IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 May 2025, accepted 4 June 2025, date of publication 30 June 2025, date of current version 8 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3583646

RESEARCH ARTICLE

Automatic Guitar Transcription With Deep Neural Networks

SIMONE CHIEPPA[1], PIERPAOLO BRUTTI[1], AND RUI PEDRO PAIVA[2]

[1]Facoltà di Ingegneria dell'Informazione, Informatica e Statistica, Dipartimento di Scienze Statistiche, Sapienza Università di Roma, 00185 Rome, Italy
[2]Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, LASI, University of Coimbra, 3030-290 Coimbra, Portugal

Corresponding author: Simone Chieppa (simone.chieppa99gmail.com)

This work was supported in part by the FCT-Foundation for Science and Technology, I.P., financed with the National Funds (PIDDAC) via Portuguese State Budget under Project MERGE-DOI: 10.54499/PTDC/CCI-COM/3171/2021; and in part by European Social Fund through the Regional Operational Program Centro 2020 under Project CISUC-UID/CEC/00326/2020.

ABSTRACT Guitar tablature is a form of musical notation designed specifically for fretted string instruments. It is crucial for musicians, especially beginners, as it provides an easy-to-read format that visually represents the placement of the fingers on the instrument, making it more accessible to learn and play songs accurately. Automatic transcription of guitar music into tablature is a complex task in Music Information Retrieval, which has been enhanced by recent advances in deep learning. This article builds on a state-of-the-art note-level transcription model that uses a self-attention mechanism. The model incorporates beat-informed quantisation to accurately convert audio signals into tablature, overcoming challenges such as determining where notes are played on the guitar neck. In replicating this model, its properties have been investigated, and the results have been analysed. A key aspect of this article is the development of a new dataset designed to assess the robustness of the model in different scenarios. This is crucial due to the limited availability of data in this area. Through extensive experimentation and analysis, this study evaluates the performance of the model on unseen data, identifying its strengths and areas for improvement. In addition, this article provides insights into the mechanism of self-attention and its effectiveness in tasks such as Automatic Music Transcription. The model was tested with multiple attention heads to study their impact on performance, but this modification did not show significant improvement. Therefore, it suggests that other areas, such as improving the quality and quantity of data, may be more crucial to improve performance.

INDEX TERMS Automatic guitar transcription, deep learning, MIDI standard, self-attention, tablature.

I. INTRODUCTION

Automatic Music Transcription (AMT) represents one of the most fascinating and complex challenges in the field of audio signal processing and music information retrieval (MIR). AMT involves the process by which an automatic system converts a musical audio signal into readable musical notation. This notation can be in the form of traditional sheet music or more specific formats such as Musical Instrument Digital Interface (MIDI) standard or guitar tablature (tab). The ability to automatically transcribe music facilitates learning and teaching, improves music production workflows, and opens new possibilities for computational musicology.

Interest in AMT has grown exponentially due to advances in deep learning. These technologies have enabled the development of more sophisticated models capable of accurately transcribing music by learning complex representations of audio signals. This has revolutionised AMT by producing more accurate and reliable results, making it an important area of study.

The guitar, a widely popular and versatile instrument, poses unique challenges for automatic transcription. Unlike the piano, where each note corresponds to a single key, guitar notes can be played in various positions on the fretboard. This requires not only recognising the notes but also determining

The associate editor coordinating the review of this manuscript and approving it for publication was Syed Mohammad Zafaruddin.

their correct positions on the neck, adding complexity to the transcription process.

Guitar tablature is an essential tool for guitarists. Shows the exact string-fret position where the note should be played. Tablature is more intuitive than traditional sheet music, especially for those without formal musical training.

Therefore, studying and improving a system for automatic guitar transcription into accurate tablatures holds significant practical value and addresses a critical need in music education and performance.

A. OBJECTIVES AND APPROACHES

This study focuses on the implementation of a note-level automatic guitar transcription model with a self-attention mechanism designed by Kim et al. [1]. The model employs a beat-informed quantisation to generate a note-level transcription and self-attention mechanism to improve transcription accuracy by effectively capturing long-range dependencies in the audio signal.

In reproducing this model, its characteristics have been thoroughly investigated and the results have been analysed. A key aspect of this research was the creation of a specific dataset to further test the model and evaluate its generalisation capabilities. Given the scarcity of data available in the field of automatic guitar transcription, the creation of this dataset was crucial.

Using the created dataset, several experiments were conducted to test the model's performance on unseen data, examining its accuracy and ability to generalise beyond the training data.

B. RESULTS AND CONTRIBUTIONS

The results of this research show that Kim's et al. model [1] is very promising in the field of automatic guitar transcription. However, by analysing the strengths and weaknesses of the model, it became clear that additional and varied data were needed for more comprehensive testing. This realisation led to the creation of a new dataset, which is a significant contribution to the field, especially given the scarcity of such detailed data.

This new dataset is particularly important because it includes MIDI transcriptions for each guitar string, a feature that allows tablatures to be derived with a high degree of accuracy. Such detailed data is extremely rare and represents a significant added value for research.

A key contribution of this article is to understand the potential and effectiveness of Kim's et al. model when tested on unseen data, and to identify areas for future improvements. The application of a self-attention mechanism was shown to improve transcription accuracy by effectively capturing long-range dependencies in the audio signal. In addition, the model was tested with two attention heads to investigate whether increased attention to short and long term dependencies would lead to better performance. However, the results did not show a significant improvement, suggesting that increasing the number of attention heads alone does not necessarily improve performance. Other factors, such as the variety and amount of training data, may be more important in improving transcription accuracy.

II. LITERATURE REVIEW

This section starts with a review of the main current works on Automatic Music Transcription (AMT), in general, and guitar transcription in particular. We then review the datasets that are available to conduct such tasks.

A. AUTOMATIC MUSIC AND GUITAR TRANSCRIPTION

Over the past decades, researchers have developed numerous approaches to AMT, as highlighted by Benetos et al. [2] and Bhagwat et al. [3]. This section is based on their comprehensive reviews, which outline various techniques such as traditional signal processing methods [4], [5], probabilistic modelling [6], Bayesian approaches [7], non-negative matrix factorization [8], [9], [10], and neural networks [11], [12].

Each of these methods has its own strengths and weaknesses. Traditional signal processing methods are simple, fast, and generalise well across different instruments. In contrast, deep neural network methods tend to achieve higher accuracy, particularly with specific instruments such as the piano. Bayesian approaches offer comprehensive modelling of the sound generation process, but these models can be very complex and slow.

As outlined in [13], given the large amount of data available and the well-defined sound structure, music transcription research has typically focused on transcribing solo piano recordings. As a result, there are a large number of transcription models whose success is based on handmade representations for piano transcription, such as [14], [15], and [16]. Single-instrument transcription models have also been developed for other instruments such as guitar and percussion ([17], [18], [19]), although these instruments have received less attention than the piano.

In the last few years, with the improvement of deep learning models, some multi-instrument transcription systems have been developed. In [20], Manilow et al., presented the Cerberus model, which simultaneously performs source separation and transcription for a fixed and predefined set of instruments. In [21] the authors also developed a model performing both separation and transcription. ReconVAT [22] uses an approach based on U-Net and unsupervised learning techniques to perform transcription on low-resource datasets. However, the model outputs only a single piano roll that combines all instruments into a single "track". A similar limitation applies to the early transcription system introduced alongside the MusicNet dataset [23].

In 2022, a study from Google Research addressed these problems by introducing Magenta MT3 [13], which generates a sequence of events that represent notes from an arbitrary number of instruments with each note explicitly assigned

to an instrument. This model learns to directly detect the presence or absence of instruments using audio spectrograms.

