manually define a style mapping function to map the original style labels to newly defined ones. [3] The newly defined styles are: "Pop-standard", "Pop-complex", "Dark", and "R&B". While Pop-standard contains only triad chords, Pop-complex contains seventh, ninth, and sometimes even more chromatic ones. Note that some templates do not have an original style label and are thus labeled as "Unknown". Third, we remove all the redundant progressions.

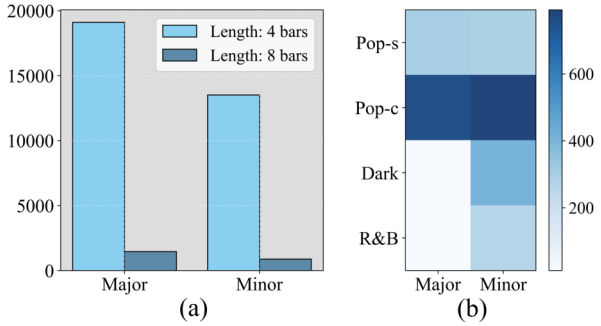


Figure 2: Statistics of the curated dataset.

The final curated dataset contains 5762 pieces of chord progression templates. Each template has 3 additional labels: 1) The length label. Templates are either of 4 bars or 8 bars in length in our dataset. Figure 2 (a) shows a distribution of the length of the templates. 2) The mode label. We currently only label the mode of the templates as either major or minor. 3) The style label. As mentioned above, four styles in total are acquired. The distribution of different styles among modes are shown in Figure 2 (b).

Finally, we represent the reference template space as a collection of tuples:

$$r = \{(r_m^{\text{chord}}, r_m^{\text{label}})\}_{m=1}^{N}, \tag{1}$$

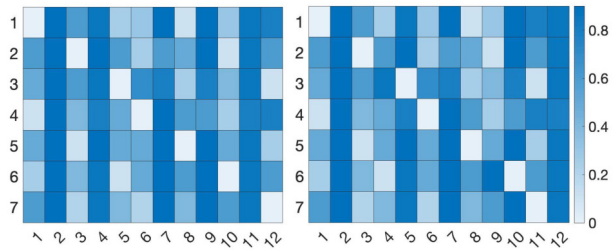
where r_m^{chord} and r_m^{label} are the chord progression and the labels of the i^{th} reference template; N is the volume of the reference space. r_m^{chord} can be seen as a sequence of chords. We quantize and sample the chords at 8th notes from the original chord progression track. Each chord is represented as its root note and a label to indicate whether the corresponding triad is a major triad or a minor triad.

3.2 Harmonization Module

We design a harmonization model that generates structured and coherent full-length chord progression for a given lead melody with phrase annotation. The model takes a multi-phrase melody as input and outputs a list of optimal chord progression identities. Each identity contains a group of progressions that have the same Roman numeral sequence (*e.g.*, I-vi-ii-V) but different styles (*e.g.*, Pop-standard or R&B) and are up for the user to choose. The model considers three levels of losses that involve melody-chord correspondence and is optimized by a dynamic programming (DP) algorithm. Specifically, the DP algorithm optimizes a multi-level loss function consisting of three terms: 1) a micro-level loss for note-wise melody-chord matching, 2) a meso-level loss for phrase-template matching, and 3) a macro-level loss for the full piece coherency.

3.2.1 *Micro-level loss*

The micro-level loss L_{mic} computes the level of dissonance between a melody phrase and candidate progressions note by note. We mainly consider the interval between a melody note and the root of the chord in the corresponding position. The more dissonant the interval is (tritone, minor seconds, etc.), the higher the loss. On the other hand, the more harmonious the interval is (unison, perfect fourth, perfect fifth, etc.), the lower the loss. In addition, the mode of the piece and the harmonic function of the chords will also affect the dissonance level. We refer to the "rank order of consonances and their degree of recurrence" [23] and design the micro-level loss shown in Figure 3, where matrix (a) is for major mode and matrix (b) is for minor. For both matrices, each row represents the degree of the chord (we only consider diatonic chords, *e.g.*, 1 for tonic and 5 for dominant) and each column represents the interval between the melody note and the key center. In Figure 3, the darker the shade, the more dissonant we consider the pair of the melody note and chord is, in which case we set a higher micro-level loss.



(a) Major (b) Minor

Figure 3: Micro-level loss indicating note-wise melody-chord dissonance.

