ROLANDS: Automatic Arrangement for Solo Guitar from Audio to Tablature

Ziyang Hu, Austin Liu

November 2025

Abstract

Arranging music for solo guitar requires reinterpreting complex textures within the constraints of six strings, which is a process that demands musical insight. While deep learning has achieved remarkable success in automatic music transcription and piano cover generation, research on automatic arrangement for solo guitar remains limited. This work introduces ROLANDS (Representation of Orchestration Listened and Arrangement with Network for Designated Strings), the first end-to-end architecture for automatic arrangement from audio to guitar tablature. ROLANDS consists of three modules: (1) an Extractor that converts song audio into MIDI-like feature representations using a hFT-Transformer; (2) a Conditioner that predicts the capo position and individual string tunings via multi-label learning with cardinality regularization; and (3) a Translator that employs a T5-based encoder–decoder to generate symbolic tablature sequences conditioned on the predicted tuning and capo tokens. ROLANDS is designed to be trained using aligned datasets of song audio, performance MIDI, and corresponding solo-guitar tablatures. Objective metrics will evaluate the similarity between the arrangement and the original song, as well as the musical quality of the arrangement. A listener study will also assesses perceived musicality and authenticity. ROLANDS aims to bridge the gap between symbolic music modeling and automatic arrangement for solo guitar.

1 Introduction

The practice of arranging music for solo guitar has a rich history dating back to the early 19th century. Guitarists have expanded the instrument's repertoire by adapting works originally composed for other media such as solo piano, orchestral ensembles, and vocal-instrumental music [1, 2, 3, 4, 5, 6]. Today, solo guitar arrangements continue to flourish on digital platforms, reflecting both artistic creativity and widespread public interest [7]. However, producing a high-quality arrangement remains a demanding task. As guitarist Roland Dyens observed, the challenge of arranging "is the restitution of the work's original essence across the space, forcibly restrained, of the six strings of the instrument" (Dyens, 1998, as cited in [8], see also [9]).

In the context of solo instrumental music, the term "arrangement" usually involves adaptation across different media or recomposition, yet it is frequently conflated with the term "transcription" [9]. Recent advances in deep learning have stimulated research on automatic music transcription and arrangement [10, 11, 12]. However, progress in automatic arrangement for solo guitar, which is one of the most prominent solo instruments, has lagged behind state-of-the-art developments in deep learning. While numerous studies have applied deep learning techniques to automatic guitar transcription [13, 14, 15], research on automatic guitar arrangement remains limited to probabilistic and statistical methods [16, 17].

To bridge this gap, we introduce ROLANDS: Representation of Orchestration Listened and Arrangement with Network for Designated Strings. ROLANDS is the first architecture for automatic arrangement for solo guitar. Drawing inspiration from automatic piano cover generation (APCG) and automatic guitar tablature generation, the architecture of ROLANDS will contain 3 stages: (1) the Extractor that extracts musical features from input audio, (2) the Conditioner that predicts string tunings and the position of capo for the final output tablature, and (3) the Translator that translates MIDI-like sequences into representation sequences of tablatures. ROLANDS aims to generate stylistically coherent solo-guitar arrangements directly from the audio of music.

2 Background

2.1 Transformers in Music

Transformers [18] are based on self-attention mechanisms. Initially proposed for natural language processing (NLP) tasks, Transformers have also been applied to symbolic music generation tasks [19, 11, 20, 21, 22, 23]. For example, [11] introduced the hFT-Transformer for automatic piano transcription. The model employs a two-level hierarchical frequency–time Transformer architecture to capture both spectral and temporal acoustic features. The input to the hFT-Transformer is a log-mel spectrogram of piano audio, and each hierarchy outputs four matrices representing distinct musical features. The four matrices together can reconstruct a MIDI file. Beyond automatic transcription, hFT-Transformer has also been utilized for automatic arrangement tasks because of its capability to capture musical features from original song audios [24, 25].