MT3 builds upon [24], which implements an encoder-decoder Transformer architecture originally designed for solo piano transcription, adapting it to handle polyphonic music involving any number of instruments. The approach uses standard components such as Mel spectrogram inputs, a standard Transformer configuration from T5, and MIDI-like output events.

Although guitar-specific models have historically received less attention due in part to the scarcity of available data, significant research has been conducted in this area. Andrew Wiggins and Youngmoo E Kim [25] give a concise overview of the recent works in this field. Previous studies have addressed the problem of automatic tablature transcription by performing multi-pitch estimation and then arranging tablature by optimising the physical ease of play. Burlet and Fujinaga outlined a framework for a guitar transcription web application [26]. Their framework combines a preexisting polyphonic transcription algorithm [27] with a novel algorithm that creates a directed acyclic weighted graph of string-fret combinations and finds an optimal path using the A* search algorithm. In [28], Burlet and Hindle utilized the previously mentioned tablature arrangement algorithm in conjunction with a multi-pitch estimation algorithm, using deep belief networks to learn framewise pitch estimates from the Short-Time Fourier Transform (STFT). In [29], the authors uses latent harmonic allocation (LHA) for multi-pitch estimation and arranged tablature by filtering the LHA results based on a set of spatial and temporal playability constraints. In [30], the authors applied knowledge of each guitarist's proficiency to further filter the pitches estimated from LHA and generate a sensible tablature arrangement.

In [31], Barbancho et al. utilised peak-picking from the magnitude spectrum to identify fundamentals and partials for candidate pitches. They then analysed the inharmonicity of the partials to determine the most probable string each pitch was played on. In [32], Kehling et al. proposed a system that applies the Blind Harmonic Adaptive Decomposition algorithm proposed in [33] for multi-pitch estimation. After aggregating framewise pitch estimates into note estimates, their system applies Support Vector Machines to classify various performance parameters, including the guitar string each note was played on.

More recently, Wiggins and Kim introduced TabCNN [25], a convolutional neural network method designed for estimating guitar tablature from solo acoustic guitar performances.

Inspired by tabCNN, Kim et al. in 2022 developed a model for note-level guitar transcription using an attention mechanism with a convolutional neural network [1]. This model captures short-term and long-term features of guitar sounds for accurate transcription, employing beat-informed quantisation for note-level precision. This research aims to analyse and expand this model to assess its strengths, generalization capabilities, and effectiveness in handling diverse datasets.

B. AVAILABLE DATASETS

As detailed by Xi et al. [19] and Riley et al. [34], the availability of note-level annotated datasets for realistic polyphonic guitar pieces is limited with few exceptions.

GuitarSet [19] is one of the most commonly used datasets for guitar transcription, consisting of approximately 3 hours of annotated guitar performances recorded with a hexaphonic pickup that captures individual string outputs. The EGDB dataset [35], provides 2 hours of guitar audio recorded by a professional guitarist using a hexaphonic pickup and direct input (DI) signals, which are then rendered through 6 different amplifier emulation plugins to increase diversity.

The IDMT-SMT-GUITAR database [32] includes recordings from 3 musicians using 6 guitars (5 electric, 1 acoustic), with DI and microphone outputs. It targets different MIR tasks, but only a subset of the audio is annotated with time-aligned note labels, limiting its applicability for transcription tasks.

Lately, Zang et al. [36] proposed a large-scale synthetic dataset derived from the DadaGP dataset [37] for pre-training. While this approach provided additional data, their results on the GuitarSet test split (86.1 F1 without offsets) were lower than models trained only on GuitarSet. This indicates that synthetic data alone does not significantly improve AMT systems.

On the other hand, there are several datasets for polyphonic instrument transcriptions in other domains. The MAPS dataset [5] contains extensive transcriptions of piano notes, chords and pieces recorded under various acoustic conditions. In a similar way, the UMA-Piano dataset [38] covers all possible combinations of notes at different dynamics, and has made a significant contribution to research in piano music transcription. Recent preservation efforts have also expanded piano transcription datasets by digitising player piano rolls [39].

MAESTRO [40] is a dataset of approximately 200 hours of virtuoso piano performances. In particular, the dataset offers finely aligned note labels with audio waveforms, achieving synchronisation accuracy of approximately 3 milliseconds.

The URMP dataset [41] consists of audio, video, and MIDI annotation of multi-instrument musical pieces assembled from coordinated but separately recorded performances of individual tracks.

The Lakh MIDI Dataset [42] is a collection of 176,581 unique MIDI files scraped from publicly available sources on the Internet, spanning multiple genres. The Synthesised Lakh Dataset [43], is a dataset constructed by creating high-quality renderings of 2100 files from Lakh MIDI using professional-quality virtual instruments.

MusicNet [23] consists of 330 recordings of classical music with MIDI annotations. The annotations were aligned to recordings via dynamic time warping, and were then verified by trained musicians.

For guitar, the Guitar Playing Techniques dataset [44] provides 6580 clips annotated with playing techniques for individual notes. The IDMT-SMT-Audio-Effects dataset [45]

provides approximately 20 hours of guitar notes and chords with various audio effects. In addition, the IDMT-SMT-Guitar dataset [32] contains various guitar data such as single notes, playing techniques, note clusters and annotations at both note and chord level for short excerpts.

III. METHODOLOGY

A. DATASETS

In this research, two datasets are used. The first is the GuitarSet dataset [19], which is used for training and is the same dataset referred to in the paper by Kim et al. [1], upon which we base our approach. The second is a novel recorded and labelled dataset, named GM Dataset, used to test the model on more diverse and variable data.

1) GuitarSet

The GuitarSet dataset consists of audio recordings of solo acoustic guitar performances.

This dataset was created using a guitar equipped with magnetic pickups, called hexaphonic pickups, which have individual outputs for each magnet, making it possible to create a frame-level pitch annotation for each of the guitar's 6 strings. In addition to the six channels from the hexaphonic pickup, the guitar was also recorded with a single microphone for a complete audio recording.

Six experienced guitarists were recruited to record for this database. The dataset contains audio recordings of 360 solo guitar performances, each approximately 30 seconds in length. These performances span every key and cover five music genres: Rock, Jazz, Funk, Bossa Nova (BN), and Singer-Songwriter (SS).

The guitarists were instructed to play two different versions of each song: soloing, which contains mostly single notes, and comping, which means playing chords. The six guitarists contributed to the dataset, each performing a provided chord progression, but interpreting it in their own way.

For each recording, the dataset contains annotations for: tempo, key, and style; beats and downbeats; instructed chords; performed chords; Note-level transcription including: string and fret position, onsets offsets pitch contour for each note.

2) GM DATASET

For this article, a small dataset called GM Dataset was recorded and labelled to test the model on a different dataset with different genres and study its generalisation capabilities on different data.

The GM dataset was designed to complement GuitarSet by providing a more diverse and realistic benchmark for evaluating guitar transcription models. While GuitarSet is highly valuable for training in controlled settings, it is based on only three standardized harmonic progressions: 12-Bar Blues, Autumn Leaves, and Pachelbel Canon reused across 30 tracks performed by the six guitarists. This structure, although consistent, introduces redundancy and limits in harmonic and stylistic diversity. In contrast, the GM dataset includes songs from a broad range of genres (some of which are not included in GuitarSet) such as reggae, pop, rock, jazz, blues and classical music, offering greater variety in both harmonic structure and playing techniques. It features 12 unique harmonic progressions, including common pop patterns (I–V–vi–IV), standard jazz cadences (ii–V–I), modal rock loops (i–VI–III–VII), 12-bar blues, and more complex tonal forms (e.g., Tears in Heaven by Eric Clapton, Scar Tissue by Red Hot Chili Peppers). Moreover, it captures advanced guitar techniques like pitch bending (e.g., Johnny B. Goode by Chuck Berry) and palm muting, as well as ambiguous tonal centers and complex arpeggios. The recordings include performances using both pick and fingerpicking techniques, introducing additional variability in articulation and timbre. Additionally, the GM dataset includes iconic tracks from well-known artists, making it more representative of the kinds of music users are likely to engage with. As such, it provides a critical test of how well models trained on the controlled conditions of GuitarSet can generalize to the more complex and varied demands of real-world guitar transcription.

