PICOGEN2: PIANO COVER GENERATION WITH TRANSFER LEARNING APPROACH AND WEAKLY ALIGNED DATA

Chih-Pin Tan[1,2] **Hsin Ai**[1] **Yi-Hsin Chang**[1] **Shuen-Huei Guan**[2] **Yi-Hsuan Yang**[1]
[1] Graduate Institute of Communication Engineering, National Taiwan University, Taiwan
[2] KKCompany Techonologies
tanchihpin0517@gmail.com, yhyangtw@ntu.edu.tw

ABSTRACT

Piano cover generation aims to create a piano cover from a pop song. Existing approaches mainly employ supervised learning and the training demands strongly-aligned and paired song-to-piano data, which is built by remapping piano notes to song audio. This would, however, result in the loss of piano information and accordingly cause inconsistencies between the original and remapped piano versions. To overcome this limitation, we propose a transfer learning approach that pre-trains our model on piano-only data and fine-tunes it on weakly-aligned paired data constructed without note remapping. During pre-training, to guide the model to learn piano composition concepts instead of merely transcribing audio, we use an existing lead sheet transcription model as the encoder to extract high-level features from the piano recordings. The pre-trained model is then fine-tuned on the paired song-piano data to transfer the learned composition knowledge to the pop song domain. Our evaluation shows that this training strategy enables our model, named PiCoGen2, to attain high-quality results, outperforming baselines on both objective and subjective metrics across five pop genres.

1. INTRODUCTION

Piano cover generation, which involves recreating or arranging an existing music piece as a new piano version, is popular within music-creative communities and the music production industry. On media sharing sites like YouTube, piano cover creators often have lots of subscribers. Additionally, many music producers create and distribute piano arrangements on music streaming platforms.

Attempts have been made in the field of music information retrieval (MIR) to automatically generate piano covers from existing musical pieces. Takamori *et al.* [1] proposed a regression method to generate piano reductions, which can be considered simplified versions of piano covers, using acoustic features and structural analysis of the input music. With the recent surge in deep learning, Choi *et al.* [2] introduced a model named Pop2Piano that tackles piano cover generation by leveraging the concept of piano transcription and employing the MT3 architecture [3], originally designed for transcription, as their model backbone. They collected pop songs and the corresponding piano covers from the Internet, and built a song-piano *synchronized* dataset by "remapping" the piano notes to the song audio with a warping algorithm (thereby modifies, or warps, the piano cover). The algorithm entails evaluating the similarity between the pitch contour of the vocal signal extracted from the song audio with the top line of the piano MIDI. They then trained the model with the synchronized data, guiding the model to learn the pitch and onset/offset timing of each note in the generated piano cover.

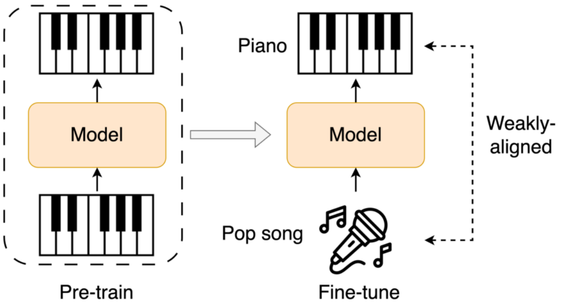


Figure 1. The proposed model is trained with two stages: firstly pre-trained on piano-only data and then fine-tuned on the weakly-aligned song-to-piano pairs.

However, as shown in Table 1, the statistics in the ratio of audio length difference and tempo difference between the original songs and original piano covers (i.e., before note-remapping) they collected [1] show that a piano cover and its original song are not perfectly aligned to each other (for otherwise the difference ratio would be equal to 1.00). This indicates that the tasks cover generation and transcription are inherently different, and that forcing a piano cover to be synchronized with its original song may be inappropriate. Actually, we notice that the note-remapping process of Pop2Piano—i.e., adjusting piano note timing according to the time mapping function obtained by synchronizing piano notes to the song audio—breaks the relation of original piano notes and thereby incurs loss of piano informa-

Attribution: C.-P. Tan, H. Ai, Y.-H. Chang, S.-H. Guan, and Y.-H. Yang, "PiCoGen2: Piano cover generation with transfer learning approach and weakly aligned data", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.,* San Francisco, United States, 2024.