For the i^{th} phrase, L_{mic} is computed by summing up the micro-level loss note by note and then dividing it by the length of the phrase. We further normalize L_{mic} to the range of $[0, 1]$.

3.2.2 *Meso-level loss*

The meso-level loss L_{mes} considers the integrity of a candidate. It encourages chord progressions with the same length as the target melody phrases and penalizes the rest. For a given phrase of melody, we consider chord progressions of the same length and we also concatenate shorter chord progressions to make them candidates. Here, we introduce a transition score to penalize the concatenated candidates. We concatenate two progressions r_m^{chord} and r_n^{chord} from the reference template space, and the transition score considers the very two bars connecting them (*i.e.*, the last bar of r_m^{chord} and the first bar of r_n^{chord}). In specific, we count the occurrence of this two-bar chord progression in the dataset and then normalize it by applying a logarithmic function with N, the size of the reference template space

[3] A detailed description of style mapping is provided in our dataset.

as the base. Suppose the occurrence is $c \in \mathbb{N}$, the transition loss of this $m \to n$ concatenated candidate is defined as:

$$T_{m\to n} := 1 + \log_N \frac{1}{c} \tag{2}$$

Moreover, as we do not wish to penalize the non-concatenated candidates, we introduce δ_1 such that $\delta_1 = 1$ if the candidate is concatenated and $\delta_1 = 0$ otherwise. On the other hand, we do not wish to consider candidates with length not equal to the length of the phrase. Hence we introduce δ_2 such that $\delta_2 = 1$ if the candidate has equal length as the melody phrase, and $\delta_2 = e^{-10}$ otherwise. The meso-level loss of the i^{th} phrase is explicitly:

$$L^i_{\text{mes}} := \delta_1 T_{m\to n} + (\frac{1}{\delta_2} - 1) \tag{3}$$

3.2.3 Macro-level loss

The macro-level loss L_{mac} computes how well the candidate phrases connect with each other. We consider how smooth the transition is from the previous progression to the current progression using the same transition loss as in Section 3.2.2. For the i^{th} phrase and its corresponding progression candidate, L^i_{mac} denotes the macro-level loss of the i^{th} phrase

$$L^i_{\text{mac}} := \begin{cases} 0 & i = 1 \\ T_{m'\to n'} & i = 2, \cdots, p \end{cases} \tag{4}$$

Here, $T_{m'\to n'}$ is the transition loss defined the same way as in Section 3.2.2. m' and n' index progression candidates to be connected. p is the total number of phrases.

Finally, we define the total loss of the s^{th} candidate of the i^{th} phrase by a weighted sum

$$\begin{aligned} L^{i,s}_{\text{total}} = (\beta(1 - L^{i,s}_{\text{mic}}) + (1-\beta)(1 - L^{i,s}_{\text{mes}})) \\ + \max_t \{L^{i-1,t}_{\text{total}} + \alpha(1 - L^{i,s}_{\text{mac}})\}, \end{aligned} \tag{5}$$

where α, β are parameters that we can tune between 0 and 1. Then we use DP to integrate the three levels of losses and search for the optimal chord progressions which minimize the total loss L^i_{total} at $i = p$.

3.3 An Overview of AccoMontage

Based on the harmonization results from the last section, we apply AccoMontage [4] to generate complete piano accompaniments in full length. The AccoMontage system offers a hybrid search-style transfer methodology for accompaniment arrangement: given a lead melody together with a chord progression inferred from Section 3.2, it first searches for reference pieces of accompaniments by dynamic programming. It then re-harmonizes the reference accompaniment to the given chord progression by deep learning-based style transfer. Such a pipeline is inspired by common practice that delicate music textures can often be applied in bulk, instead of composing from scratch.

Specifically, reference accompaniment pieces are searched phrase by phrase based on: 1) phrase-level fitness to the melody, and 2) transition smoothness between consecutive phrases. The overall searching process is optimized by the Viterbi algorithm [24]. The re-harmonization is implemented with a VAE framework which is capable of chord-texture disentanglement [1]. By varying the chord representation, we can re-harmonize the accompaniment while keeping its texture. The whole pipeline of AccoMontage secures a delicate accompaniment arrangement with coherent and structured texture. We refer readers to the original work [4] for more technical details.

3.4 Graphical User Interface for Controllability

AccoMontage2 provides a GUI for users to select styles of harmonized chords and accompaniment textures. The process begins with uploading a MIDI file with a melody track and setting the original chord progression styles and piano texture styles. Three labels have to be assigned manually, including phrase boundaries, key, and mode (major or minor in the current version of the system).

