class. We also consolidated the chord qualities into 11 distinct qualities for our purposes. Consequently, this framework yields a total of 133 unique chord classes.

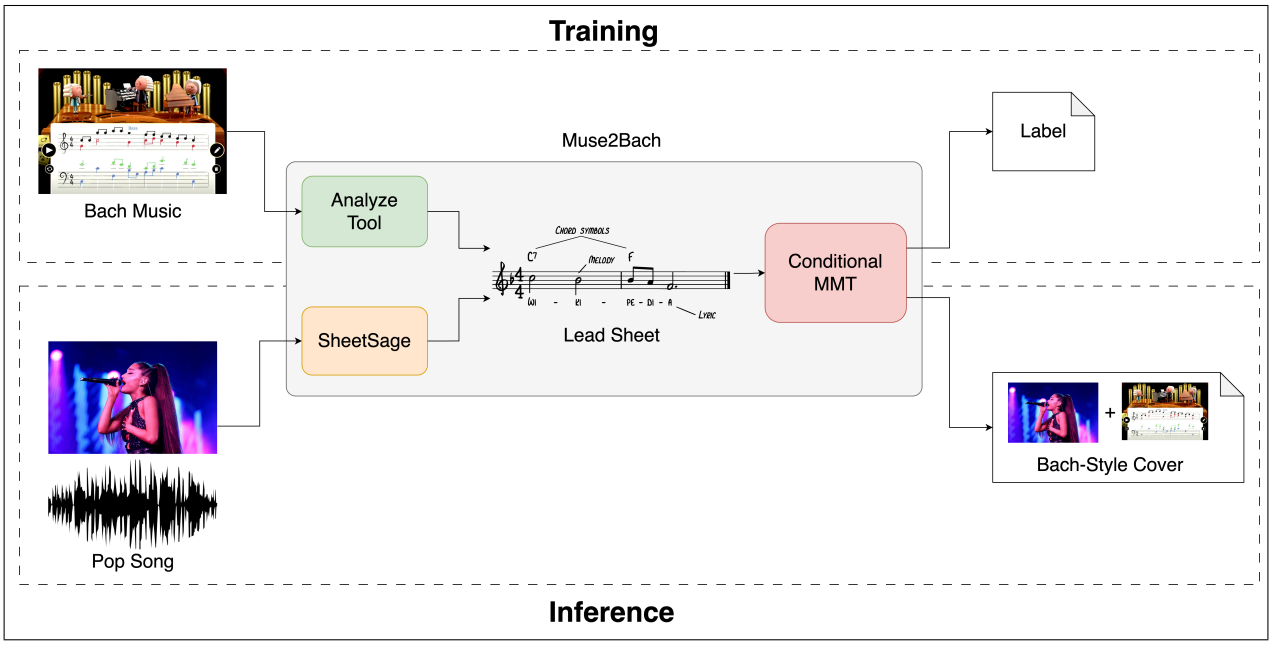


Figure 1. Architecture

This collection comprises 132 specific chords, in addition to a 'None' category, which is used to represent a non-note event. In our system, the variable x^{chord} is non-zero only under two conditions: first, when the x^{type} variable is categorized as a note event, and second, when there is a chord progression coinciding with the note onset.

3.2.2 Model

We implement our model with the same structure as MMT [6] which is an N-layers decoder-only transformer consisting of 7 individual pairs of learnable embedding layers and dense, softmax layers for each variable x, illustrated in Figure 2.