[1] https://github.com/sweetcocoa/pop2piano/blob/main/train_dataset.csv

duration deviation	tempo deviation	IOI deviation
1.10 ± 0.12	1.16 ± 0.25	1.14 ± 0.17

Table 1. The first two statistics contrast the original songs with their original piano covers (i.e., no note-remapping) in the Pop2Piano dataset [2], evaluating the length of the **duration** (in seconds) of the longer one divided by that of the shorter one, and similarly the deviation ratio in **BPM**. The last statistic is similarly the deviation ratio in terms of the average inter-onset intervals (**IOIs**; in seconds), but between the original & adjusted (synchronized) piano covers.

tion. Moreover, from a musical perspective, the way human creates piano covers is by nature different from the way human transcribes music. For cover generation, musicians may firstly analyze the original song in terms of aspects such as melody, chord progression and rhythm section, then decide how to interpret the original song with their composition knowledge, and finally make the piano cover based on the piano performance techniques.

Inspired by the process of human composition for piano cover songs, we propose in this paper a novel approach for piano cover generation by involving the concept of transfer learning [4]. Instead of relying on the *strongly-aligned* pairs [5] that necessitates note-remapping, we use *weakly-aligned* data with the correspondence in "beat" level between song-piano pairs. This approach incurs no rhythmic distortion of the piano covers, retaining their musical quality. Besides, to mitigate the inaccuracy of data alignment, the model is pre-trained on piano-only data to learn the concept of piano performance first, and then fine-tuned on the weakly-aligned paired data to learn the conversion of song to piano, as shown in Figure 1. We also employ a prior model SheetSage [6], pre-trained for lead sheet transcription, as an encoder component that helps our model learn high-level musical concepts for cover generation.

We compare the proposed model, named "PiCoGen2", against other baselines with objective and subjective measures, validating the effectiveness of the weak-alignment method for pairing and the two-step training strategy. We share source code and audio samples at a project page. [2]

2. BACKGROUND

Piano arrangement, i.e., the process of reconstructing and reconceptualizing a piece, is related to various conditional music generation tasks, including lead sheet [3] -conditioned accompaniment generation, transcription and reorchestration, and piano reduction [7–9]. Beyond arrangement, piano cover generation involves creating new musical elements and modifying the original elements via improvisation, tempo changes, stylistic shifts, etc. We briefly review some related topics below.

Symbolic-domain music generation is about generating music in a symbolic form such as pianorolls [10] and discrete MIDI- (Musical Instrument Digital Interface) [11] or REMI-like tokens [12–15], rather than audio signals. The task encompasses unconditional generation (i.e., from-scratch generation) and conditional generation. While the goal of piano cover generation is to generate piano audio given a song audio input, we can treat it as a conditional symbolic music generation task, for we can generate piano in the MIDI domain first, and then use off-the-shelf high-quality piano synthesizers to convert it into audio.

Automatic music transcription (AMT), which aims to precisely transcribe music content from audio signals into a symbolic representation over time, is also related to piano cover generation. AMT tasks can be categorized based on the completeness of information captured from the input audio. One category of AMT tasks aims to capture all music content presenting in the audio, such as automatic piano transcription [3, 16–20]. These methods transcribe the complete polyphonic piano performance from the audio signal. Another category focuses on transcribing a reduced representation of the input, like melody transcription [21, 22] and lead sheet transcription [6, 23, 24]. These tasks extract only the lead melody line and chord progressions, representing a sparse subset of the full musical content. Piano cover generation also requires the exploration of music content reduction and additionally relies on generative modeling conditioned on the reduced representation. For example, Pop2Piano uses MT3 [3] as its backbone to convert audio features into a symbolic piano performance representation. However, following the paradigm of transcription approaches, Pop2Piano requires paired data consisting of pop songs and their corresponding temporally-synchronized piano cover.

Transfer learning is generally consider as the concept of adopting the model to the target domain by re-using parameters that are trained on a source domain, thereby transferring the knowledge between the domains [25]. There have been several works on transfer learning in the field of MIR, e.g., music classification [26–28] and music recommendation [29, 30]. However, to our best knowledge, little attempts have been made to apply transfer learning to the task of cover song generation.

Besides Pop2Piano [2], this work is also closely related to PiCoGen [31], an early version of the current work. We explore the two-stage training strategy for piano cover generation for the first time there. However, in PiCoGen we use discrete symbolic lead sheet as the intermediate representation, instead of continuous conditions supplied by an encoder as done here (see Section 3.2). We note that the sampling process of lead sheet extraction in PiCoGen might loss musical information such as instrumentation and vibes of the input audio. Moreover, we do not explore the idea of transfer learning (Section 3.3) there. [4]

The work of Wang *et al.* [32] is also related, for they deal with the similar problem of converting audio signals

[2] https://tanchihpin0517.github.io/PiCoGen/

[3] A music notation consisting of lead melody and chord progression.