The system will then proceed to generation. When the generation is finished, users are able to: 1) listen to and download the generated audio; 2) select a new harmonization style for each individual phrase or the whole; 3) reset the texture style and re-generate the accompaniment. Figure 4 shows the GUI interface of AccoMontage2.

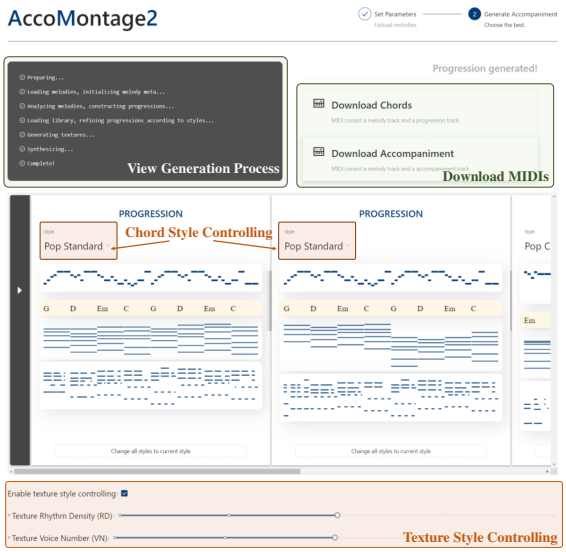


Figure 4: A screenshot of the Graphical User Interface.

4. GENERATION RESULTS

In this section, we showcase one long-term generation result of the AccoMontage2 system. We set $\alpha = 0.1$ and $\beta = 0.5$ in the harmonization algorithm and test our model on a 24-bar melody piece with phrase label `A8A8B8`. The first track in Figure 5 shows the original melody piece and the other two tracks are two different versions of accompaniment generation.

The result shows that AccoMontage2 is able to achieve high controllability in chord style and texture style. On the top of track two (I) and track three (II) in Figure 5, we present using chord notations the harmonization results