Given the difficulty of replicating the recording conditions of GuitarSet, an alternative strategy was adopted.

The main idea for creating the dataset efficiently was to record the songs using existing transcriptions, adhering as closely as possible to the scores provided. This approach was chosen to avoid the time-consuming process of manually transcribing each song after recording and to overcome the lack of hexaphonic pickups.

For this reason, the first step was to select songs to be recorded. The chosen songs vary in genre, styles and techniques as described above.

Once the songs were chosen, their MIDI files and tablatures were obtained from Ultimate Guitar [46], a website with an extensive collection of songs and their transcriptions. For some songs, it was necessary to adapt the score and correct minor errors. To accomplish this, the Guitar Pro software [47] was used.

A fundamental feature of Guitar Pro, used to create this database, is the ability to export the transcription as a MIDI file.

Some songs were recorded directly using a mobile phone to test the model's effectiveness with lower-quality recordings. Others were recorded using a semi -professional sound card and a Digital Audio Workstation to capture higher-quality audio.

It is important to note that some songs were recorded with an acoustic guitar, while others were recorded with an electric guitar using a clean sound. This was done to test the adaptability of the model to different types of guitars and because certain genres are more suited to a particular type of guitar.

Another crucial step to enhance the transcription's accuracy was data alignment. Alignment between the recording and transcription involves precisely synchronising and

matching the timing and content of a recorded audio signal with its corresponding transcription. This alignment was facilitated using a Digital Audio Workstation (DAW). The MIDI file of the transcription exported from Guitar Pro was imported into the DAW and aligned with the recording.

At this stage, the remaining challenge was to move from having a single MIDI file for each transcription (generated by Guitar Pro) to having six MIDI files, each containing the MIDI transcription of one guitar string. This step was crucial because having such a string-by-string transcription (as in GuitarSet) allows the actual tablature to be derived; in fact, knowing which MIDI note is played on a string allows the fret pressed to be easily determined. By subtracting the MIDI note corresponding to the string from the MIDI note played, the fret pressed can be identified.

This was done using Guitar Pro. The transcription for each song was duplicated six times (once for each string), and in each copy, the notes for all strings except the one in question were deleted and replaced with pauses to maintain the correct timing. This was a time-consuming process, but thanks to some of Guitar Pro's editing features, it was manageable and much quicker than transcribing each string individually.

The final content of the dataset was the following:

- 4 rock recordings totalling 7 minutes and 12 seconds, including guitar riffs, solos, and chord progressions.
- 5 jazz/blues recordings totalling 6 minutes and 40 seconds, including riffs, solos, comping, and arpeggios.
- 5 pop recordings totalling 6 minutes and 58 seconds, including one chord progression and three arpeggios.
- 2 reggae recording with a duration of 1 minute and 57 seconds, including a riff and a chord progression.
- 2 classical recordings totalling 2 minutes and 42 seconds, composed of one arpeggio and one solo.

B. DATA PREPROCESSING

Before training the model, some data preprocessing to both audio and labels (MIDI transcriptions) was conducted to ensure that the data were formatted correctly for input into the model. This preprocessing stage closely follows the approach outlined in TabCNN [25] and it is described in this section.

1) AUDIO PREPROCESSING

During the preprocessing stage, the audio data was downsampled from 44100 Hz to 22050 Hz to reduce the dimensionality of the input signal, assuming that there was not too much relevant information above 11025 Hz. Each audio clip was then normalised to its maximum value to account for possible amplitude variations between clips.

Given that the model incorporates a convolutional stack and a conformer block (architectures designed to learn spatial features from images) the raw audio is converted into an image representation.

Since the task involves recognizing musical pitches, it is advantageous to use a representation where the frequency axis is linearly spaced according to pitch. To achieve this, the constant-Q transform (CQT) is employed, as it effectively reduces the dimensionality of the frequency axis by distributing the frequency bins linearly in accordance with musical pitch.

Motivated by previous work [25], [48], a CQT with 192 bins spanning 8 octaves is used. This equates to 24 bins per octave, or 2 bins per semitone.

The hopsize was set to 512, which corresponds to a frame rate of about 43 frames per second. A hop size of 512 samples is commonly chosen for the CQT because it balances temporal and frequency resolution, facilitates efficient computation, and aligns well with a sampling rate of 22050 Hz.

Since the model processes input in 4-bar segments, the CQT is split into 4-bar sections and then compressed into numpy (.npz) files. In the GuitarSet Dataset, each audio file has a number of bars that is a multiple of 4. However, this is not the case for the recorded dataset. To enable the model to handle audio content of any duration, padding is applied at the end of audio files that do not have a number of bars divisible by 4. For example, if an audio file has 15 bars, the CQT is divided into four segments of equal length, with padding added to the last segment to make it 4 bars, resulting in three full 4-bar segments and one segment padded to reach 4 bars.

2) LABEL PREPROCESSING

As explained in the dataset description, the audio transcriptions are provided as six MIDI files for each recording, one for each string. However, for both training and testing the model, it is necessary to extract additional transcriptions that are more useful for this task. The most important are the ground truth note-level tablature and the ground truth frame-level tablature, which are essential for computing the loss and evaluating the model's performance. Additionally, the note fundamental frequencies (note F0) and frame fundamental frequencies (frame F0) are important for analysing the results and assessing how accurately the model predicts the correct pitch.

To obtain the frame-level tab annotations from the six MIDI files, the start time, duration, and pitch of each note are extracted from each MIDI file. The pitch, which represents the tonal pitch of the note, is then converted to a fret number relative to the string on which it is played. This is done by subtracting the open pitch value of the corresponding string from the pitch of the note. The resulting integer is the fret number, with a value of zero representing an open string (no frets pressed). Since most guitars have 19 frets and strings can also be muted (no sound produced), this gives 21 possible fret classes for each string: open, muted, or any of the 19 frets.

Having determined the fret numbers for all notes on each string, this information is used to construct a tabular representation at the frame-level. The state of each string in a frame is encoded in a one-hot representation, resulting in a label size of 6×21 for each frame.

To obtain the note-level tab annotation, the process is very similar to the frame-level annotation, but with one

key difference. Instead of creating a frame-level matrix, a note-level matrix is created. This involves calculating the duration of each note relative to a fixed note resolution. For example, with a note resolution of 16, each bar is divided into 16 equal parts, and the duration of each note is calculated as a 1/16th bar segment. This note duration is then used to create the note-level matrix, which is initialised and updated based on the presence and position of notes.

To derive fundamental frequency (F0) annotations from MIDI data, two distinct matrices, note_F0 and frame_F0, are used to capture both note-wise and frame-wise representations of pitch activity. Each matrix is designed to accommodate the full range of pitches typically found in musical compositions, adjusted by subtracting 40 from each MIDI note's pitch to align with matrix indexing.

In the note-wise F0 annotation phase, the algorithm iterates through each MIDI string to identify and annotate notes. For each note encountered, its pitch is translated to the appropriate index in the note F0 matrix. During annotation, the algorithm marks the duration of each note by setting corresponding matrix entries to 1, employing integer divisions of the note's start and end times by a predetermined note duration factor that is the same computed for the note-level tab.