[4] As the previous work [31] was also under review at the time we submitted the current paper, we did not empirically compare PiCoGen and PiCoGen2 in the experiments here. Instead, we provide examples of their generation results for the same input songs on the demo page, which should demonstrate that PiCoGen2 works better than PiCoGen.

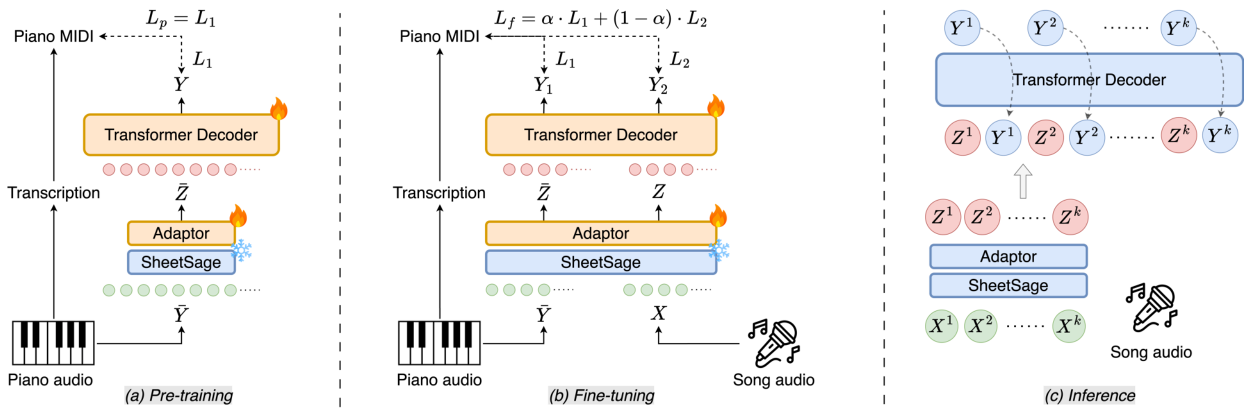


Figure 2. A diagram of the proposed model, PiCoGen2. The fire and snowflake symbols indicate the trainable and frozen parts. For example, the parameters for SheetSage [6], a model pre-trained for lead sheet transcription, are always frozen.

into piano MIDI performances. However, they apply a piano transcription prior and thus using strongly-aligned data as Pop2Piano [2], and they employ a more sophisticated disentanglement-based method to get an intermediate representation. Moreover, they assume that the vocal of the input audio has been removed beforehand, thus actually generating a piano backing track rather than a piano cover.

3. METHODOLOGY

Viewing piano cover generation as a conditional symbolic music generation task, we formulate it as a sequence-to-sequence problem. The objective is to generate a sequence of symbolic tokens Y representing the piano performance, conditioned on the input audio X of the original song.

3.1 Weakly-Aligned Data

In Pop2Piano, Choi *et al.* [2] propose a data preprocessing algorithm to synchronize the piano MIDI to the song audio. They utilize SyncToolBox [33] to analyze the chroma features of two audio segments to obtain a warping path of mapping the time from the piano performance to the song audio. Based on the analysis, they adjust the timing of notes transcribed from the piano performance by using a linear mapping function calculated from the temporal warping information. These remapped notes is then quantized to align with the beat locations of the song audio. However, the rhythmic distortion caused by note-remapping is practically unavoidable, even disregarding the inaccuracy of the synchronization process. The chroma feature only reflects a rough overall alignment between the piano performance and song audio which cannot precisely describe the nuanced amount of timing shift for each individual note. This is evident when examining the changes in the inter-onset intervals (IOIs) between the original piano notes and the remapped version, shown in Table 1.

To avoid the rhythmic distortion of note-remapping, we propose a weak-alignment approach that does not change the timing of piano notes. The idea is to let the alignment rely on only the *beats* of each song-to-piano pair. We construct the time mapping function F_{time} by computing the warping path for the audio pair like the way of Pop2Piano. Given a time of piano performance t_p, the function outputs the corresponding time of song audio $t_s = F_{\text{time}}(t_p)$ according to the temporal warping information. Specifically, we detect the beat locations with Beat Transformer [34] to get the beat times $Q^p = [q_1^p, \ldots, q_{l_p}^p]$ of the piano performance and $Q^s = [q_1^s, \ldots, q_{l_s}^s]$ of the song audio, where l_p and l_s denote the number of beats of each of them. Then we define an aligning function F_{beat} as:

$$F_{\text{beat}}(i) = \arg\min_j (F_{\text{time}}(q_i^p) - q_j^s). \tag{1}$$

For any beat index $i \in [1, ..., l_p]$ of the piano performance, the aligning function outputs the corresponding beat index $j \in [1, ..., l_s]$ of the song audio, and q_j^s is the nearest beat time to $F_{\text{time}}(q_i^p)$. We consider a song-piano pair to be weakly-aligned if the correspondence between them is determined by F_{beat}. See the project page for an illustration.