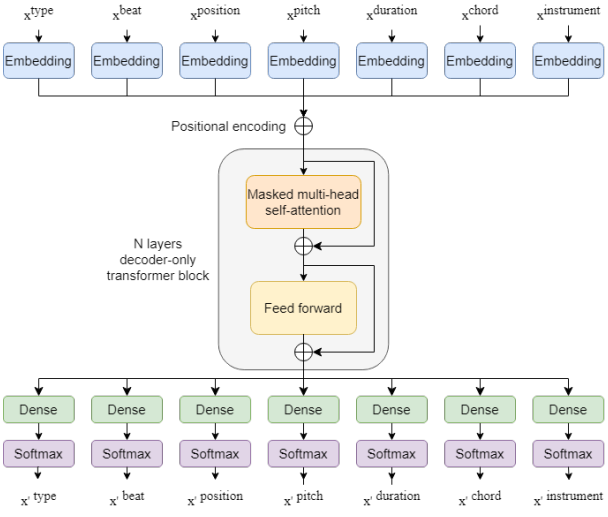


Figure 2. Model structure in our task

In the inference phase, our model requires a soprano melody as an input condition. Based on this, it generates the accompanying parts for the remaining three voices in a chorale. Due to the structure of Bach Doodle Dataset [18], where all pieces are 2 bars in length, we fix the processing window and hop length to one bar. This restriction, however, means that the model is not typically able to generate notes with duration exceeding 8 beats. The generation process advances to the next bar when a 'sep' token is produced and concludes either when the 'end-of-song' token is emitted or when the soprano melody input is fully utilized. It's important to note that the original MMT incorporated certain constraints to ensure that beats are generated in a forward motion, avoiding any regression. However, in our specific task, the target beat range is predefined by the soprano melody condition. Consequently, we have removed this constraint, allowing for more flexibility in the generation process while still adhering to the structural guidelines provided by the soprano melody.

4. DATASET

We use Bach Doodle Dataset and Bach Chorale in our work. The Bach Doodle comprises about 8.5 million sessions which contain a melody provided by users and 3-parts 4-voice harmonization which is composed by Coconet [19]. Each session contains multiple data points with only two bars. We select data with the highest user rating (feedback=2) and randomly choose 100,000 data points for model training. The Bach Chorale comprises 189 4-part chorals (BWV250-438) from Bach's compositions.

5. EXPERIMENTAL SETUP

Our model contains 6 layers of transformer blocks with the 512 embedding length and 8 attention heads. The dropout probability of the dense layer is set to 0.2. We pad the input sequence length to 1024 and the max beat in the model is set to 8 and 256 for Bach Doodle Dataset and

Bach Chorale, respectively.

5.1 Training & Inference

During training, we keep the original augmentation of pitch by shifting all pitches from -5 to 6 with a uniform probability, but discard the random start beat because of the data length misalignment between two datasets. We set the maximum training steps to 200K and validate the data every 1K steps, which is just same as original MMT model. Referring to the T5 [20] conditioning method, we arrange the melody and accompany staggered by bar. We train the model with 4 different setup, composed by the combinations of two dataset and using chord information or not.

For data preprocessing, we apply lead sheet extraction method outlined in section 3 to extract melodies of both Bach and pop music. Finally, we convert melodies into MMT's events with chord labels as the input of the model, generate the remaining three tracks, and then combine the output of the model with melodies to form the full Bach's style music.

6. DISCUSSION

6.1 Lead Sheet Usage

Texture and polymorphic are important in classical music while lead sheet loses lots of information in quantization process. In this study, we utilized the lead sheet as an intermediate representation. However, guiding the model to generate music solely based on the lead sheet imposes limitations on its creative freedom. Our future aim is to treat the lead sheet as a reference, allowing the model more flexibility rather than strictly adhering to it in music generation.

6.2 Data Analysis

We observe that there might be great distinctions between Bach and pop music genres (we adopt *Pop1k7* compiled in [4]) and their potential impact on experimental outcomes. To better understand these differences, we conducted a comprehensive analysis of the harmonic progression and pitch distribution within both genres. Two distinct types of analyses were performed: t-distributed Stochastic Neighbor Embedding (t-SNE) and distribution analysis.

Figure 3 is the t-SNE analysis, we employed one-hot encoding for chord tokens and pitch tokens at each bar, generating separate vectors for pitch and chord information. Subsequently, we applied t-SNE to these vectors, producing t-SNE representations for both pitch and chord data. We observe unclear presentation of differences in our t-SNE result, and suspect that t-SNE analysis is not suitable one-hot encoding is not suitable. One possible solution is to use a Variational Autoencoder (VAE) to directly encode both datasets, generating latent vectors by bar, which may be better to reveal the distinctions between the two datasets.