In a similar way, in the frame-wise F0 annotation stage, the process repeats with a focus on temporal segmentation. Here, the frame F0 matrix is populated based on the start and end times of each note, scaled by a normalised length factor. This operation ensures that the frame F0 matrix accurately reflects when each pitch is active across consecutive frames.

Similar to the audio preprocessing stage, the annotations are divided into 4-bar segments and then compressed into numpy (.npz) files. To accommodate labels where the number of bars is not a multiple of 4, padding is also applied at this stage.

C. NETWORK ARCHITECTURE

The model structure, as detailed by Kim et al. in their study [1], comprises four primary components: a convolution stack acting as a local feature extractor, conformer blocks for capturing global interactions from features extracted by the convolution stack, a beat-informed quantisation layer designed to accurately quantify latent features based on Beats Per Minute (BPM) information while minimising information loss, and two output layers. These output layers generate frame-level and note-level predictions, facilitating the integration of multi-task learning.

1) CONVOLUTION STACK

The convolution stack consists of several layers designed to effectively process the input features. As shown in Figure 1, the stack starts with two convolution blocks, each of which sequentially integrates 2D convolution, batch normalisation, and Rectified Linear Unit (ReLU) activation function. Latent

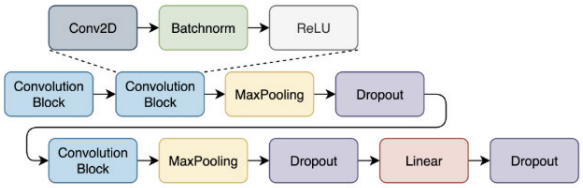

FIGURE 1. Network structure of the convolution stack from [1].

features from these two blocks are then subsampled through a max-pooling layer. This process is repeated with another convolution block and max pooling layer to further refine the features. Finally, a linear layer is applied to reduce dimensionality for output. In addition, three dropout layers are strategically placed after the max pooling and linear layers to mitigate overfitting during training.

2) CONFORMER BLOCK

The Conformer architecture [49], integrates convolution improvements into the Transformer architecture. Within the Conformer block, several critical components contribute to its efficiency:

The dot-product self-attention module incorporates relative positional encoding, enabling the model to capture dependencies between different positions within the sequence effectively.

Furthermore, the convolution module within the Conformer block includes both pointwise and depthwise convolutions. Residual connections are incorporated in both the self-attention and convolution modules, facilitating direct information flow from one layer to the next.

In addition, feedforward modules positioned before and after the self-attention and convolution components use half-step residual connections. These connections help to maintain gradient flow during training, thereby enhancing the model's ability to learn and generalise from data.

For the implementation of the network architecture in Python, the conformer block is realised using the ESPNet2 framework [50]. This framework provides robust capabilities for constructing and integrating the conformer block into the model's structure.

3) BEAT-INFORMED QUANTISATION

Beat-informed quantisation is a non-trainable process designed to convert a frame-level representation of the audio (where the signal is divided into very small time slices) into a note-level sequence that aligns more closely with musical timing. By using tempo information (BPM) and a predefined rhythmic resolution (such as 16th notes), the method ensures that notes are placed in musically meaningful positions. This beat-aware alignment helps reduce small timing inconsistencies that often occur in raw frame-based outputs, leading to transcriptions that are both rhythmically cleaner and more musically interpretable.

The quantisation ratio $K \in \mathbb{R}$ is calculated using the formula:

$$K = \frac{60 \cdot f_s}{q/4 \cdot B \cdot h} \tag{1}$$

where f_s denotes the sampling rate, q represents the tatum, which is the smallest rhythmic unit or subdivision of a beat, used here to specify the minimum quantisation unit in the qth note (in this case is 16 as the note resolution), B indicates the BPM, and h is the hop length.

In an attempt to preserve as much information as possible when performing beat-informed quantisation, the authors of this model introduced a new method. The method is expressed as:

$$\hat{X}(t_n) = \frac{1}{K}\left[(\lceil Kt_n \rceil - Kt_n)X(\lfloor Kt_n \rfloor)\right] + \sum_{t_f=\lceil Kt_n \rceil}^{\lfloor K(t_n+1) \rfloor - 1} X(t_f) + \{K(t_n+1) - \lfloor K(t_n+1) \rfloor\} X(\lfloor K(t_n+1) \rfloor)] \tag{2}$$

where X denotes input latent features generated from the conformer blocks with a framewise sequence length, $\hat{X}$ denotes an output with a notewise sequence length, and t_f and t_n denote framewise and notewise times, respectively.

4) MULTI-TASK LEARNING

In order to obtain probability distributions over each string, a linear layer employing a string-wise softmax function is used within the output layers of the system. The model is trained using multi-task learning, which includes both frame-level and note-level estimation. In addition, guided attention loss [51] is integrated to increase training stability and accelerate convergence. This technique directs the model's attention to relevant aspects of the input or desired output, optimising training efficiency.

The system's loss function (ℓ) is formulated as:

$$\ell_{total} = \ell_{frame} + \ell_{note} + \ell_{att} \tag{3}$$

where ℓ_{frame} represent the frame loss, ℓ_{note} represent the note loss and ℓ_{att} represent the guided attention loss. The frame loss and note loss can be expressed as:

$$\ell_{frame} = \frac{1}{6 \cdot 21 \cdot T} \sum_{s=1}^{6} \sum_{f=1}^{21} \sum_{t=1}^{T} \{y_{s,f,t} \log(\hat{y}_{s,f,t}) + (1 - y_{s,f,t}) \log(1 - \hat{y}_{s,f,t})\} \tag{4}$$

$$\ell_{note} = \frac{1}{6 \cdot 21 \cdot N} \sum_{s=1}^{6} \sum_{f=1}^{21} \sum_{n=1}^{N} \{z_{s,f,n} \log(\hat{z}_{s,f,n}) + (1 - z_{s,f,n}) \log(1 - \hat{z}_{s,f,n})\} \tag{5}$$

where y represents the frame-level ground truth label, obtained from a string-by-string MIDI transcription that indicates the note played on each string over time, with values of $y_{s,f,t} \in \{0, 1\}$ indicating the absence (0) or presence (1) of a note on string s, fret f, at frame t; $\hat{y}$ represents the frame-level prediction from the model. Similarly, z denotes the note-level ground truth label, also derived from the MIDI transcription, with values $z_{s,f,n} \in \{0, 1\}$ indicating the absence (0) or presence (1) of a note on string s, fret f, at note position n; $\hat{z}$ denotes the note-level prediction from the model. T and N denote the framewise and notewise lengths.

The guided attention loss instead can be expressed as:

$$\ell_{att}(A) = \frac{\alpha}{T^2} \sum_{t_1=1}^{T} \sum_{t_2=1}^{T} A(t_1, t_2)[1 - exp\{\frac{-(t_1/T - t_2/T)^2}{2g^2}\}] \tag{6}$$

where A denotes the attention weight matrix, α denotes the scaling coefficient. In addition, t_1 and t_2 denote, respectively the source and target frames, and T denotes the total number of frames. Lastly, g is a hyperparameter for controlling the strength of the effect of the guided attention.

IV. EXPERIMENTAL SETTING AND RESULTS

To provide an unbiased evaluation of the model's performance and to explore its potential for further advancement in the field of automatic guitar transcription, three experiments were conducted. In the first phase, following the experimental configurations defined by Kim et al. in their study [1], the original experimental setup is replicated using the GuitarSet data set. The results are then analysed and compared to the original results.