3.2 Model

An aerial view of our model is depicted in Figure 2. We employ a decoder-only Transformer to accept an input sequence bundling condition X (song audio) and target Y (piano performance) together, and generates the output tokens for Y autoregressively. This approach of providing both the condition and target as a bundled input sequence to the Transformer has been applied in previous studies [13, 14, 35] and has shown success in better informing the model of the temporal correspondence between the condition and desired output. We divide Y into *bars* with the detected beat information and get $Y = [Y^1, \ldots, Y^{B_p}]$, where B_p is the number of bars in the piano cover, and there exists an song audio sequence $X = [X^1, \ldots, X^{B_p}]$ for Y, where each sub-sequence X^k is weakly aligned to Y^k. We then rearrange them with an interleaving form and train the decoder with the bar-wise mix $S = [X^1, Y^1 \ldots, X^{B_p}, Y^{B_p}]$. The decoder model would learn to generate k-th bar of piano performance Y^k depending on (i.e., can attend to) the current and preceding sub-sequences of song audio $[X^1, \ldots, X^k]$ and the preceding sub-sequences of piano performance $[Y^1, \ldots, Y^{k-1}]$.

To reduce the sequence length of X and extract better musical information, we employ a prior audio encoder to transform X into an intermediate representation Z. Different from those works which use Mel-spectrograms [3] or audio codecs [36] for Z, we use SheetSage [6], which is trained for lead sheet transcription, cascaded with an neural adapter as the prior audio encoder. We consider the output embeddings of SheetSage more suitable for representing the input, since they carry information of musical elements connecting a cover with the original song, such as melody, chords and vibes. With the prior encoder, the song audio $[X^1, \ldots, X^{B_p}]$ is transformed into a sequence of latent embeddings $[Z^1, \ldots, Z^{B_p}]$ before being passed to the decoder, yielding the input sequence $[Z^1, Y^1 \ldots, Z^{B_p}, Y^{B_p}]$ of the decoder, as illustrated in Figure 2c.

3.3 Transfer Learning

While the weak-alignment approach eliminates inner temporal distortions for piano performance, there can still be alignment errors between the piano segments and their corresponding song segments. This is because a piano cover is not guaranteed, in the beat level, to have a strict one-to-one mapping with the original song.

To abate such alignment errors, we propose a transfer learning-based training strategy, dividing the training into two steps: pre-training (Figure 2a) and fine-tuning (Figure 2b). In the *pre-training* stage, we train the model with an input sequence $\bar{S} = [\bar{Y}^1, Y^1, \ldots, \bar{Y}^{B_p}, Y^{B_p}]$ where $\bar{Y}$ is the original piano audio recording of the symbolic piano tokens Y. The same as the song audio, the original recording $\bar{Y}$ is encoded to an inter-representation $\bar{Z}$ by the prior encoder. We expect the model to learn to generate piano performances Y with high-level musical features extracted by SheetSage from the piano audio $\bar{Y}$, rather than merely detecting note onsets/offsets like in a piano transcription task. Importantly, there will be no alignment errors between Y and $\bar{Y}$, ensuring that the model can firstly learn the complete concept of piano composition and generation in the pre-training stage, without being impeded by cross-domain alignment issues.

In the *fine-tuning* stage, we train the model with the mixture of $\bar{S}$ and S. Following [36–38], we train the model with the objective of minimizing the cross entropy loss on the tokens of piano performance Y. Let L_1 and L_2 stand for the cross entropy losses for $\bar{S}$ and S, respectively. The loss L_p in the pre-training stage and the loss L_f in the fine-tuning stage can be writeen as:

$$\begin{aligned} L_p &= L_1\,, \\ L_f &= \alpha \cdot L_1 + (1-\alpha) \cdot L_2\,, \end{aligned} \tag{2}$$

where α is the weighting factor determining the proportion of losses contributed from $\bar{S}$ and S during fine-tuning. We expect that α helps the model retain the knowledge about piano performance learned from the pre-training stage.