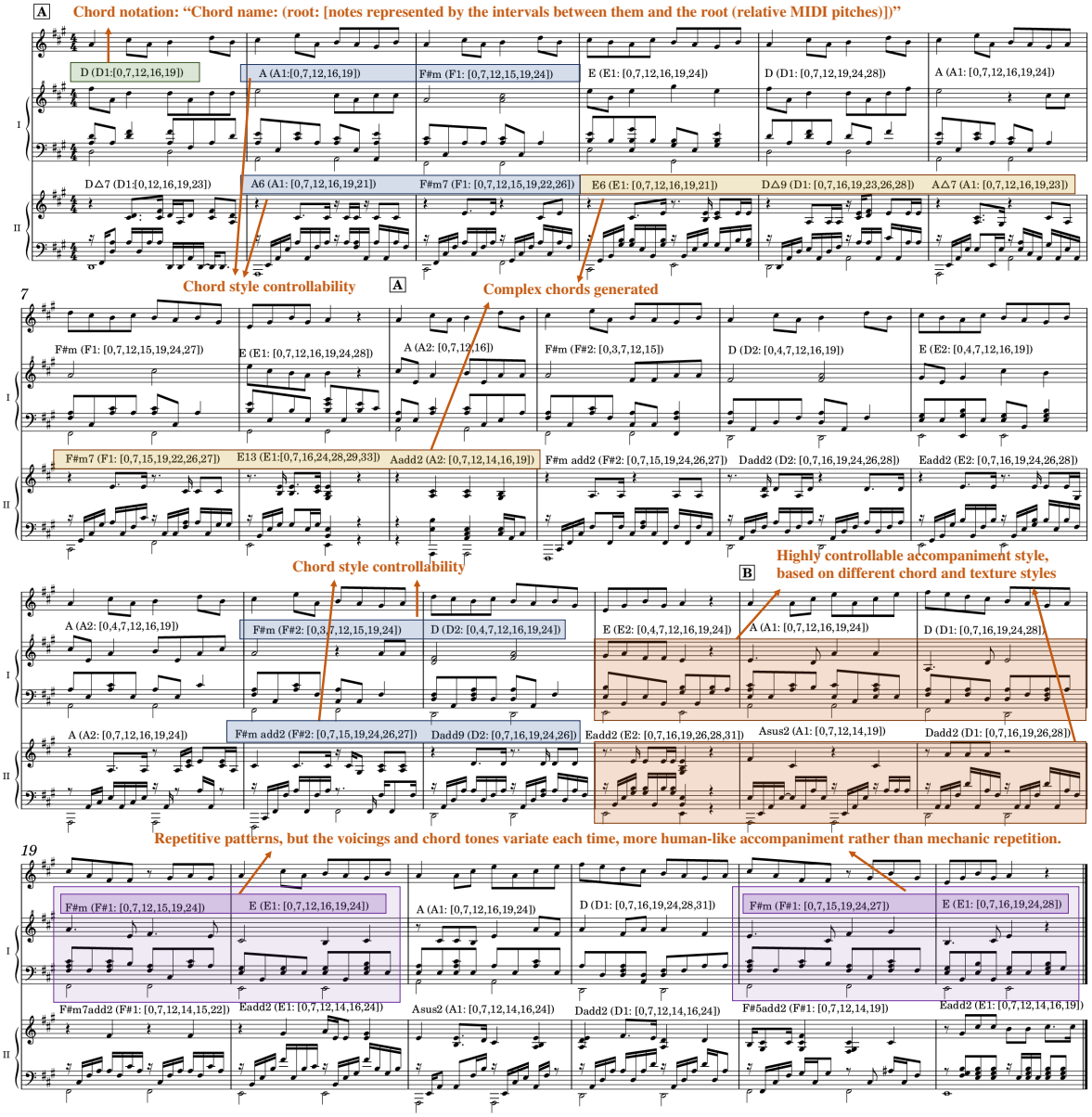


Figure 5: Harmonization and accompaniment arrangement results for *Dinners 1* from the Nottingham Dataset. The 24-bar melody has an `A8A8B8` phrase structure, and the AccoMontage2 system achieves a high degree of style controllability.

of chord style "Pop-standard" and "Pop-complex" respectively. Track two and track three are the whole accompaniment results. While track two (I) is based on "Pop-standard" and a sparse texture style, track three (II) is based on "Pop-complex" and a dense texture style. We see that both results match with the melody in harmonicity and are able to provide chord and texture variations.

5. EVALUATION

We conduct two comparative experiments to validate our AccoMontage2 system, one for harmonization and the other for accompaniment arrangement. We first show the dataset and the baseline model in Section 5.1 and 5.2. Then we present the experimental results in Section 5.3 and 5.4. Audio examples of our proposed system and ablation studies are available via our GUI link.

5.1 Dataset

For the harmonization experiment, we use Nottingham Dataset [25], which provides around 1000 pairs of the query lead melody and the ground-truth chords. For arrangement generation experiments, we use the POP909 Dataset [26], in which each piece contains a lead melody, annotated chords, and a piano accompaniment. For both data sources, we first select the pieces of meter $\frac{2}{4}$ and $\frac{4}{4}$ and then randomly sample 6 pieces for our experiment. We manually annotate their phrase segmentation and only allow the phrase length of 4 bars and 8 bars.

5.2 Baseline Method

We apply the chord generation model [9] based on bidirectional long short-term memory networks (BLSTM) [27] as our baseline model. This model takes a symbolic melody

with the information of time signature, measure, and key as input, and outputs a harmonization result of a sequence of major and minor triads. The BLSTM model considers temporal dependencies by storing both past and future information, reflecting musical context in both forward and backward directions, It is trained on a lead sheet database provided by Wikifonia.org and has achieved reasonable results quantitatively and qualitatively.

5.3 Harmonization Results

We conduct a survey to evaluate the harmonization performance of our model. Our survey has 6 groups of harmonization results and each subject is required to listen to 3 (chosen randomly). Within each group, the subject first listens to a single melody. Each melody is a full-length song randomly selected from the Nottingham Dataset, with an average length of 32 bars (64 seconds). We harmonize the melody using our model and the BLSTM baseline, and additionally acquire an original harmonization using the ground truth chord labels. The subjects are then required to evaluate all three versions of harmonization. The rating is based on a five-point scale from 1 (very poor) to 5 (very high) according to three criteria:

1. **Harmonicity**: How well do the melody and chords stay in harmony with each other;
2. **Creativity**: How creative the harmonization is;
3. **Musicality**: The overall musicality.