Afterwards, the performance of the model is evaluated using the new dataset to assess its generalisation capabilities. An insight into the potential strengths of the model and areas for improvement can be gained from a comprehensive examination of these results.

Finally, a new experiment is presented to determine whether increasing the number of attention heads improves the model's ability to capture intricate musical details and patterns.

Through these experiments and analyses, this study aims to provide an unbiased evaluation of the model's performance and its potential for further advancement in the field of automatic guitar transcription.

A. EVALUATION METRICS

The model is evaluated using both tablature estimation metrics, which measure its ability to identify the fret position of each note played, and multi-pitch estimation metrics, which measure its ability to identify the correct pitch regardless of its fret position. This dual approach ensures a comprehensive evaluation of the system's performance.

All metrics are calculated at both frame and note levels. Frame-level evaluation provides a detailed temporal analysis of the system's performance at each frame. Note-level evaluation provides a broader view, focusing on the recognition of complete notes to ensure that the system is accurately transcribing the music as it is played.

This combined evaluation approach ensures a thorough and multi-dimensional assessment, confirming the system's

ability to accurately identify both guitar positions and the actual notes being played.

The metrics used in this paper are precision, recall, and F1-score, calculated at both the note and frame levels, as well as for tablature and multi-pitch estimation. For tablature estimation, the metrics are denoted as P_{tab}, R_{tab}, and $F1_{tab}$. For multi-pitch estimation, the metrics are denoted as P_{pitch}, R_{tab}, and $F1_{tab}$. To distinguish between note-level and frame-level metrics, the former are labeled with the prefix "Note", while the latter are labeled with the prefix "Frame". Another important metric used in this paper is the Tablature Disambiguation Rate (TDR) [25], introduced by Wiggins et al. This metric evaluates tablature estimation by computing the ratio of correctly identified string-fret combinations to the total number of correctly identified pitches. It measures how often the model assigns the correct tablature position to a correctly predicted pitch. As with other metrics,the TDR score is calculated at both the note-level and the frame-level.

B. FIRST EXPERIMENT

In the first experiment, the model was trained and tested under its original conditions to evaluate its reproducibility and verify the results.

1) TRAINING CONDITIONS

To ensure a fair comparison with prior work the training conditions were the same as those detailed by Kim et al. in their work [1]. For a robust evaluation, a sixfold cross-validation approach was employed. In this approach, each guitarist is used as a test subject cyclically, while the remaining five guitarists' data are split into training and validation sets with a ratio of 0.9. This approach is particularly suitable for evaluating performance in a leave-one-subject-out scenario, which reflects real-world cases where a system trained on certain players needs to generalize to others with different playing styles. A step decay learning rate strategy was used, starting with an initial learning rate of 0.005. This learning rate is halved every 32 epochs to allow the model to make larger updates early on and fine-tune more precisely in later stages, reducing the risk of overshooting during optimization. Rectified Adam (RAdam) [52] was used as the optimiser which was chosen due to its improved stability over early stages of training compared to standard Adam, especially in small-batch settings. All network parameters were initialised using the Xavier initialiser [53], which sets the weights to values that maintain the variance of activations through layers, promoting faster convergence. This choice is particularly effective in transformer-based architectures like the Conformer, as it avoids vanishing or exploding gradients, contributing to more stable training dynamics. The conformer architecture was implemented with one conformer block and one attention head. Guided attention loss was used to encourage monotonic alignment between the audio input and output sequence. The parameter $g = 0.4$ controls the width of the diagonal alignment region in the attention matrix, allowing flexibility for natural timing variations while still guiding the model. The loss weight $\alpha = 1$ ensures balanced contribution between the guided attention and the main loss, promoting stable training and better alignment without overpowering the primary objective. The attention mechanism dimension was set to 64, which defines the size of the vector representing each attention head's output. The tatum q, set to the 16th note, aligns the model with standard musical time divisions, ensuring that it can capture fine rhythmic details, such as faster note transitions. The number of epochs was set to 192, providing sufficient training time for the model to converge while preventing overfitting, ensuring a stable and generalized performance across different guitar tracks. Alternative epoch counts were tested, but 192 yielded the most effective balance between performance.

2) ANALYSIS OF THE RESULTS

The results (Table 1) matched closely those reported in the work by Kim et al. [1], with only minor differences within the expected range of variability for experimental replication. Specifically, the model achieved a note-level tablature precision of 0.772 (compared to 0.781 in Kim et al.'s paper [1]), a note-level tablature recall of 0.773 (versus 0.777), a note-level tablature F1 score of 0.768 (versus 0.775), and a TDR score of 0.913 (compared to 0.919). This TDR score indicates that over 91% of correctly identified pitches were assigned the correct fingering.

Table 1 also reports the frame-level results of another state of the art model (TabCNN [25]), which is limited to frame-level predictions. Notably, the proposed note-level approach achieves higher performance compared to TabCNN, even though they are evaluated at different levels.

For note-level multi-pitch estimation, this model achieved a precision of 0.852, a recall of 0.846, and an F1 score of 0.844, indicating strong reliability and stability. These metrics reflect the model's competence in multi-pitch and tablature estimation.

TABLE 1. Comparison of metrics between Kim et al. model (note-level) [1], TabCNN (frame-level) [25] and this study (note-level).

Model	P_{tab}	R_{tab}	$F1_{tab}$	TDR
Kim. et al.[1] (note-level)	0.781 ± 0.031	0.777 ± 0.039	0.775 ± 0.029	0.919 ± 0.021
Proposed approach (note-level)	0.772 ± 0.036	0.773 ± 0.047	0.768 ± 0.034	0.913 ± 0.025
TabCNN[25] (frame-level)	0.809 ± 0.029	0.696 ± 0.061	0.748 ± 0.047	0.899 ± 0.033

These results indicate that the proposed system behaves conservatively in multi-pitch estimation. The higher multi-pitch precision compared to multi-pitch recall suggests that the system is more likely to miss a detection than to report a pitch that is not present in the signal. This behaviour is common for multi-pitch estimation systems, as a pitch can easily be missed if it blends into the overtones of another pitch present simultaneously. It was observed that this often occurs

with pitches an octave apart; the higher pitch is frequently missed, as it may be perceived as merely the overtones of the lower pitch.

C. SECOND EXPERIMENT

In the second experiment, the model was trained on the GuitarSet dataset and then tested on GM Dataset.

The goal of this experiment was to analyse the behaviour of the model on data that have different characteristics from GuitarSet. In fact, in GuitarSet, each player performed similar chord progressions, changing the interpretation but still performing similar musical patterns, introducing certain similarities in the GuitarSet data.

As detailed in the datasets section, the new testing data set (GM Dataset) contains music that was created using a different technique, played on a different guitar, and recorded in a different way. In addition, the test data include different genres of music and chord progressions that are not present in GuitarSet. This variability and difference provide a good basis for evaluating the model's generalisation capabilities and its ability to adapt to new and diverse musical inputs.

1) TRAINING CONDITIONS

For this experiment, the training conditions remained the same as those used in the previous experiment and described in the original paper, with a few minor changes. To test the model on GM Dataset, all six performances from the GuitarSet were used for training. Due to memory constraints resulting from the increased training data, the batch size was reduced from 32 to 16.

2) ANALYSIS OF THE RESULTS

The tablature estimation metrics in Table 2 provide detailed insights into the performance of the model. These results suggest that while the system achieves reasonably high precision, there may be instances where some notes are missed during identification.