T5 [26] is another Transformer architecture that has been successfully applied to symbolic music generation. Originally developed as an encoder–decoder model for text-to-text tasks such as translation and question answering, T5 can be adapted to music by employing tokenizers that represent symbolic music data as sequences of discrete tokens [27, 28, 29]. This enables models to treat musical sequences analogously to natural language, facilitating effective representation and generation of musical data [10, 30, 15].

2.2 Automatic Piano Cover Generation

A piano cover refers to an arrangement for piano performance based on musical components of an existing song [30]. Studies on Automatic Piano Cover Generation (APCG) have proposed frameworks that can generate arrangements at a quality level comparable to that of composers, plus robust and reproducible evaluation metrics [30, 12, 25].

[25] proposed Etude, one of the newest APCG frameworks. That 3-stage framework consists of (1) an Extractor to capture musical features of original songs [11, 24]; (2) a Structuralize stage that extracts beats timing [31]; and (3) a Decoder to synthesize the final piano cover [32]. In the experiment, Etude's Extractor and Decoder were trained separately on a pop song vs. piano cover pairs dataset, which is synchronized using the weakly-alignment method [33].

To evaluate piano covers generated by Etude, other models, and human composers, [25] introduced both objective and subjective evaluation metrics. The objective metrics measure similarity to the original song, rhythmic coherence, and the diversity of rhythmic motifs, where a performance closer to the human benchmark indicates better quality. The subjective evaluation involves a listener study, where participants rate each cover across four criteria (similarity, fluency, dynamic expression, and overall quality), where higher scores reflect better performance. Etude achieved outcomes closest to the human benchmark on objective metrics and the highest ratings in subjective evaluation.

Despite fundamental differences between guitar and piano (e.g., strings whose ranges of pitches can overlap, tuning/capo conditioning), Etude's data representation of piano covers [27], framework design, training strategy, and evaluation metrics suggest promising directions to bridge the gap in automatic arrangement for guitar.

2.3 Automatic Guitar Tablature Generation

Numerous studies explored guitar tablature generation in the contexts of composition [34, 35, 22], transcription [13, 14, 15], and arrangement [16, 17]. As noted in the introduction, research on automatic arrangement for guitar remains limited to probabilistic and statistical methods. Nevertheless, recent progress in guitar tablature generation provides conceptual and methodological parallels to those used in state-of-the-art APCG frameworks.

For example, Etude [25] employs REMI-based tokenizer [27] to encode piano covers into sequences of tokens that represent notes information such as pitches, durations, and grace notes. Analogously, attention-based architectures for guitar tablature generation adopted similar tokenizers that encode guitar tablatures [28, 35, 15], demonstrating the feasibility of adapting such representations for guitar arrangement tasks.

3 Methodology

With inspiration from musical feature extraction in APCG and token representation of music symbolic data, we propose ROLANDS. ROLANDS's architecture, as illustrated in Figure 1, contains three stages that will be trained separately: Extractor, Conditioner, and Translator. The tablatures generated by ROLANDS will be evaluated by both objective metrics and subjective listening test.

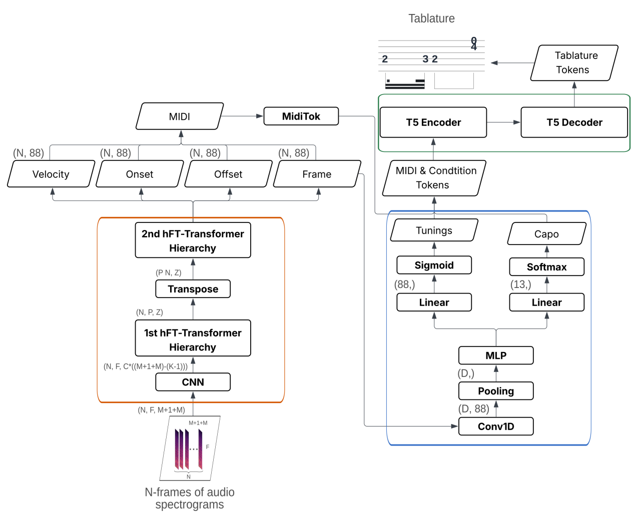

Figure 1: The architecture of ROLANDS with modules highlighted by color: Extractor (orange), Conditioner (blue), and Translator (green)