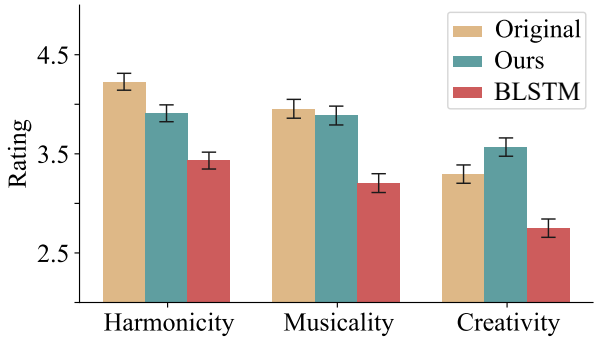


Figure 6: Subjective evaluation for melody harmonization.

A total of 15 subjects with diverse musical backgrounds participated in our survey and we obtain 44 effective ratings for each criterion. As shown in Figure 6, the height of the bars denotes the mean values of the ratings. The error bars stand for the mean square errors (MSEs) computed via within-subject ANOVA [28]. For harmonicity, the original receives the best rating, while our model performs significantly better than the BLSTM baseline. As for the overall musicality, our model is comparable with the original harmonization. For creativity, our model reaches the best. Note that our model supports complex harmonization with voice leading and ninth chords, while the original and the baseline support up to seventh chords in plain root position. Such evaluation results demonstrate that our model introduces *tension* to "flavor" the music while the overall musicality is not affected. For all three criteria, the rating results are statistically significant (p-value $p < 0.05$).

5.4 Accompaniment Generation Results

We conduct another survey to evaluate our model in terms of overall accompaniment generation. In our survey, each subject still listens to 3 groups of generation results (randomly chosen from 6 groups). Within each group, the subjects first listen to a 32-bar melody randomly selected from the POP909 Dataset. Our model generates piano accompaniment through the complete AccoMontage2 pipeline. For the BLSTM baseline, we feed its harmonization results to AccoMontage and obtain the baseline accompaniment. As POP909 contains piano arrangements created by professional musicians for each song, we also have the original accompaniment. The subjects are then required to evaluate all three versions of accompaniment based on the same scale and criteria as in Section 5.3.

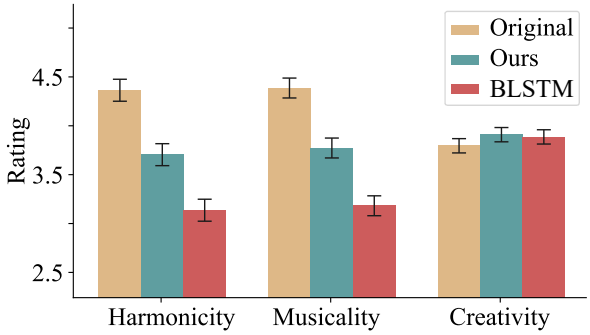


Figure 7: Subjective evaluation for accompaniment arrangement.

We collect a total of 44 effective ratings for each criterion. Figure 7 shows the evaluation results in the same format as in Section 5.3. We report a significantly better performance of our model compared with the BLSTM baseline in harmonicity and musicality ($p < 0.05$), and a marginally better performance than both the baseline and the original in creativity.

6. CONCLUSION AND FUTURE WORK

In conclusion, we contribute a pipeline of algorithms to automatically harmonize and arrange piano accompaniments for melodies of whole-piece popular and folk songs. The system is named after AccoMontage2, built upon its original version which uses a hybrid approach to select accompaniment candidates, edit the candidates using style transfer, and then concatenate them into an organic whole. AccoMontage2 contains two novel modules: a state-of-the-art harmonizer and a GUI for controllable arrangement via interfering in the harmonization and arrangement styles.

In the future, we plan to further optimize the arrangement pipeline by: 1) automatically labeling the melody phrase, 2) extending the model capability to deal with triple meters, and 3) exploring full-band arrangement.

7. REFERENCES

[1] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 662–669.

[2] G. Xia, "Expressive collaborative music performance via machine learning," Ph.D. dissertation, Carnegie Mellon University, USA, 2016. [Online]. Available: https://doi.org/10.1184/r1/6716609.v1

[3] L. Lin, G. Xia, Q. Kong, and J. Jiang, "A unified model for zero-shot music source separation, transcription and synthesis," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 381–388.

[4] J. Zhao and G. Xia, "Accomontage: Accompaniment arrangement via phrase selection and style transfer," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 833–840.

[5] C.-H. Chuan, E. Chew *et al.*, "A hybrid system for automatic generation of style-specific accompaniment," in *Proceedings of the 4th international joint workshop on computational creativity*. Goldsmiths, University of London London, 2007, pp. 57–64.