3.4 Data Representation

For the piano performance sequences Y, we adopt a modified version of the REMI token representation [12], which has been shown to work well for modeling pop piano. Our representation consists of 7 token classes. `Spec` contains special tokens such as `[bos]` (beginning-of-sentence) and `[ss]` (song-start) for controlling the model behavior. `Bar` indicates the property of each bars. `Position`, `Chord` and `Tempo` are metric-related tokens for 16th-note offsets within bars, chord changes (11 roots × 12 qualities), and tempo changes (64 levels). `Pitch`, `Duration` and `Velocity` are note-related tokens for note pitches (A0 to C8), durations (1 to 32 16th-notes), and note velocities (32 levels). There are in total 428 tokens in the vocabulary. In our implementation, `[Bar_start]` and `[Bar_end]` always occur at the start and end of each bar in the input sequence S and $\bar{S}$.

4. EVALUATION

4.1 Dataset

We follow the instructions provided in the Pop2Piano source code to rebuild the training dataset, collecting 5,844 pairs of pop songs and their corresponding piano covers from the Internet. We filter out song pairs with a melody chroma accuracy (MCA) [39] lower than 0.05 or an audio length difference exceeding 15%, leaving 5,503 remaining pairs. In the pre-training stage, all the piano performances from these remaining pairs are used for training. In the fine-tuning stage, we remove invalid bars from the piano performances where the first and last beats of a bar were mapped to the same beat of the original song by the mapping function F_{time}. Around 50% of the bars are removed from the piano performances accordingly. We note that the large number of such invalid bars implies the alignment algorithm of Pop2Piano [2] may not be robust enough and future work can be done to study this.

For objective and subjective evaluations, we collect additional 95 song-to-piano pairs from the Internet, containing 19 Chinese Pop (**Cpop**), 20 Korean Pop (**Kpop**), 16 Japanese Pop (**Jpop**), 20 Anime Song (**Anime**), 20 Western Pop (**Western**) pairs. All the songs contain vocals. We share the URLs of these songs at the project page.

4.2 Experiment Setup

We implement PiCoGen2 using GPT-NeoX [40] as the piano token decoder and SheetSage [6] cascaded with an adapter network as the song audio encoder. The decoder consists of 8 layers, each with 8 attention heads. The adapter is a 4-layer Transformer encoder with 8 attention heads per layer. Our full model has approximately 39M learnable parameters, not counting the SheetSage part for we use it as is with its parameters frozen.

There are 2 ablations compared in the experiment, both of them sharing the same architecture as our full model, but one ablation (Ablation 1) is trained on song-to-piano data *without pre-training*, and the other ablation (Ablation 2) is trained on piano-only data (i.e., *without fine-tuning*). For baselines, besides Pop2Piano, we also include the piano transcription model by Kong *et al.* [20] to validate the effectiveness of the encoder component in our model.

Model	*objective evaluation*			*subjective evaluation* ($\in [1, 5]$)		
	$MCA\uparrow$	$GS\uparrow$	$H_4\downarrow$	OVL ↑	SI ↑	FL ↑
Pop2Piano [2]	**0.42** ± 0.07	0.86 ± 0.09	2.46 ± 0.18	2.71 ± 0.98	2.63 ± 1.01	2.72 ± 1.1
Transcription [20]	0.19 ± 0.06	0.67 ± 0.09	2.78 ± 0.30	1.48 ± 0.74	1.69 ± 0.88	1.45 ± 0.71
Proposed (PiCoGen2)	0.17 ± 0.06	0.84 ± 0.06	2.46 ± 0.22	**3.48** ± 0.93	**3.55** ± 1.06	**3.66** ± 1.02
- Ablation 1 (w/o pre-training)	0.16 ± 0.05	**0.87** ± 0.06	**2.45** ± 0.23	3.09 ± 1.03	2.96 ± 1.02	3.22 ± 1.09
- Ablation 2 (w/o fine-tuning)	0.15 ± 0.05	0.81 ± 0.06	2.57 ± 0.19	3.09 ± 1.02	3.30 ± 1.07	3.08 ± 1.16
Human	0.16 ± 0.06	0.81 ± 0.06	2.59 ± 0.18	4.30 ± 0.87	4.23 ± 0.95	4.33 ± 0.9

Table 2. The results of objective evaluations and the MOS of the subjective study (↑ / ↓: the higher / lower the better).

We train the models with Adam optimizer, learning rate 1e−4, batch size 4 and segment length 1,024. The full model is pre-trained for 100K steps on the piano-only data, and then fine-tuned for an additional 70K steps on the song-to-piano paired data. Ablation 1 is trained from scratch for 100K steps directly on the paired data. Ablation 2 is trained for 50K steps only on the piano-only data, without any exposure to the song-to-piano pairs. During the fine-tuning stage for the full model, we tune the weighting factor α that controls the balance between the piano-only loss and song-to-piano loss, and find that the model achieved the best performance when α is set to 0.25.