As in the previous experiment, the multi-pitch estimation metrics in Table 3 show high precision, recall, and F1 scores, indicating a strong performance that is competitive with state-of-the-art models. In particular, Table 3 presents the note-level multi-pitch estimation results of the proposed approach in comparison with two state-of-the-art approaches: TabCNN [25] and Deep Salience [54].

Comparing the multi-pitch estimation results (Table 2) with the tabs estimation results (Table 3) it is notable that there are significant differences in precision, recall and F1. The higher performance of multi-pitch estimation underlines the fact that the most difficult challenge in this context is tablature estimation.

However, the Tablature Disambiguation Rate (TDR) being this high (0.830) demonstrates the system's ability to associate identified pitches with correct guitar string and fret positions once detected, and this is an encouraging and important result.

TABLE 2. Tabs evaluation metrics results of the proposed approach tested on GM dataset.

Note P_{tab}	Note R_{tab}	Note $F1_{tab}$	Note TDR
0.648	0.595	0.613	0.830

TABLE 3. Multi pitch evaluation metrics results compared with TabCNN [25] and deep salience [54].

Model	P_{pitch}	R_{pitch}	$F1_{pitch}$
Proposed approach tested on GM Dataset (note-level)	0.800	0.710	0.742
TabCNN [25] tested on GuitarSet (frame-level)	0.900	0.764	0.826
Deep Salience [54] tested on GuitarSet (frame-level)	0.778	0.562	0.646

Overall, the evaluation metrics of the model tested on GM Dataset show lower performance compared to the previous experiment, which aligns with expectations. Some differences observed in the metrics can be reasonably attributed to the discrepancy between the trained and tested data.

As described in the datasets section, GuitarSet was recorded using hexaphonic pickups, which capture clean and isolated signals from each string, resulting in extremely high-fidelity recordings. In contrast, the GM dataset was recorded using more accessible and less standardized equipment. While the recordings are generally clear and suitable for transcription, they naturally contain more background ambience, and dynamic variability than those in GuitarSet. Furthermore, the GM Dataset was designed to introduce greater harmonic, rhythmic, and stylistic diversity, infact GM Dataset contains more unique progressions, as well as more expressive performance elements (e.g., pitch bending, palm muting). This increased complexity provides a more challenging and realistic benchmark for evaluating a model trained primarily in controlled environments, which may explain the observed drop in performance.

Observing the results allowed for an analysis of the reasons behind this performance. The analysis revealed that the system performs better when there are fewer notes or chords in the audio content. It was also noted that songs with simpler and more common chords are detected with higher accuracy, while songs with more complex chords present greater challenges for the model. For instance, the system performed exceptionally well on "La Canzone del Sole" by Lucio Battisti, which features three simple chords: A, E, and D. In contrast, the system struggled with "Black Orpheus," a blues song with more complex chords such as Am7, E7(b9), and Bm7(b5), leading to poorer performance. This suggests that the complexity and density of the chord structures impact the model's ability to accurately predict the musical content.

In some cases, the evaluation metrics may overestimate the number of errors due to inconsistencies in the ground truth for certain playing styles. For example, as illustrated in Figure 2, which represents the ground truth and the detected tablature for a guitar arpeggio, when dealing

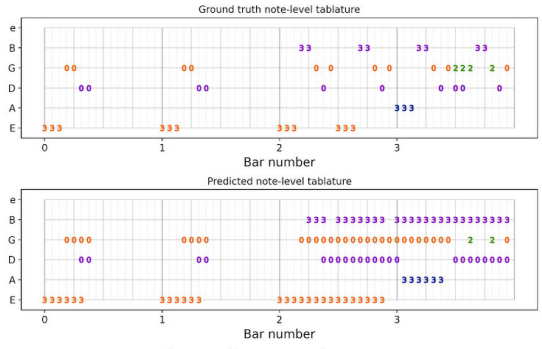

FIGURE 2. Visualisation of a model prediction compared to the ground truth for a guitar arpeggio. The detected notes appear longer than they should; however, due to the continuous ringing of strings in arpeggios, this error is not as significant as it seems.

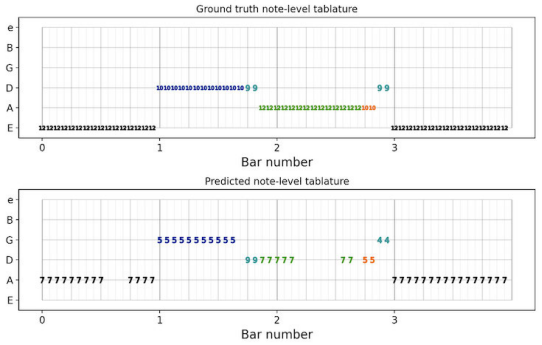

FIGURE 3. Visualization of a model prediction with the corresponding ground truth, illustrating an example missfretting error.

with arpeggios, the ground truth may not always accurately reflect the sustained ringing of the strings, which often results in a sound more similar to a chord. This discrepancy can lead to significant differences in note duration between the system's predictions and the ground truth, even if the predictions are theoretically correct.As a result, the metrics may indicate a higher error rate than what is actually present in the music. In the figure, the y-axis corresponds to the guitar strings, the x-axis represents the measure bars and the numbers indicate the fret position.

It is also important to analyse missfreting errors that occur when a pitch is correctly detected, but the wrong string-fret combination is assigned. This error has a negative impact on both tablature precision and tablature recall. The TDR metric directly indicates how often this type of error is avoided. Figure 3 shows the predicted tablature alongside the ground truth, where the numbers indicate the frets and the colors represent the pitch of the notes (notes with the same color correspond to the same pitch). This visualization highlights the model's errors: for example, when two notes share the same color but appear on different strings or frets, it indicates that the model predicted the correct pitch but assigned it to the wrong position on the fretboard.

It was interesting to observe that, most of the time, even when these errors occurred, the predicted note positions were often close to each other (as in Figure 3), resulting in a tablature that remained playable. This reduced the impact of the errors and highlighted the importance of the self-attention mechanism, which considers surrounding note positions to create more coherent and realistic transcriptions.

D. THIRD EXPERIMENT

Given that the previous experiments emphasised the importance and effectiveness of the self-attention block, this third experiment was designed to assess the effect of integrating an additional attention head within the conformer block of the model. The aim of the study was to investigate whether this addition could enhance the model's ability to capture both local and global dependencies in the musical sequences it transcribed, thereby improving its generalisation capabilities.

1) TRAINING CONDITIONS

For this experiment, the model was trained on GuitarSet and then evaluated on the recorded dataset, as in the second experiment. Most of the training conditions remained the same as in the previous experiments, with the exception of configuring the model with two attention heads. Due to the increased complexity of the model, the batch size had to be further reduced to 8 due to memory constraints.

Initially, it was considered to train the model under the same experimental conditions as the first experiment, using a sixfold cross-validation approach on GuitarSet, with each guitarist acting cyclically as a test-set. However, due to the significant memory requirements (the model would need to be trained five times) it was decided to test the model only on the new dataset. This decision was also driven by the fact that the primary focus of this study was to improve model performance on diverse new data.

E. ANALYSIS OF THE RESULTS

Table 4 presents the metric results comparing the two-attention-head model configurations to those with a single attention head, providing a comprehensive comparison of their respective performances.

The comparison between models with one and two attention heads provides a detailed assessment of their performance in tablature and multi-pitch estimation tasks.

For tablature estimation, the increase to two attention heads resulted in a modest improvement in recall, indicating better coverage in identifying tablature notes. However, this improvement was accompanied by a decrease in precision and F1 score, suggesting that while more notes were correctly identified, there was also an increase in false positives.

The models also behaved similarly for the multi-pitch estimation, with the two attention heads model showing a slight increase in recall but a decrease in precision, resulting in a lower F1 score. In this case, too, there was an increase in pitch recognition, but there was also an increase in the number of incorrect identifications.