3.1 Model Architecture

3.1.1 Extractor

Similar to the counterparts in state-of-the-art architectures for APCG, such as AMP-APC [24] and Etude [25], the Extractor module of ROLANDS is also based on the hFT-Transformer [11]. The model takes as input N frames of the spectrogram, where each frame consists of F frequency bins and includes forward and backward margins of size M. Thus, the input shape is (N, F, M+1+M). Ahead of the Transformers in the first hierarchy, a convolutional neural network (CNN) with C channels and a kernel size of K is applied as a 1-D convolution to the $M + 1 + M$ dimension. The embedding vector size for Transformers is Z. The frequency dimension is converted from F to the number of pitches (88). The output of each hierarchy comprises four matrices of shape (N, 88), where 88 corresponds to the distribution of pitches. Each of these four matrices represents different acoustic features that together can reconstruct a MIDI file.

3.1.2 Conditioner

String tunings and capo position play a crucial role in guitar composition and performance. In [15], special tokens that represent capo and tuning were added into the input to condition the model. Inspired by this, ROLANDS contains a Conditioner module that predicts both the capo position and the tunings of the six strings for the outcome guitar tablature.

Previous studies on guitar tuning identification have typically formulated the task as a multi-class classification problem, where each class corresponds to a six-string tuning configuration (e.g., standard tuning, Celtic tuning, open A) [36, 37]. However, arrangements for solo guitar sometimes employ unique tunings, such as Hirokazu Sato's arrangement of Merry Christmas Mr. Lawrence by Ryuichi Sakamoto [38]. To address this limitation, the Conditioner module does not predict a single categorical tuning for all 6 strings. Instead, it estimates the tuning of each string individually by learning from the Frame matrix produced by the second hierarchy of hFT-Transformer, which encodes the activation of quantized pitches across time-processing frames.

The Frame matrix, with a shape of (N,88), represents the activation of pitches. It is first passed through a one-dimensional convolutional block to capture local pitch-wise dependencies, producing an embedding of shape (D,88), where D is the dimensionality of the pitch embedding. A pooling operation is then applied across the pitch dimension to summarize the representation into a single D-dimensional vector, which is subsequently processed by a multilayer perceptron (MLP).

The model branches into two output heads: the tuning head and the capo head. The tuning head consists of a linear layer followed by a sigmoid activation, producing a vector of length 88 that represents the probability distribution over all possible pitches, where the six highest-probability values correspond to the predicted tunings of the six guitar strings. The capo head consists of a linear layer followed by a softmax activation, outputting a 13-dimensional probability vector that represents capo positions ranging from 0 to 12.

3.1.3 Translator

Inspired by [15], the Translator module of ROLANDS employs a reduced architecture of the T5 model [26] for translation between MIDI and guitar tablature token sequences. The MIDI file is reconstructed by the 4 output matrices generated by the second hierarchy of the Extractor module. MidiTok [29] converts the reconstructed MIDI file into a sequence of representation tokens as the input of the Translator module. We also adapt the conditioning method in [15] by converting the output of the Conditioner into `CAPO<#>` and `TUNING<#,#,#,#,#,#>` tokens, which are prepended to the input sequence. The Translator outputs a sequence of tokens representing guitar tablature, which can then be decoded into a GuitarPro file [28, 39].

3.2 Training

3.2.1 Extractor

The Extractor module of ROLANDS will be trained in two stages, following the procedure of AMT-APC [24]. In the pre-training stage, the Extractor functions as an automatic music transcription (AMT) system as the hMT-Transformer in [11], learning to transcribe guitar performance audio into the MIDI representation of the same piece. In the fine-tuning stage, the pre-trained Extractor is adapted from transcription to arrangement understanding. It will be trained on paired data of original song audio and the MIDI of the corresponding guitar arrangement.

