ChordFormer: A Conformer-Based Architecture for Large-Vocabulary Audio Chord Recognition

Muhammad Waseem Akram, Stefano Dettori, Valentina Colla, *Fellow, IEEE*, and Giorgio Carlo Buttazzo, *Fellow, IEEE*

***Abstract*—Chord recognition serves as a critical task in music information retrieval due to the abstract and descriptive nature of chords in music analysis. While audio chord recognition systems have achieved significant accuracy for small vocabularies (e.g., major/minor chords), large-vocabulary chord recognition remains a challenging problem. This complexity also arises from the inherent long-tail distribution of chords, where rare chord types are underrepresented in most datasets, leading to insufficient training samples. Effective chord recognition requires leveraging contextual information from audio sequences, yet existing models, such as combinations of convolutional neural networks, bidirectional long short-term memory networks, and bidirectional transformers, face limitations in capturing long-term dependencies and exhibit suboptimal performance on large-vocabulary chord recognition tasks. This work proposes ChordFormer, a novel conformer-based architecture designed to tackle structural chord recognition (e.g., triads, bass, sevenths) for large vocabularies. ChordFormer leverages conformer blocks that integrate convolutional neural networks with transformers, thus enabling the model to capture both local patterns and global dependencies effectively. Experimental results show that ChordFormer achieves a 2.3% improvement in frame-wise accuracy and a 6% increase in class-wise accuracy on large-vocabulary chord datasets. Moreover, our evaluation demonstrates that ChordFormer handles class imbalance more effectively than existing models, achieving balanced recognition across a wide range of chord types. We further assess the ChordFormer effectiveness through detailed ablation studies and additional metrics, including ARWCSR and chord inversion matrices. This approach bridges the gap between theoretical music knowledge and practical applications, advancing the field of large-vocabulary chord recognition.**

***Index Terms*—Automatic chord estimation, deep learning, music information processing, machine leanrning.**

I. Introduction

AUTOMATIC chord recognition (ACR) forms the foundation of music information retrieval, acting as a key tool for understanding and analyzing the harmonic structure of music. ACR systems aim at decoding audio recordings into sequences of time-synchronized chord labels, enabling various applications such as automatic lead-sheet creation [1], music structure analysis [2], key classification [3], and cover song identification [4]. These systems not only assist musicians, composers, and educators but also facilitate a deeper understanding of musical content, aiding in the preservation and exploration of diverse musical traditions.

Despite significant progress, ACR remains a challenging task due to the inherent complexities of musical chords, their temporal structures [5], and the uneven distribution of chord types in real-world music datasets [6], [7]. Large-vocabulary chord recognition is particularly difficult: extended chords (e.g., ninths, elevenths, thirteenths) often share overlapping chromatic content, making them harder to distinguish, while common chord types dominate datasets, leading to model bias toward frequent classes.

Another major challenge lies in modeling harmonic context over time. In fact, chords are more than just combinations of notes [8], since they carry rich musical semantics following well-defined harmonic progressions that exhibit long-term dependencies. Modeling such dependencies is a challenging task [5], but it is crucial for accurate chord recognition.

Traditional statistical and neural architectures, such as Hidden Markov Models (HMMs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs), have shown effectiveness in capturing local or short-term context but are less effective for long-range dependencies. Such limitations have encouraged the exploration of attention mechanisms and transformer-based architectures, which have revolutionized sequence modeling in fields like natural language processing [9], [10], speech recognition [11], and music generation [12]. In the context of ACR, such capabilities hold promise for addressing long-term chord dependencies, a challenge that has historically limited the performance of many systems [6], [13], [14], [15], [16], [17], [18], [19], [20].

To tackle these challenges, this paper presents ChordFormer, a conformer-based [11] model for structural chord recognition. The conformer architecture combines convolutional layers with self-attention mechanisms and is particularly well-suited for this task, as it effectively balances the modeling of local and global dependencies.

ChordFormer incorporates a reweighted loss function [6], mitigates class imbalance by assigning higher importance to rare chord types. Furthermore, the model employs a structured chord label representation [6] that encodes musically meaningful

Received 20 February 2025; revised 14 August 2025, 4 October 2025, and 10 December 2025; accepted 11 December 2025. Date of publication 23 December 2025; date of current version 14 January 2026. This work was supported in part by the Italian Ministry of University Research (MUR) in the context of the doctoral scholarship for cultural heritage, and in part by the European Union - Next Generation EU and implemented with the Decree of the MUR n. 351 of 09 April 2022 and subsequent amendments. The associate editor coordinating the review of this article and approving it for publication was Prof. Juhan Nam. *(Corresponding author: Muhammad Waseem Akram.)*

The authors are with the Department of Excellence in Robotics and AI, Sant'Anna School of Advanced Studies, 56127 Pisa, Italy (e-mail: muhammadwaseem.akram@santannapisa.it; stefano.dettori@santannapisa.it; valentina.colla@santannapisa.it; giorgio.buttazzo@santannapisa.it).