[6] I. Simon, D. Morris, and S. Basu, "Mysong: automatic accompaniment generation for vocal melodies," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2008, pp. 725–734.

[7] H. Tsushima, E. Nakamura, K. Itoyama, and K. Yoshii, "Function- and rhythm-aware melody harmonization based on tree-structured parsing and split-merge sampling of chord sequences," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR*, 2017, pp. 502–508.

[8] ——, "Generative statistical models with self-emergent grammar of chord sequences," *Journal of New Music Research*, vol. 47, no. 3, pp. 226–248, 2018.

[9] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR*, 2017, pp. 621–627.

[10] Y.-C. Yeh, W.-Y. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H.-M. Liu, H.-W. Dong, Y. Chen, T. Leong, and Y.-H. Yang, "Automatic melody harmonization with triad chords: A comparative study," *Journal of New Music Research*, vol. 50, no. 1, pp. 37–51, 2021.

[11] C.-E. Sun, Y.-W. Chen, H.-S. Lee, Y.-H. Chen, and H.-M. Wang, "Melody harmonization using orderless nade, chord balancing, and blocked gibbs sampling," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4145–4149.

[12] M. Bretan, G. Weinberg, and L. Heck, "A unit selection methodology for music generation using deep neural networks," *arXiv preprint arXiv:1612.03789*, 2016.

[13] P.-C. Chen, K.-S. Lin, and H. H. Chen, "Automatic accompaniment generation to evoke specific emotion," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.

[14] Y.-C. Wu and H. H. Chen, "Emotion-flow guided music accompaniment generation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 574–578.

[15] C.-H. Liu and C.-K. Ting, "Polyphonic accompaniment using genetic algorithm with music theory," in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–7.

[16] T. Fujishima, "Realtime chord recognition of musical sound: a system using common lisp music," in *Proceedings of the 1999 International Computer Music Conference, ICMC*. Michigan Publishing, 1999.

[17] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[18] H.-M. Liu and Y.-H. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 722–727.

[19] B. Jia, J. Lv, Y. Pu, and X. Yang, "Impromptu accompaniment of pop music using coupled latent variable model with binary regularizer," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–6.

[20] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "Popmag: Pop music accompaniment generation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1198–1206.

[21] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "Xiaoice band: A melody and arrangement generation framework for pop music," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2837–2846.

[22] N. Kotoulas, "Supersize your midi collection." [Online]. Available: https://www.pianoforproducers.com/nikos-ultimate-midi-pack/

[23] L. Trulla, N. Di Stefano, and A. Giuliani, "Computational approach to musical consonance and dissonance," *Frontiers in Psychology*, vol. 9, p. 381, 04 2018.

[24] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[25] E. Foxley, "Nottingham database," [EB/OL], https://ifdo.ca/~seymour/nottingham/nottingham.html Accessed May 25, 2021.

[26] 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 *Proceedings of 21st International Conference on Music Information Retrieval, ISMIR*, 2020.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] H. Scheffe, *The analysis of variance*. John Wiley & Sons, 1999, vol. 72.

SUPERVISED AND UNSUPERVISED LEARNING OF AUDIO REPRESENTATIONS FOR MUSIC UNDERSTANDING

Matthew C. McCallum **Filip Korzeniowski** **Sergio Oramas**
Fabien Gouyon **Andreas F. Ehmann**
SiriusXM, USA

ABSTRACT

In this work, we provide a broad comparative analysis of strategies for pre-training audio understanding models for several tasks in the music domain, including labelling of genre, era, origin, mood, instrumentation, key, pitch, vocal characteristics, tempo and sonority. Specifically, we explore how the domain of pre-training datasets (music or generic audio) and the pre-training methodology (supervised or unsupervised) affects the adequacy of the resulting audio embeddings for downstream tasks.

We show that models trained via supervised learning on large-scale expert-annotated music datasets achieve state-of-the-art performance in a wide range of music labelling tasks, each with novel content and vocabularies. This can be done in an efficient manner with models containing less than 100 million parameters that require no fine-tuning or reparameterization for downstream tasks, making this approach practical for industry-scale audio catalogs.

Within the class of unsupervised learning strategies, we show that the domain of the training dataset can significantly impact the performance of representations learned by the model. We find that restricting the domain of the pre-training dataset to music allows for training with smaller batch sizes while achieving state-of-the-art in unsupervised learning—and in some cases, supervised learning—for music understanding.