These results suggest that increasing the number of attention heads in the model had both positive and negative effects on its performance. In particular, the addition of a

TABLE 4. Comparison of evaluation metrics results for multi-pitch and tablature estimation between the model with two attention heads and the model with one.

N° Attention Heads	Note P_{tab}	Note R_{tab}	Note $F1_{tab}$	Note TDR
1	0.648	0.595	0.613	0.830
2	0.602	0.605	0.593	0.819

N° Attention Heads	Note P_{pitch}	Note R_{pitch}	Note $F1_{pitch}$
1	0.800	0.710	0.742
2	0.755	0.730	0.728

second attention head generally improved the model's recall, suggesting that the model with two attention heads was able to capture more complex patterns in the data, recognising more tablature notes and multi-pitch elements.

However, alongside these gains in recall, the decrease in precision suggests that the model produced more false positives, indicating an increase in noise and less accurate output in many cases.

In addition, the TDR score also decreased, although not significantly, indicating a slight overall decline in the model's performance.

It is interesting to analyse the attention maps generated alongside the model outputs to gain deeper insights into the results and better understand why performance decreased.

The attention maps show how much attention the model assigns to different time frames in the input data when making predictions. By examining these maps, it is possible to visualise which features or elements in the input are being attended to more strongly by the model. This analysis helps in understanding how the attention mechanism influences the model's decision-making process.

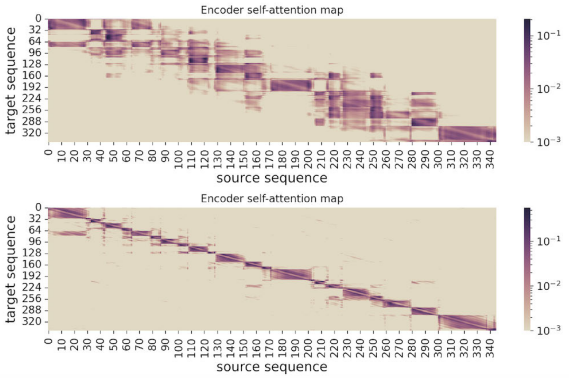

FIGURE 4. Two attention maps generated by the model for the song 'Redemption Song' by Bob Marley.

Figure 4 shows the two attention maps of a model output, where it is interesting to notice the two maps have very different behaviour. The top attention map, associated with the first attention head, shows a more dispersed pattern across the source sequence. When generating output, it does not just focus on the exact corresponding time frame but also on more distant frames. This indicates a broader attention span, highlighting global dependencies.

The bottom attention map, associated with the second attention head, exhibits instead a distinct diagonal pattern. This pattern suggests a focus on local dependencies, where each target position primarily attends to its corresponding source position and nearby elements. This localised focus enables the model to capture the immediate context and detailed nuances within the musical sequence. Similar behaviour was evident in most of the outputs of the model, indicating a common pattern.

There are several reasons why the model did not improve its performance with an additional attention head. Increasing the number of attention heads increases the complexity of the model which can affect model performance, especially if the training data lacks diversity.

Also, the second attention head focused on local time frames and did not capture global dependencies, thus degrading performance without providing additional information. Furthermore, some of the local dependencies it identified may have already been learned by the other attention head, introducing redundancy. In fact, some patterns in the bottom map recall patterns from the other attention map.

V. CONCLUSION AND FUTURE WORK

In this research, a promising model for note-level automatic guitar transcription was explored and evaluated. The study comprised three key experiments, each contributing valuable insights into the model's strengths, limitations, and potential areas for improvement.

The first experiment focused on analysing the model and understanding its characteristics. It was found that the self-attention mechanism plays a crucial role in the model's ability to capture musical nuances and dependencies. The model is very good at predicting string and fret positions that are close to one another, ensuring that the resulting transcriptions are playable on the guitar.

The results obtained from the replicated model are very close to those of the original paper. The differences are small and fall within the expected margins of variability in an experimental replication context.

The second experiment introduced modifications to the data preprocessing stage to handle songs of varying duration using a padding technique and adapting the model to audio data different from the GuitarSet dataset [19]. This adaptation was necessary to evaluate the model's robustness and generalisability across diverse musical contexts.

The results highlighted the importance of incorporating new and diverse data to fully understand the model's capabilities. Although there was a performance degradation when the model was tested on data different from the training set, the model still performed quite well.

The model exhibited higher precision than recall, suggesting that it is more likely to miss certain notes rather than incorrectly identifying notes that are not present. This finding underscores the model's conservative approach, prioritising accuracy over completeness, which is essential in reducing false positives in music transcription tasks.

This experiment revealed that the model excels in transcribing chords, particularly common chords, but faces challenges with solo transcriptions.

A critical insight from this experiment was the limitation of the model due to the low diversity in the training data. To overcome this, it is essential to integrate the training dataset with more diverse and extensive data. For example, the newly introduced dataset, if expanded, could significantly improve the training process. In addition, the use of data augmentation techniques like pitch shifting, time stretching or adding noise could further diversify the training data, thereby improving the robustness and performance of the model.

The experiment also identified a few problems with the model's handling of overtones, suggesting that further research in this area could lead to significant improvements. It was also observed that errors often occur at the beginning or end of a note. One strategy for improving estimation performance could be to incorporate a temporal smoothing algorithm.

The third experiment examined the effect of increasing the number of attention heads on the model's performance. Contrary to expectations, increasing the number of attention heads did not necessarily lead to performance gains. Instead, it introduced additional complexity that did not translate into better results. This finding suggests that simply adding one more attention heads is not a guaranteed path to improvement, as the added complexity may outweigh the potential benefits. However, future experiments exploring architectural variations such as increasing the number of layers or using three to four attention heads could provide valuable insights. It would be particularly interesting to assess whether adding more heads beyond two leads to diminishing returns or even degrades performance due to increased model complexity.

This experiment highlighted the need to focus on other aspects of model improvement rather than increasing the number of attention heads. In particular, improving the quality and diversity of the training data seems to be a more promising approach to achieving significant performance gains.

Another significant improvement to the model could be the integration of a BPM estimation component, rather than relying on beat-informed quantisation. This improvement would eliminate the need for pre-existing BPM information, allowing the model to determine tempo independently, thereby streamlining the transcription process.

The creation of the new dataset proved to be extremely useful and important. This dataset should be integrated in order to have even more diverse and complete data. This study has also shown an efficient working strategy to record and label the data. This strategy proved to be effective and also more accessible than the strategy used to record the guitar set. In terms of the data set, one adjustment that could be made is to adjust the ground truth for styles such as arpeggios to reflect the sustained ringing of the strings.

In conclusion, this article has demonstrated the significant potential of advanced neural network models for automatic guitar transcription. These developments not only improve the accuracy and usability of transcription models, but also play a crucial role in facilitating music learning and the sharing of musical ideas between people. By making music transcription more accessible and accurate, these models can empower musicians, educators and enthusiasts to explore and understand music more effectively.

REFERENCES

[1] S. Kim, T. Hayashi, and T. Toda, "Note-level automatic guitar transcription using attention mechanism," in *Proc. 30th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2022, pp. 229–233.

[2] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 20–30, Jan. 2019.

[3] P. Bhagwat, V. Shelke, A. Murugkar, K. Dakwala, and D. S. C. Dharmadhikari, "A survey on automatic music transcription," Tech. Rep., 2023.

[4] L. Su and Y.-H. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 10, pp. 1600–1612, Oct. 2015.

[5] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1643–1654, Aug. 2010.