For the objective and subjective evaluations, all models are used to generate piano covers of the 95 testing songs (cf. Section 4.1). To eliminate the bias caused by the varying quality of piano recordings, the ground truth human piano performances are first transcribed into MIDI note sequences. These MIDI sequences are then synthesized back into audio using the same FluidSynth-based MIDI synthesizer [41] employed for the model outputs.

4.3 Objective Metrics

We adopt the following existing metrics to assess the quality of the generated piano covers from different aspects, including similarity to the original song and coherence of the piano performance itself.

- **Melody Chroma Accuracy** (MCA) [39] evaluates the similarity between two monophonic melody sequences. The melody line plays a crucial role in deciding whether a song cover resembles the original song. Following Pop2Piano [2], we compute the MCA between the vocals extracted by Spleeter [42] from the test song audio, and the top melodic line extracted from the generated piano cover MIDI using the skyline algorithm [43].
- **Pitch Class histogram Entropy** (H_4) [37] evaluates the harmonic diversity of a musical segment by computing the entropy of the distribution of note pitch class counts. A lower entropy value indicates lower harmonic diversity, but implies a more stable and consistent tonality across the segment. The subscript ("4") indicates the number of bars over which the entropy is calculated.
- **Next-Bar Grooving Pattern Similarity** (GS) is modified from the grooving pattern similarity proposed in [37]. It originally measures the global rhythmic stability across an entire song. Instead of calculating over all pairs in the target, we adapt the metric to focus on local rhythmic stability within a song, evaluating the rhythmic coherence between each bar and its succeeding bar.

4.4 User Study

For subjective evaluation, we conduct an online listening test involving 52 volunteers: 5 professional music producers, 13 amateurs, and 34 pro-amateurs with more than 3-year music training. The volunteers are randomly assigned to distinct test sets, with each set containing 3 songs randomly selected from different genres, and for each song, there are 6 piano performances presented anonymously in random order. These piano performances include: a human piano performance, outputs of our full model and the two ablated versions, and outputs of the Pop2Piano and piano transcription model baselines. All of them are truncated to 40-second audio clips from the beginning. Subjects are asked to listen to these audio clips and provide ratings on a 5-point Likert scale for the following aspects:

- **Similarity (SI):** The degree of similarity between the piano performances and the original song.
- **Music Fluency (FL):** The degree of perceived fluency in the music, representing the smoothness and coherence of the piano performances.
- **Overall (OVL):** How much do the participants like the piano cover in the personal overall listening experience?

4.5 Results

Table 2 displays the results of the objective evaluation metrics and mean opinion scores (MOS) from the user study. In the objective evaluation, Pop2Piano shows a leading MCA score compared to other models and the human piano performances, which indicates it excels at matching the original song's melodic contour. Except for the transcription baseline model, there is no significant difference in GS and H_4 across models, suggesting comparable local rhythmic coherence and harmonic variety.

Next, we pay attention to the result of user study. Much to our delight, the full model leads with the best scores across all aspects in the user study with statistical significance ($p < 0.05$), but there remains a gap compared to the human reference performances. Ablation 1 achieves higher scores than Pop2Piano in all aspects of the user study, both of which are trained on the paired data. This suggests that

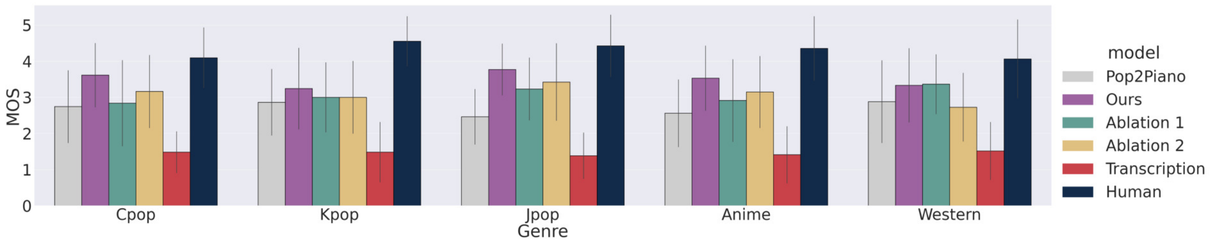


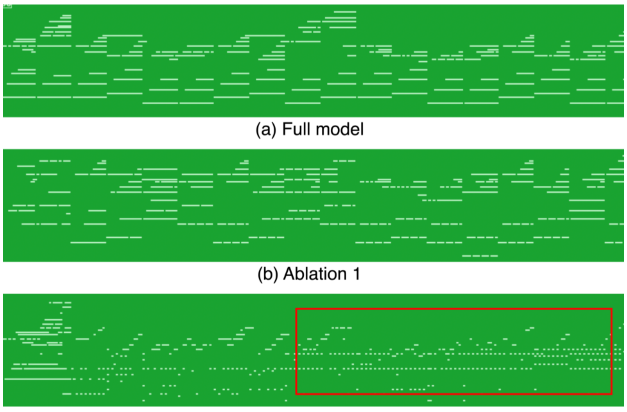
Figure 3. The MOS in overall scores (**OVL**) of the user study in different genres.