The loss calculation for both stages is derived from both [11] and [24]:

$$L_{bce}^{m} = \frac{1}{NP} \sum_{n=0}^{N-1} \sum_{p=0}^{P-1} l_{bce}(y_{n,p}^{m}, \hat{y}_{n,p}^{m}) \tag{1}$$

$$L_{cce}^{velocity} = \frac{1}{NP} \sum_{n=0}^{N-1} \sum_{p=0}^{P-1} l_{bce}(y_{n,p}^{velocity}, \hat{y}_{n,p}^{velocity}) \tag{2}$$

$$L_{ext} = \beta L_1 + (1 - \beta) L_2 \tag{3}$$

Based on [11], binary cross-entropy (l_{bce}) is used for frame, onset, and offset predictions, while velocity is trained using 128-category cross-entropy (l_{cce}). The ground-truth and predicted values of each output are denoted as y and $\hat{y}$. As adapted from [24], the loss for each hierarchy is obtained by averaging the four component losses, and the final loss of the ROLANDS Extractor combines the two hierarchical losses with a weighting ratio of β.

3.2.2 Conditioner

The Conditioner module is designed to be optimized with a combination of three core losses: capo cross-entropy, tuning multi-label binary cross-entropy, and a cardinality regularization term that enforces six predicted string tunings:

$$L_{capo} = -\sum_{j=0}^{12} \mathbf{e}_{c^\star}(j) \, \log q(j) \tag{4}$$

$$L_{tune} = -\frac{1}{88} \sum_{i=1}^{88} \Big[y_i \log p_i + (1 - y_i) \log(1 - p_i) \Big] \tag{5}$$

$$L_{card} = \Big(\sum_{i=1}^{88} p_i - 6 \Big)^2 \tag{6}$$

$$L_{cond} = \lambda_{capo} L_{capo} + L_{tune} + \lambda_{card} L_{card} \tag{7}$$

To train the capo head, which outputs a 13-dimensional probability distribution over capo positions (0–12), we will use the standard categorical cross-entropy, where $\mathbf{e}_{c^\star}$ is the one-hot vector for the ground-truth capo position and $q(j)$ is the predicted probability for capo position j. The tuning head outputs a sigmoid-activated vector of length 88, representing the probability of each pitch being an open-string tuning. $y_i \in \{0, 1\}$ indicates whether pitch i corresponds to an open string, and p_i is the predicted probability for pitch i.

Since a guitar has six strings, the model should predict exactly six active open-string pitches. To encourage this, a cardinality regularization term penalizes deviations from six active notes. This constraint ensures that the overall probability mass of open-string predictions aligns with the six-string structure of the guitar. The final Conditioner loss is computed as the sum of the tuning loss, a weighted capo classification loss, and a weighted cardinality constraint. No coefficient is applied to the tuning loss, which serves as the primary objective term.

3.2.3 Translator

We will deploy the training procedure of [15] for the Translator module of ROLANDS.

3.3 Dataset

Training the three modules of ROLANDS requires a dataset that aligns the original song audio and MIDI with the corresponding solo guitar arrangement, including its tablature, performance audio, and MIDI file. Since the Extractor processes the log-mel spectrogram as sequences of 512 frames rather than entire tracks, the dataset must be segmented and aligned at this resolution. Each 512-frame segment of the input spectrogram will be synchronized with the corresponding portion of the arrangement's tablature and MIDI, ensuring that audio–symbolic correspondences are preserved across all training samples. In addition, the pre-training of the Translator module requires a separate dataset that pairs MIDI files of guitar performance with their corresponding audio.

Previous research in computer music has produced a wide range of music datasets. Some of these datasets provide sufficient guitar tablatures [28, 40, 41], while others focus on audio and/or MIDI data of music from different performance media [42, 43, 44, 45]. Identifying and pairing songs with their corresponding solo guitar arrangements across these datasets would greatly facilitate the construction of a dataset for ROLANDS.