For the data distribution analysis, we conducted an examination of the pitch and chord distributions in both musical styles. Regarding chord distributions, we specifically focused on the root and quality components. Due to the tonal variations, we also considered the distribution of qualities alone. From Figure 4, 5, 6, we can see the differences in pitch distribution and the frequency of chord occurrences, which highlight clear differences between these two genres. However, the use of the absolute pitch and chord representation method may impact the results of this distribution. Therefore, in our future work, we will take into account tonality and scale to ensure a more precise analysis.

6.3 MMT Representation

We noticed that using the MMT token representation with our chord tokens may present limitations. Since MMT only utilizes the melody track as input, instances where a melody segment lacks notes could result in the model being unaware of chord variations in other tracks. Consequently, the model may not have a comprehensive understanding of the input containing all chords. As a result, we plan to design alternative representations for our data in future work.

6.4 Evaluation Metrics

In this work, the evaluation of generated music quality relies primarily on subjective evaluation through human listening. Without an objective evaluation metric, it is time-consuming and subjective for our evaluation. Consequently, future considerations may involve exploring matrices or quantitative measures that can be employed to objectively evaluate the quality of the Bach's style music we generate.

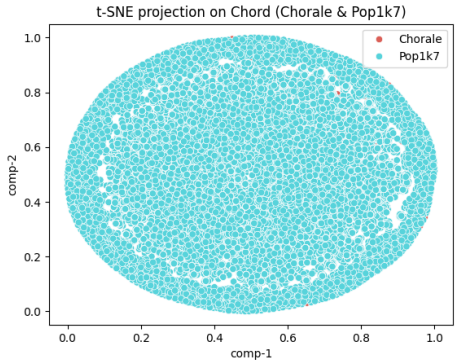
6.5 More Condition

Currently, our model relies on lead sheets and six types of MMT tokens. We aspire to introduce additional conditions. For instance, considering the rhythmic differences between Bach and pop music, we aim to incorporate rhythm as a new condition. This would enable our model to generate music in the style of Bach with specific rhythmic characteristics, offering a more genre-specific approach to music generation.

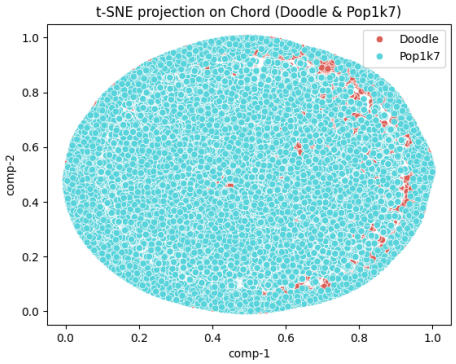
7. REFERENCES

[1] J. Choi and K. Lee, "Pop2piano: Pop audio-based piano cover generation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

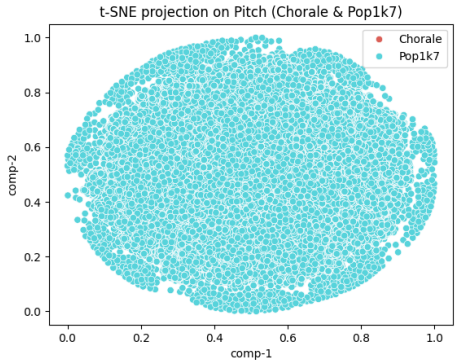
[2] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck, "Music transformer: Generating music with long-term structure," *arXiv preprint arXiv:1809.04281*, 2018.