[6] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio*, vol. 18, no. 8, pp. 2121–2133, Feb. 2010.

[7] P. H. Peeling, A. T. Cemgil, and S. J. Godsill, "Generative spectrogram factorization models for polyphonic piano transcription," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 519–527, Mar. 2010.

[8] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2003, pp. 177–180.

[9] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 528–537, Mar. 2010.

[10] E. Benetos and S. Dixon, "Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model," *J. Acoust. Soc. Amer.*, vol. 133, no. 3, pp. 1727–1741, Mar. 2013.

[11] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 927–939, May 2016.

[12] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," 2016, *arXiv:1612.05153*.

[13] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "MT3: Multi-task multitrack music transcription," 2021, *arXiv:2111.03017*.

[14] 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," 2017, *arXiv:1710.11153*.

[15] R. Kelz, S. Böck, and G. Widmer, "Deep polyphonic ADSR piano note transcription," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 246–250.

[16] 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 Trans. Audio, Speech, Language Process.*, vol. 29, no. 12, pp. 3707–3717, Dec. 2021.

[17] M. Cartwright and J. P. Bello, "Increasing drum transcription vocabulary using data synthesis," in *Proc. Int. Conf. Digit. Audio Effects (DAFx)*, Jan. 2018, pp. 72–79.

[18] L. Callender, C. Hawthorne, and J. Engel, "Improving perceptual quality of drum transcription with the expanded groove MIDI dataset," 2020, *arXiv:2004.00188*.

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

[20] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 771–775.
[21] L. Lin, Q. Kong, J. Jiang, and G. Xia, "A unified model for zero-shot music source separation, transcription and synthesis," 2021, *arXiv:2108.03456*.
[22] K. W. Cheuk, D. Herremans, and L. Su, "ReconVAT: A semi-supervised automatic music transcription framework for low-resource real-world data," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3918–3926.
[23] J. Thickstun, Z. Harchaoui, and S. Kakade, "Learning features of music from scratch," 2016, *arXiv:1611.09827*.
[24] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," 2021, *arXiv:2107.09142*.
[25] A. Wiggins and Y. E. Kim, "Guitar tablature estimation with a convolutional neural network," in *Proc. ISMIR*, Nov. 2019, pp. 284–291.
[26] G. Burlet and I. Fujinaga, "Robotaba guitar tablature transcription framework," in *Proc. ISMIR*, Jan. 2013, pp. 517–522.
[27] R. Zhou and J. D. Reiss, "A real-time polyphonic music transcription system," in *Proc. 4th Music Inf. Retr. Eval. Exchange (MIREX)*, Jan. 2008.
[28] G. Burlet and A. Hindle, "Isolated guitar transcription using a deep belief network," *PeerJ Comput. Sci.*, vol. 3, p. 109, Mar. 2017.
[29] K. Yazawa, D. Sakaue, K. Nagira, K. Itoyama, and H. G. Okuno, "Audio-based guitar tablature transcription using multipitch analysis and playability constraints," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 196–200.
[30] K. Yazawa, K. Itoyama, and H. G. Okuno, "Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 3122–3126.
[31] A. M. Barbancho, A. Klapuri, L. J. Tardón, and I. Barbancho, "Automatic transcription of guitar chords and fingering from audio," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 915–921, Mar. 2012.
[32] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller, "Automatic tablature transcription of electric guitar recordings by estimation of Score- and instrument-related parameters," in *Proc. Int. Conf. Digit. Audio Effects (DAFx)*, Sep. 2014, pp. 219–226.
[33] B. Fuentes, R. Badeau, and G. Richard, "Blind harmonic adaptive decomposition applied to supervised source separation," in *Proc. 20th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2012, pp. 2654–2658.
[34] X. Riley, Z. Guo, D. Edwards, and S. Dixon, "GAPS: A large and diverse classical guitar dataset and benchmark transcription model," 2024, *arXiv:2408.08653*.
[35] Y.-H. Chen, W.-Y. Hsiao, T.-K. Hsieh, J.-S.-R. Jang, and Y.-H. Yang, "Towards automatic transcription of polyphonic electric guitar music: A new dataset and a multi-loss transformer model," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 786–790.
[36] Y. Zang, Y. Zhong, F. Cwitkowitz, and Z. Duan, "SynthTab: Leveraging synthesized data for guitar tablature transcription," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2024, pp. 1286–1290.
[37] P. Sarmento, A. Kumar, C. Carr, Z. Zukowski, M. Barthet, and Y.-H. Yang, "DadaGP: A dataset of tokenized GuitarPro songs for sequence models," 2021, *arXiv:2107.14653*.
[38] A. M. Barbancho, I. Barbancho, L. J. Tardón, and E. Molina, *Database of Piano Chords: An Engineering View of Harmony*. Cham, Switzerland: Springer, 2013.
[39] Z. Shi, K. Arul, and J. O. Smith, "Modeling and digitizing reproducing piano rolls," in *Proc. ISMIR*, Jan. 2017, pp. 197–203.
[40] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z.-A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," 2018, *arXiv:1810.12247*.
[41] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 522–535, Feb. 2019.
[42] C. Raffel, "The lakh midi dataset v0. 1," Tech. Rep., 2016.
[43] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity," in *Proc. IEEE Workshop Appl. Signal Process. to Audio Acoust. (WASPAA)*, Oct. 2019, pp. 45–49.
[44] L. Su, L.-F. Yu, and Y. Yang, "Sparse cepstral, phase codes for guitar playing technique classification," in *Proc. ISMIR*, Jan. 2014, pp. 9–14.
[45] M. Stein, J. Abeßer, C. Dittmar, and G. Schuller, "Automatic detection of audio effects in guitar and bass recordings," in *Proc. Audio Eng. Soc. Conv.*, May 2010, pp. 1–21.
[46] *Ultimate Guitar*, UltimateGuitar, San Francisco, CA, USA, 2015.
[47] *Guitarpro*, Arobas Music, Lille, France.
[48] E. J. Humphrey and J. P. Bello, "From music audio to chord tablature: Teaching deep convolutional networks toplay guitar," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 6974–6978.
[49] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020, *arXiv:2005.08100*.
[50] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent developments on espnet toolkit boosted by conformer," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5874–5878.
[51] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4784–4788.
[52] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," 2019, *arXiv:1908.03265*.
[53] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Mar. 2010, pp. 249–256.
[54] R. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 estimation in polyphonic Music," in *Proc. ISMIR*, Jan. 2017, pp. 63–70.

SIMONE CHIEPPA received the bachelor's degree in statistics and the master's degree in data science from Sapienza University of Rome, in 2021 and 2024, respectively. During his master's studies, he developed a strong interest in music information retrieval and completed his thesis on automatic music transcription. Currently, he is in the field of deep learning applied to audio signals, with research interests spanning machine learning, deep learning, and their applications to audio analysis.

PIERPAOLO BRUTTI is currently an Associate Professor with the Department of Statistical Sciences, Sapienza University of Rome. He specializes in statistical methodologies with applications in various fields, including topological data analysis, Bayesian statistics, biostatistics, and data science. His academic contributions extend to areas, such as predictive modeling, nonparametric statistics, and the integration of machine learning approaches for complex datasets.

RUI PEDRO PAIVA received the bachelor's, master's, and Ph.D. degrees from the Department of Informatics Engineering, University of Coimbra, in 1996, 1999, and 2007, respectively. He is currently a Professor with the Department of Informatics Engineering, University of Coimbra. He is also a member with the CMS Group, CISUC. His main research interests include MIR and health informatics. His common research hat is the study of feature engineering, machine learning, and signal processing to analyze musical and bio signals.