3.4 Evaluation

3.4.1 Objective Metrics

We will utilize Melody Chroma Accuracy (MCA) [46] to measure the similarity between the original song and the arrangement for solo guitar. This reproducible metric is implemented in a publicly available Python library [46] and has been widely adopted in subsequent studies to evaluate similarity between original pop songs and generated piano covers [30, 33, 12].

Besides the similarity to the original song, we also plan to evaluate the musical quality of the tablatures themselves. Therefore, we will also utilize two other metrics: pitch class histogram entropy (PCE) and groove consistency (GC) [47]. PCE measures the pitch class diversity. The dominance of specific pitch classes can cause low PCE scores, which indicate a stable and consistent tonality [35, 33]. GC measures the rhythmic stability across different segments (like measures). Both metrics are implemented in Python library MusPy [48].

All 3 metrics require pitch or beat sequences as input. We will utilize PyGuitarpro [49] to parse the GuitarPro files of generated tablatures to extract the input for metrics computation.

3.4.2 Subjective Evaluation

We also designed a listening test that needs recruitment of participants with different levels of music background. First, a panel of professional guitarists and musicians is needed to select pairs of songs that are stylistically and structurally similar to those in the training set to ensure fair comparison. Then listeners will be recruited, which include professional musicians, trained amateurs, and individuals with no formal musical background. For each pair selected by music professionals, listeners will be presented with two audio excerpts: one from a human-composed guitar arrangement and another generated by ROLANDS for a similar song. Listeners will be asked to identify which arrangement they believe was created by a human and to rate each sample on musicality, similarity to the original song, and perceived playability on guitar. By analyzing preferences and confidence levels across groups, this survey will provide a human-centered measure of quality that complements objective evaluation metrics.

4 Discussion

4.1 Challenges

One of the primary challenges is dataset construction. Training ROLANDS would require a sufficient number of paired examples between the audio of original songs and the tablature of their corresponding solo-guitar arrangements. Although existing datasets can be leveraged by extracting pairs of song audio and solo-guitar arrangement tablatures, it remains uncertain whether existing datasets can provide enough pairs. In addition, musicians sometimes introduce structural modifications in their arrangements, such as removal of repetitions or modulations [9]. These alterations increase the difficulty to align the song audio to the corresponding parts of tablature when they are segmented into smaller sequences.

Another difficulty comes from the availability of digital tablature data. Many high-quality arrangements exist only in PDF format and may be written as sheet music instead of tablature, whereas the training procedure requires structured format of tablatures such as GuitarPro [39]. Currently, there is no reliable pipeline to convert PDF files of tablature or sheet music directly into GuitarPro format. Overcoming this bottleneck will expand the dataset scale and diversity that benefit the training of ROLANDS.

The complexity of guitar performance technique also introduces additional modeling challenges. To represent the original orchestration, guitarists may creatively deploy unusual techniques in their arrangement [6]. Therefore, existing tokenizers for guitar tablatures [28, 15] may need an extension of the vocabularies to include new representation tokens for those techniques.

4.2 Future Directions

Similar to previous research on guitar tablature generation [35, 15], the guitar tablature is represented as one sequence of special tokens in ROLANDS. For future work, automatic guitar arrangement can be extended beyond sequence-to-sequence translation tasks to incorporate string-aware modeling, where six parallel sub-architectures represent individual strings. Each sub-architecture, conditioned by values of capo position and tuning of the string, would process the longitudinal order of notes with corresponding playing techniques on its respective string, while cross-string attention mechanisms capture chord and harmonic relationships among strings.

There is also potential for innovation in automatic tablature extraction and data augmentation. Integrating optical music recognition (OMR) and symbolic music modeling could enable automatic conversion from PDF scores of solo guitar arrangements into GuitarPro format. The automatic digitization of existing collections, including handwritten or scanned scores, would expand the diversity and volume of training data for symbolic music generation and bridge the gap between traditional sheet music archives and structured symbolic music data formats.