Digital Object Identifier 10.1109/TASLPRO.2025.3646468

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

components (root, triad, bass, and extensions), improving interpretability and enabling finer discrimination between similar chords.

A. Paper Contributions

This work presents the following novel contributions:

1) A conformer-based architecture is proposed for large-vocabulary structural chord recognition, effectively capturing both local harmonic patterns and long-term dependencies in musical sequences.
2) A comprehensive ablation studies is carried out to systematically evaluate the impact of key architectural components, including convolutional layers, attention mechanisms, and positional encoding strategies, to understand their role in the ChordFormer performance.
3) An extended evaluation metrics (including ARWCSR) is proposed to assess the performance of ChordFormer also in terms of chord inversion sensitivity, providing deeper insights into recognition quality.

The effectiveness of the proposed approach has been evaluated by conducting extensive experiments on a chord recognition dataset comprising 1,217 songs compiled by Humphrey and Bello [16], [21]. The achieved results demonstrate that the ChordFormer model outperforms state-of-the-art methods, including those employing large-vocabulary transcription and chord structure decomposition. It exhibits superior performance across various metrics, with notable improvements in recognizing rare and complex chord types.

The remainder of this paper is organized as follows. Section II reviews related work in chord recognition. Section III details the design and implementation of the ChordFormer model, including its reweighted loss function and structured label representation. Section IV presents the experimental setup and evaluation metrics, while the results and detailed ablation study are discussed in Sections V, VI. Finally, Section VII concludes the paper and outlines some directions for future research in this domain.

II. Related Work

Automatic Chord Recognition (ACR) has advanced substantially since Fujishima's seminal 1999 work [22], which introduced a real-time system using 12-dimensional chroma features and Hidden Markov Models (HMMs) to model chord sequences. This shift laid the groundwork for transitioning from handcrafted feature-based methods to data-driven paradigms enabled by machine learning and, later, deep learning [23].

Early ACR approaches concentrated on feature extraction and chord sequence decoding. Representations such as chroma vectors and tonal centroids, including the Tonnetz [24], effectively captured harmonic content. Probabilistic models, notably Gaussian Mixture Models (GMMs) [25] and HMMs [26], [27], [28], provided the backbone for sequence decoding. While incremental refinements such as higher-order HMMs [29], [30] partially addressed these models' limitations, performance remained constrained by the reliance on manually designed features.

The advent of deep learning marked a paradigm shift in ACR [13], [31], [32]. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [6], [14], [16], [17], [18], [19], [20] emerged as dominant architectures, with CNNs capturing local harmonic structures and RNNs modeling temporal dependencies across sequences.

Hybrid models [15], [33] that integrate CNNs with Conditional Random Fields or RNNs have achieved notable gains by combining learned auditory features with sequence modeling. Fully convolutional architectures and Bidirectional LSTMs [6], [17] further demonstrate deep learning's capacity to handle ACR's complexities. Reported improvements, however, vary with chord vocabulary, feature representation, and evaluation metrics, indicating that hybridization benefits are often task-specific rather than universally superior.

Attention mechanisms and Transformer-based architectures [9] have been explored to overcome CNNs limitations in modeling long-range dependencies. Self-attention layers enable Transformers to capture harmonic context across entire sequences without recurrence [11]. Yet, their impact in ACR remains mixed: the Bi-directional Transformer [34] performed competitively, while the CNN method of Korzeniowski and Widmer [15] slightly outperformed it. Similarly, O'Hanlon and Sandler [35] reported that the feed-forward network proposed by Korzeniowski and Widmer [31] matched CNN results for the majmin task. In another study [36], the same authors found that HMMs can surpass RNNs [16], [31] for the majmin task. These findings suggest that the advantage of complex temporal models over simpler alternatives is context-dependent rather than absolute.

ACR systems must contend with large vocabularies and class imbalance [5], particularly the scarcity of rare chord qualities. Structured chord representations address these issues by encoding musically meaningful components such as root notes, bass positions, and chord qualities in a manner consistent with music theory, thereby reducing learning complexity [16]. For example, the HPA system [29] employs latent variables for root-position form and bass note within an HMM framework. Likewise, McFee and Bello [16] introduced a 36-dimensional binary encoding of root, bass, and chord degrees, enabling improved modeling of harmonic relationships. Hierarchical and decompositional schemes further enhance generalization to rare and extended chord types.