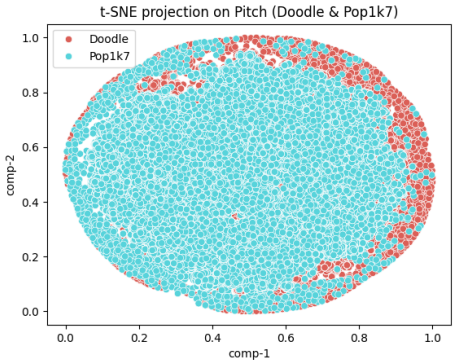
(a) t-SNE on chord (Chorale & Pop1k7)



(b) t-SNE on chord (Doodle & Pop1k7)

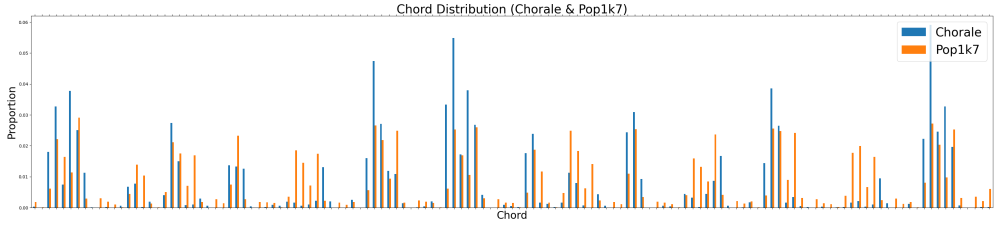


(c) t-SNE on pitch (Chorale & Pop1k7)

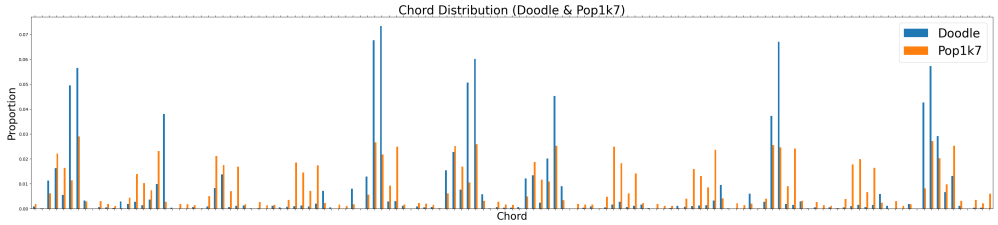


(d) t-SNE on pitch (Doodle & Pop1k7)

Figure 3. t-SNE on chord and pitch

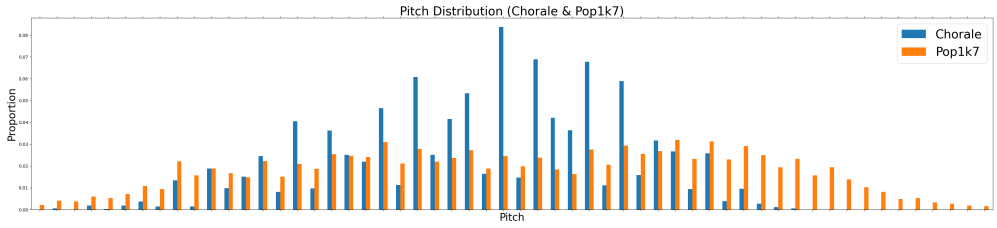


(a) Chord distribution (Chorale & Pop1k7)

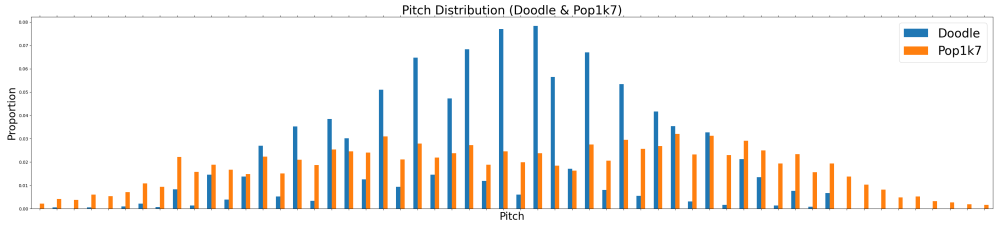


(b) Chord distribution (Doodle & Pop1k7)