References

[1] S. Beavers, *Homage in the solo guitar music of Roland Dyens.* The Florida State University, 2006.

[2] G. Zangari, "Mauro giuliani (1781-1829): instrumental and vocal style in le sei rossiniane," Master's thesis, University of Sydney., 2013.

[3] C. STAVRAT, "Suite pictures at an exhibition–arrangements and transcriptions for orchestra and chamber groups," *Artes. Journal of musicology*, no. 14, pp. 145–153, 2014.

[4] Y.-Y. Chien, *A Comparison of the Piano and Guitar Versions of Isaac Albéniz's Spanish Suite Op. 47*. PhD thesis, University of Oregon, 2016.

[5] D. Lee, B. Baker, and N. Haywood, "Idiosyncrasies in australian guitar culture: An examination of developments within the paradigm of popular music," 2018.

[6] K. Koltai, "The tradition of illusion: Guitar arrangement as a post-canonical patchwork," *Soundboard Scholar*, vol. 9, no. 1, p. 15, 2024.

[7] C. Abidin, "Qualities of internet celebrity," in *Internet Celebrity: Understanding Fame Online*, pp. 19–36, Emerald Publishing Limited, 2018.

[8] M. Birch, "Jazz mind and classical hands-roland dyens and his style of arranging and performing," *Research Paper, Massey University*, 2005.

[9] G. C. L. Vincens, *The arrangements of Roland Dyens and Sergio Assad: Innovations in adapting jazz standards and jazz-influenced popular works to the solo classical guitar*. PhD thesis, The University of Arizona, 2009.

[10] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "Mt3: Multi-task multitrack music transcription," *arXiv preprint arXiv:2111.03017*, 2021.

[11] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W.-H. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time transformer," *arXiv preprint arXiv:2307.04305*, 2023.

[12] T. Y. Yip and C. Chau, "Music2midi: Pop music to midi piano cover generation," in *International Conference on Multimedia Modeling*, pp. 101–113, Springer, 2024.

[13] S. Kim, T. Hayashi, and T. Toda, "Note-level automatic guitar transcription using attention mechanism," in *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 229–233, IEEE, 2022.

[14] S. Chieppa, P. Brutti, and R. P. Paiva, "Automatic guitar transcription with deep neural networks," *IEEE Access*, 2025.

[15] A. Hamberger, S. Murgul, J. Schmidt, and M. Heizmann, "Fretting-transformer: Encoder-decoder model for midi to tablature transcription," *arXiv preprint arXiv:2506.14223*, 2025.

[16] A. Moreau, *Automatic generation of solo-guitar arrangements from lead sheets*. PhD thesis, Technische Universität Wien, 2010.

[17] S. Ariga, S. Fukayama, and M. Goto, "Song2guitar: A difficulty-aware arrangement system for generating guitar solo covers from polyphonic audio of popular music.," in *ISMIR*, pp. 568–574, Suzhou, 2017.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] 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)*, pp. 1–5, IEEE, 2023.

[20] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, "Figaro: Controllable music generation using learned and expert features," in *The Eleventh International Conference on Learning Representations*, 2023.

[21] X. Qu, Y. Bai, Y. Ma, Z. Zhou, K. M. Lo, J. Liu, R. Yuan, L. Min, X. Liu, T. Zhang, *et al.*, "Mupt: A generative symbolic music pretrained transformer," *arXiv preprint arXiv:2404.06393*, 2024.

[22] P. Sarmento, *Guitar tablature generation with deep learning*. PhD thesis, Queen Mary University of London, 2024.

[23] D.-V.-T. Le, L. Bigo, D. Herremans, and M. Keller, "Natural language processing methods for symbolic music generation and information retrieval: A survey," *ACM Computing Surveys*, vol. 57, no. 7, pp. 1–40, 2025.

[24] K. Komiya and Y. Fukuhara, "Amt-apc: Automatic piano cover by fine-tuning an automatic music transcription model," *arXiv preprint arXiv:2409.14086*, 2024.