Recent advances have tackled class imbalance and enhanced model robustness. Jiang et al. [6] introduced a reweighting loss that assigns greater importance to underrepresented chord components, reducing learning bias and promoting balanced recognition across classes. Semi-supervised approaches and variational autoencoders [37] integrate generative and discriminative methods to exploit unlabeled data effectively. Curriculum learning [7] has also been applied to gradually introduce rare chords, leveraging hierarchical relationships between base and extended qualities to improve classification accuracy.

Contrastive learning frameworks [38], coupled with noisy-student models, have advanced the use of unlabeled data to improve recognition of both frequent and rare chord classes. By combining robust representation extraction via contrastive

learning with pseudo-labeling and data augmentation, these methods expand training sets and reduce annotation biases. Evaluations on benchmark datasets, including the Billboard and Humphrey–Bello collections [16], [21], demonstrate their effectiveness in tackling large-vocabulary challenges while maintaining balanced classification.

Building on these foundations, our ChordFormer architecture leverages conformer-based modeling alongside structured chord representations and the reweighted loss of [6] to address the twin challenges of long-term harmonic modeling and class imbalance in large-vocabulary chord recognition.

III. Methodology

This section presents the proposed ChordFormer architecture for large-vocabulary chord recognition. The methodology integrates several components, including a structured representation of chord labels, pre-processing, feature extraction using conformer blocks, a decoding model based on Conditional Random Fields, and a class re-weighting mechanism to address imbalance issues. Each component is detailed in the following subsections.

A. Chord Labels Structured Representation for Recognition

An important insight from the literature [6], [16], [29] is that the effectiveness of models in chord classification tasks heavily depends on the chord representation employed in terms of embeddings. A good chord representation not only enhances classification performance but can also help in addressing the challenge of class imbalance in large chord vocabularies. Representing a chord by its meaningful components, such as root, triad, bass, and extensions (7^{th}, 9^{th}, 11^{th}, and 13^{th}), captures the hierarchical and semantic relationships among the notes that compose complex chords. It further accommodates all possible inversions through bass identification. Such a representation not only mitigates the complexity associated with large chord vocabularies, but also standardizes component labels for a precise and interpretable recognition.

The proposed chord representation uses the following labels:

- *Root + triad* combination: (e.g., D♭ min, Gaug, etc.) where root ∈{N, C, C#/D♭,..., B} and triad ∈{N, maj, min, sus4, sus2, dim, aug}
- *Bass:* {N, C, C#/D♭,..., B}
- *7^{th}:* {N, 7, ♭7, ♭♭7}
- *9^{th}:* {N, 9, #9, ♭9}
- *11^{th}:* {N, 11, #11}
- *13^{th}:* {N, 13, ♭13}

In the notation presented above, it is worth observing that:

- The "N" class included in each component denotes the absence of that component. For example, bass is allowed to be "N" as it may be undefined. This happens, for example, when it is not possible to identify a defined tonal or harmonic centre in an audio section and therefore no chord (and no root and bass) can be associated with it (e.g. sections without harmonic accompaniment, noise, speech, atonal sections).
- The double flat seventh (♭♭7) denotes the sixth and it is used to maintain a theoretical consistency with certain notations encountered in complex modulations.
- The sixth and thirteenth extensions refer to the same absolute note, but their naming depends on the harmonic context and chord function: while a sixth chord refers to a major triad with an added sixth (e.g., C6 = C-E-G-A) without a seventh, a thirteenth chord normally includes the dominant seventh (♭7) and the ninth (9), extending beyond the octave (e.g., C13 = C-E-G-B♭-D-A).

The proposed ChordFormer represents chords using a six-dimensional vector, with each dimension corresponding to a specific chord component. In particular, the first dimension encodes the combination of the root and triad types (e.g., C# augmented, D minor). The second to sixth dimensions represent the bass (capturing inversions), 7^{th}, 9^{th}, 11^{th}, and 13^{th} extensions, respectively. Each component label in the output is encoded using a one-hot representation, enabling the model to focus on specific chord components. This structured approach, proposed by Jiang et al. [6], has been selected for improving the interpretability of the classification output and facilitating precise recognition of chords, even in scenarios involving rare extensions or complex inversions. This approach also allows a more consistent comparison of the results with respect to the current state-of-the-art methods.

B. ChordFormer Architecture Overview

The ChordFormer architecture integrates three main components: a pre-processing module, a conformer block, and a decoding model. They work together to achieve efficient chord recognition. The pre-processing module converts audio into a spectrogram computed by the Constant Q Transform (CQT) [39], which is more suitable for music, since it provides a time-frequency representation where frequency bins are logarithmically spaced, having higher resolution for lower frequencies and lower resolution for higher frequencies. The CQT spectrogram is then processed by the conformer block that produces frame-wise activations that capture chord component values. By leveraging contextual frames, the conformer block enable advanced feature extraction which enhances the accuracy of the predictions. Finally, the decoding model interprets these activations to generate the final chord sequence. An overview of the ChordFormer architecture is illustrated in Fig. 1.