We also corroborate that, while achieving state-of-the-art performance on many tasks, supervised learning can cause models to specialize to the supervised information provided, somewhat compromising a model's generality.

1. INTRODUCTION

In this work, we consider a broad array of classification and labelling tasks under the umbrella of music understanding. Such tasks include the labelling of genre, origin, mood, musical key, instruments, era, emotion and pitch present in music. These tasks have many applications in industry, particularly in music streaming and recommendation services where automated understanding of audio can assist in a range of tasks such as organizing, filtering and personalizing content to a listener's taste and context.

Recent research in automated audio understanding has focused on training convolutional [1–12] and / or attention based [13–21] networks on moderately large collections of frequency-domain audio [2,5–8,12–14,18,20,21], time-domain audio [3, 10, 11, 19] or multi-format / multi-modal [4, 9, 21, 22] data. Such models are often trained on tags encompassing some of the musical labels listed above, achieving promising results [1,3,6,11,13–15,22]. More recent works propose unsupervised strategies for music understanding such as contrastive learning [2, 4, 5, 8–10, 21] or predictive / generative approaches [7, 20, 21, 23]. Unsupervised strategies are appealing because they require no annotated data and generalize well to new tasks [2, 21], but lag the performance of supervised learning at a similar scale [2, 10]. Generative learning strategies [7, 23] have been shown to achieve competitive, and sometimes state-of-the-art (SOTA), performance in several music understanding tasks [24], although, currently there is no evaluation demonstrating the effectiveness of this approach to any of the aforementioned approaches, at comparable scale.

Modern music streaming services have very large music catalogs that amount to many petabytes of audio data if uncompressed. Due to the scale of this data it is desirable to build models that are efficiently scalable, and understand audio in a general enough way that, as needs or requirements change, they may be used to solve novel problems without reprocessing such data. Models in the order of 10M or 100M parameters are currently relatively cost-effective to both train and apply inference to industry-scale catalogs, whilst models consisting of billions of parameters, e.g. that evaluated in [24], are typically impractical, or very expensive, for both training and inference.

More recently, research has adopted approaches producing generalized audio embeddings [2, 4–10, 12, 18, 19, 25] in a supervised or unsupervised way, by training models on large amounts of labelled or unlabelled audio. When such models are applied to novel audio, the internal state of the models has been found to contain much of the information necessary for previously unseen tasks. This is demonstrated by training shallow classifiers (probes) on embeddings consisting of the activations of a given model layer, that map these values to a downstream task. Such an approach achieves competitive results using either unsuper-

vised [25] or supervised [6] learning. Most importantly, the embeddings on which the probes are trained are many orders of magnitude smaller than the audio itself and only need to be computed once per audio file. Such embeddings can be stored efficiently, and downstream classifiers can be trained with significantly less resources. The excellent performance, generality, and scalability of this approach are crucial factors for its utility in industry.

This approach to audio understanding has been highlighted in recent benchmarks such as HARES [2] and HEAR [26], where embeddings are evaluated across a number of audio understanding tasks pertaining to a range of content types. Any score aggregation across these benchmarks to determine a "best" embedding is difficult due to the disparate range of metrics employed and furthermore, may obfuscate the strengths and weakness of any given approach. However, evaluating across a common range of tasks can be useful in comparing such strengths and weaknesses. We find that the current tasks evaluated in the HEAR and HARES benchmarks are lacking in evaluation on music content. Wrt. polyphonic music, the HARES benchmark includes only the Magnatagatune dataset, and the HEAR benchmark includes only GTZAN genre and music / speech datasets. While other public music datasets exist, such benchmarks are somewhat limited by the requirement to provide access to the audio of all datasets.

Here, we do not intend to establish a new open benchmark, but investigate the effectiveness of supervised and unsupervised learning for audio embeddings employed specifically for music understanding, across as broad an array of tasks as is available within time and resource constraints. For supervised learning we train models on large scale datasets of annotated magnitude log-mel spectrograms both in the music domain and in the general audio domain. For unsupervised learning we train contrastive models using SimCLR loss [27, 28] on the same sets of magnitude log-mel spectrograms, excluding annotations.