[25] T.-Y. Che and Y.-J. Joung, "Etude: Piano cover generation with a three-stage approach–extract, structuralize, and decode," *arXiv preprint arXiv:2509.16522*, 2025.

[26] 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," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[27] 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*, pp. 1180–1188, 2020.

[28] P. Sarmento, A. Kumar, C. Carr, Z. Zukowski, M. Barthet, and Y.-H. Yang, "Dadagp: A dataset of tokenized guitarpro songs for sequence models," *arXiv preprint arXiv:2107.14653*, 2021.

[29] N. Fradet, J.-P. Briot, F. Chhel, A. E. F. Seghrouchni, and N. Gutowski, "Miditok: A python package for midi file tokenization," *arXiv preprint arXiv:2310.17202*, 2023.

[30] 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)*, pp. 1–5, IEEE, 2023.

[31] J. Zhao, G. Xia, and Y. Wang, "Beat transformer: Demixed beat and downbeat tracking with dilated self-attention," *arXiv preprint arXiv:2209.07140*, 2022.

[32] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, *et al.*, "Gpt-neox-20b: An open-source autoregressive language model," *arXiv preprint arXiv:2204.06745*, 2022.

[33] C.-P. Tan, S.-H. Guan, and Y.-H. Yang, "Picogen: generate piano covers with a two-stage approach," in *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pp. 1180–1184, 2024.

[34] Y.-H. Chen, Y.-H. Huang, W.-Y. Hsiao, and Y.-H. Yang, "Automatic composition of guitar tabs by transformers and groove modeling," *arXiv preprint arXiv:2008.01431*, 2020.

[35] P. Sarmento, A. Kumar, Y.-H. Chen, C. Carr, Z. Zukowski, and M. Barthet, "Gtr-ctrl: instrument and genre conditioning for guitar-focused music generation with transformers," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, pp. 260–275, Springer, 2023.

[36] V. Khatri and L. Dillingham, "Guitar tuning identification," *University of Rochester, Department of Electrical and Computer Engineering*, 2020.

[37] E. Hulme, D. Marshall, K. Sidorov, and A. Jones, "Acoustic classification of guitar tunings with deep learning," in *Proceedings of the 11th International Conference on Digital Libraries for Musicology*, pp. 6–14, 2024.

[38] K. Muraji and H. Sato, *Muraji, Kaori. Guitar Solo Collection Vol. 2.* Gendai Guitar, 1990.

[39] A. Music, "Guitar pro," *Computer software. Recuperado em*, vol. 23, 1997.

[40] X. Riley, Z. Guo, D. Edwards, and S. Dixon, "Gaps: A large and diverse classical guitar dataset and benchmark transcription model," *arXiv preprint arXiv:2408.08653*, 2024.

[41] Y. Zang, Y. Zhong, F. Cwitkowitz, and Z. Duan, "Synthtab: Leveraging synthesized data for guitar tablature transcription," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1286–1290, IEEE, 2024.

[42] J. J. Bosch, R. Marxer, and E. Gómez, "Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music," *Journal of New Music Research*, vol. 45, no. 2, pp. 101–117, 2016.

[43] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A dataset for guitar transcription.," in *ISMIR*, pp. 453–460, 2018.

[44] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm," *arXiv preprint arXiv:1906.10606*, 2019.

[45] H. Bang, E. Choi, M. Finch, S. Doh, S. Lee, G.-H. Lee, and J. Nam, "Piast: A multimodal piano dataset with audio, symbolic and text," *arXiv preprint arXiv:2411.02551*, 2024.

[46] 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 *ISMIR*, vol. 10, p. 2014, 2014.

[47] S.-L. Wu and Y.-H. Yang, "The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures," *arXiv preprint arXiv:2008.01307*, 2020.

[48] H. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "Muspy: A toolkit for symbolic music generation," *arXiv preprint arXiv:2008.01951*, 2020.

[49] S. Abalumov, "Pyguitarpro (2014)," 2022.