utilizing the weakly-aligned paired data, which avoids distorting the original piano performances, helps increase the overall listening experience quality of the model outputs for human raters. Moreover, both Ablation 2 and the transcription baseline are trained on piano-only data, but Ablation 2 performs significantly better than the baseline in both objective and subjective evaluations. This can be seen as evidence that SheetSage, as the encoder, extracts more relevant features beneficial for the piano cover generation task compared to the baseline transcription model.

5. DISCUSSION

In the experiment, we note that while Pop2Piano exhibits a significantly higher MCA score than the other models, even higher than the human performances, it fails to achieve comparably high SI ratings in the user study. We suggest this conflict arises from the assumption in MCA that two melodies must temporally correspond to each other on a fixed "time grid." That is, the corresponding chroma features must be located at precisely the same time instants. For human listening experiences, two similar melodies only need to be coordinated on beats rather than a rigid time grid. Specifically, human perception of melodic similarity allows for the tempo or duration to be slightly changed in the same ratio, as long as their notes are located on the same underlying musical beat positions. As mentioned in Sections 1 & 2, different from transcription or arrangement, a cover song is not usually temporally aligned to the original song, i.e., the musical elements such as tempo, melody, rhythmic changed in the composition process of the piano cover. This temporal flexibility suggests that MCA as an objective measure for the cover generation task may not be adequate and calls for future endeavor to develop better alternatives.

We also find that the two ablated models have the same **OVL** scores in the subjective evaluation, even though Ablation 2 has never seen any pop song data during training. To investigate the reason behind this, we first examine the piano covers generated by Ablation 2. Figure 4 shows a snippet of a cover generated by this model. We note that it tends to generate repeated short notes, resulting in an unnatural-sounding performance. However, Figure 3 demonstrates the **OVL** scores across different music genres. Interestingly, we see that Ablation 2 outperforms Ablation 1 for the *Cpop*, *Jpop*, and *Anime* genres. Additionally, as shown in Table 2, the former ablation also achieves higher **SI** and lower **FL** scores than the latter. From this observation, we suggest that (i) for short audio clips (less than 40 seconds), human raters may place more emphasis on initial melodic accuracy when judging the overall perceived quality, even if Ablation 1 tends to generate more coherent and natural-sounding results; (ii) Ablation 1 does not effectively learn to precisely capture the melodic contour from the reference song condition due to the inherent alignment errors present in the weakly-aligned song-to-piano paired data it was trained on.


(a) Full model

(b) Ablation 1

(c) Ablation 2

Figure 4. The pianoroll representation of a snippet from an example generated by the models. We observe that Ablation 2, which trained on piano-only data, tends to generate repeated short notes.

6. CONCLUSION

In this paper, we have presented PiCoGen2, which applies the concept of transfer learning to the piano cover generation task. We propose a training strategy that involves two stages: pre-training on piano-only data to learn fundamental piano performance skills, followed by fine-tuning on weakly-aligned song-to-piano paired examples for the cross-domain translation. A comprehensive set of experiments validate the effectiveness of the proposed transfer learning approach and the use of weakly-aligned data.

As we still require weakly-aligned data, future work can be done to tackle cover generation without relying on data alignment at all. Moreover, it is useful to have a systematic analysis to evaluate the quality of piano covers and identify the key factors influencing the result, e.g., by studying the performance difference between PiCoGen [31] and PiCoGen2. It is also interesting to generate other covers, such as orchestral covers, and to develop better objective metrics.

7. ACKNOWLEDGMENT

The work is supported by a grant from the National Science and Technology Council of Taiwan (NSTC 112-2222-E-002-005-MY2). We are also grateful to the reviewers and meta-reviewer for helpful comments that help improve the quality of the paper.

8. REFERENCES

[1] 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 *Proc. MultiMedia Modeling Conference (MMM)*, 2019.

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

[3] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "MT3: Multi-task multitrack music transcription," in *Proc. International Conference on Learning Representations (ICLR)*, 2022.

[4] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, 2020.

[5] F. Zalkow and M. Müller, "Using weakly aligned score-audio pairs to train deep chroma models for cross-modal music retrieval." in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2020.

[6] C. Donahue, J. Thickstun, and P. Liang, "Melody transcription via generative pre-training," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2022.