Figure 4. Chord distribution

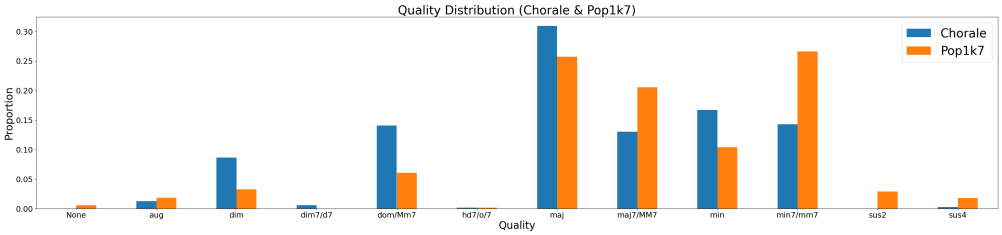


(a) Pitch distribution (Chorale & Pop1k7)

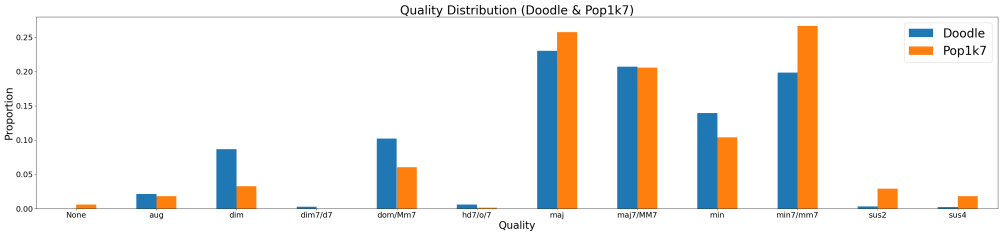


(b) Pitch distribution (Doodle & Pop1k7)

Figure 5. Pitch distribution



(a) Quality distribution (Chorale & Pop1k7)



(b) Quality distribution (Doodle & Pop1k7)

Figure 6. Quality distribution

[3] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1180–1188.

[4] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 178–186.

[5] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," 2017.

[6] H.-W. Dong, K. Chen, S. Dubnov, J. McAuley, and T. Berg-Kirkpatrick, "Multitrack music transformer," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[7] S.-L. Wu and Y.-H. Yang, "Compose & Embellish: Well-structured piano performance generation via a two-stage approach," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023. [Online]. Available: https://arxiv.org/pdf/2209.08212.pdf

[8] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "Musiclm: Generating music from text," 2023.

[9] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *arXiv preprint arXiv:2306.05284*, 2023.

[10] J. Melechovsky, Z. Guo, D. Ghosal, N. Majumder, D. Herremans, and S. Poria, "Mustango: Toward controllable text-to-music generation," 2023.

[11] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," 2023.

[12] O. Cífka, U. Şimşekli, and G. Richard, "Groove2groove: One-shot music style transfer with supervision from synthetic data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2638–2650, 2020.

[13] S.-L. Wu and Y.-H. Yang, "MuseMorphose: Full-song and fine-grained piano music style transfer with one Transformer VAE," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[14] C. Donahue, J. Thickstun, and P. Liang, "Melody transcription via generative pre-training," in *ISMIR*, 2022.

[15] J. Ens and P. Pasquier, "Mmm: Exploring conditional multi-track music generation with the transformer," *arXiv preprint arXiv:2008.06048*, 2020.

[16] Z. Wang and G. Xia, "Musebert: Pre-training music representation for music understanding and controllable generation." in *ISMIR*, 2021, pp. 722–729.

[17] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, "Figaro: Generating symbolic music with fine-grained artistic control," *arXiv preprint arXiv:2201.10936*, 2022.

[18] C.-Z. A. Huang, C. Hawthorne, A. Roberts, M. Dinculescu, J. Wexler, L. Hong, and J. Howcroft, "The Bach Doodle: Approachable music composition with machine learning at scale," in *International Society for Music Information Retrieval (ISMIR)*, 2019. [Online]. Available: https://goo.gl/magenta/bach-doodle-paper

[19] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Counterpoint by convolution," in *International Society for Music Information Retrieval (ISMIR)*, 2017.

[20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

8. APPENDIX

8.1 Personal Contribution

Wei-Jaw takes responsibility of the MMT model training and inference. Ai Hisn takes responsibility of data preprocessing and data analysis. Chih-Pin takes responsibility of building SheetSage.