The contributions of this work are as follows: we provide a broad analysis of supervised and unsupervised learning strategies for pre-training audio models for music understanding; we show that for multilabel / multiclass classification of music, large-scale supervised learning on music data achieves SOTA performance, in many cases outperforming both prior SOTA and unsupervised learning by significant margins; we show that supervised learning on labelled music data does not generalize as well as unsupervised learning to novel tasks not covered in those labels; finally, we show that the domain of pre-training audio datasets has a significant impact on the performance of embeddings, particularly for unsupervised learning.

2. PRE-TRAINING METHODOLOGY

To achieve the objectives outlined in Section 1, we follow a familiar transfer learning paradigm (Figure 1) where models are pre-trained using supervised or unsupervised learning. Thereafter, the frozen activations from a layer of that model, forming embeddings, $\mathbf{z}$, are mapped to a downstream task using a simple network $p(\mathbf{z})$.

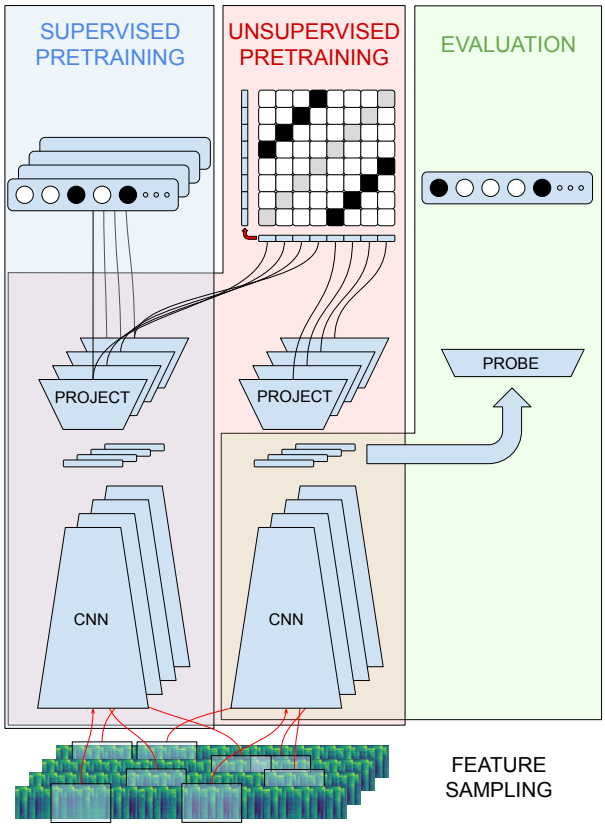


Figure 1. System diagram of both pre-training approaches employed in this paper, and evaluation.

2.1 Supervised and Unsupervised Training

In the supervised setting we learn a function $f(\mathbf{X}) \Rightarrow \hat{\mathbf{y}}$ mapping features $\mathbf{X}$ (log-mel spectrograms) to binary labels, $\mathbf{y}$, by applying Adam optimization [29] to the binary cross-entropy loss function,

$$L_s(\mathbf{y}, \hat{\mathbf{y}}) = \frac{-1}{NK} \sum_{i=0}^{N-1} \mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i),$$

where batch size $N = 512$, and K is the number of labels.

In the unsupervised setting, we employ the SimCLR objective [27, 28], which has been shown to provide promising results for both music and audio understanding [2, 10]. The SimCLR objective employs correlated (positive) pairs of samples by mapping each feature to an embedding space, $f(\mathbf{X}) \Rightarrow \mathbf{z} \in \mathbb{R}^m$, with embedding dimensionality $m = 1728$. A projector, then maps the embedding space to a loss space $h(\mathbf{z}) \Rightarrow \mathbf{v} \in \mathbb{R}^n$, with dimensionality $n = 1024$. Here, each element is then compared to all other elements in a batch via distance function, $d(\mathbf{v}_i, \mathbf{v}_j) = \mathbf{v}_i \cdot \mathbf{v}_j / \|\mathbf{v}_i\| \|\mathbf{v}_j\|$. The loss is then computed as the normalized temperature-scaled cross entropy,

$$L_u(\mathbf{v}_i, \mathbf{v}_j) = -\log \frac{\exp\left(d(\mathbf{v}_i, \mathbf{v}_j)/\tau\right)}{\sum_{k=0}^{2N-1} \mathbb{1}_{[k \neq i]} \exp\left(d(\mathbf{v}_i, \mathbf{v}_k)/\tau\right)},$$

which is summed across $2N$ examples in $N = 1920$ positive pairs, where $i = j$, for all values of both $i \in [0, N-1]$