[7] A. Elowsson and A. Friberg, "Algorithmic composition of popular music," in *International Conference on Music Perception and Cognition (ICMPC)*, 2012.

[8] E. Nakamura and S. Sagayama, "Automatic piano reduction from ensemble scores based on merged-output hidden markov model," in *International Conference on Mathematics and Computing (ICMC)*, 2015.

[9] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, G. Bin, and G. Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2020.

[10] 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," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[11] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music Transformer: Generating music with long-term structure," in *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[12] Y.-S. Huang and Y.-H. Yang, "Pop music Transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. ACM Multimedia (ACM MM)*, 2020.

[13] 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 *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[14] J. Huang, K. Chen, and Y.-H. Yang, "Emotion-driven piano music generation via two-stage disentanglement and functional representation," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2024.

[15] D.-V.-T. Le, L. Bigo, M. Keller, and D. Herremans, "Natural language processing methods for symbolic music generation and information retrieval: a survey," *arXiv preprint arXiv:2402.17467*, 2024.

[16] 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," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2018.

[17] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *In Proc. IEEE Signal Processing Magazine*, 2019.

[18] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W.-H. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time Transformer," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2023.

[19] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with Transformers," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2021.

[20] 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 (TASLP)*, vol. 29, pp. 3707–3717, 2021.

[21] R. P. Paiva, T. Mendes, and A. Cardoso, "An auditory model based approach for melody detection in polyphonic musical recordings," in *Proc. Computer Music Modeling and Retrieval (CMMR)*, 2004.

[22] ——, "On the detection of melody notes in polyphonic audio," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2005.

[23] M. P. Ryynänen and A. P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, 2008.

[24] J. Weil, T. Sikora, J.-L. Durrieu, and G. Richard, "Automatic generation of lead sheets from polyphonic music signals." in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2009.

[25] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, 2016.

[26] P. Hamel, M. Davies, K. Yoshii, and M. Goto, "Transfer learning in MIR: Sharing learned latent representations for music audio classification and similarity," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2013.

[27] A. Van Den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2014.

[28] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2017.

[29] K. Choi, G. Fazekas, and M. Sandler, "Towards playlist generation algorithms using RNNs trained on within-track transitions," in *Proc. Workshop on Surprise, Opposition, and Obstruction in Adaptive and Personalized Systems (SOAP)*, 2016.

[30] D. Liang, M. Zhan, and D. P. Ellis, "Content-aware collaborative music recommendation using pre-trained neural networks." in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2015.

[31] C.-P. Tan, S.-H. Guan, and Y.-H. Yang, "PiCoGen: Generate piano covers with a two-stage approach," in *Proc. International Conference on Multimedia Retrieval (ICMR)*, 2024.

[32] Z. Wang, D. Xu, G. Xia, and Y. Shan, "Audio-to-symbolic arrangement via cross-modal music representation learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[33] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync toolbox: A python package for efficient, robust, and accurate music synchronization," *Journal of Open Source Software*, 2021.

[34] J. Zhao, G. Xia, and Y. Wang, "Beat Transformer: Demixed beat and downbeat tracking with dilated self-attention," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2022.

[35] S.-L. Wu and Y.-H. Yang, "Compose & Embellish: Well-structured piano performance generation via a two-stage approach," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[36] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Proc. Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[37] S.-L. Wu and Y.-H. Yang, "The Jazz Transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2020.

[38] ——, "MuseMorphose: Full-song and fine-grained piano music style transfer with one Transformer VAE," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 31, pp. 1953–1967, 2023.

[39] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common MIR metrics," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2014.

[40] A. Andonian, Q. Anthony, S. Biderman, S. Black, P. Gali, L. Gao, E. Hallahan, J. Levy-Kramer, C. Leahy, L. Nestler, K. Parker, M. Pieler, J. Phang, S. Purohit, H. Schoelkopf, D. Stander, T. Songz, C. Tigges, B. Thérien, P. Wang, and S. Weinbach, "GPT-NeoX: Large scale autoregressive language modeling in PyTorch," 2023. [Online]. Available: https://www.github.com/eleutherai/gpt-neox

[41] P. H. Samuel Bianchini, J. Lee, J. Green, P. Lopez-Cabanillas, D. Henningsson, T. Moebert, J.-J. Ceresa, and M. Weseloh, "FluidSynth," 2001. [Online]. Available: https://www.fluidsynth.org/

[42] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, 2020.

[43] A. L. Uitdenbogerd and J. Zobel, "Manipulation of music for melody matching," in *Proc. ACM Multimedia (ACM MM)*, 1998.