C. Feature Extraction Using Conformer Blocks

The Conformer block, introduced by Gulati et al. [11], has become a state-of-the-art architecture for speech recognition. Recently, ChordSync [40], a Conformer-based model, has shown remarkable results compared to state-of-the-art methods in aligning chord annotations with audio. Conformer block is a modified version of the Transformer design as in [41], [42], which enables advanced feature extraction. It combines Convolutional Neural Networks (CNNs) for capturing local patterns and self-attention mechanisms for long-range dependencies, making it highly effective for sequence modeling tasks. A Conformer block includes a feed-forward module, a self-attention

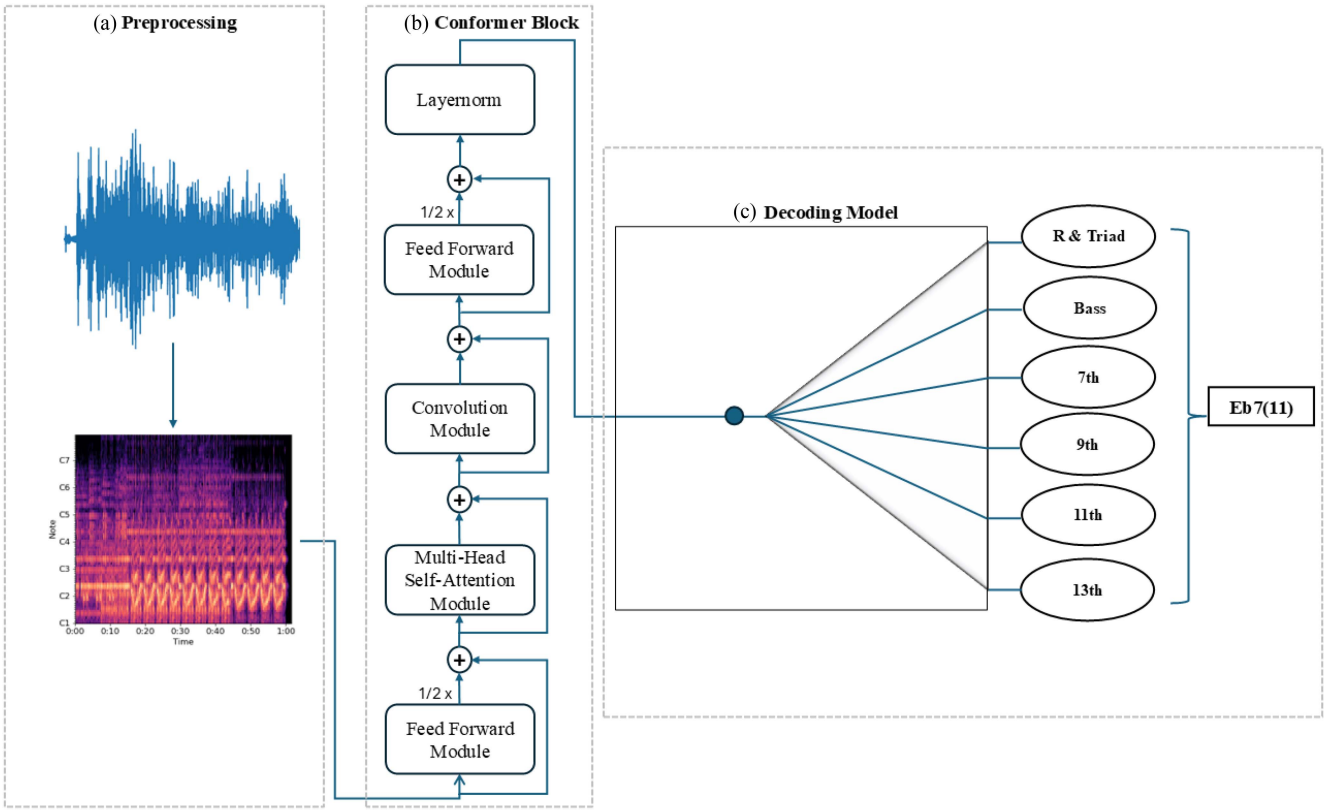

Fig. 1. Overview of the ChordFormer Architecture. The ChordFormer architecture consists of three primary components: (a) Preprocessing Module: Converts audio signals into Constant-Q Transform (CQT) representations to extract low-level time-frequency features essential for chord recognition. (b) Conformer Block: Processes the CQT features to generate frame-wise activations using a combination of convolutional layers and self-attention mechanisms. This block leverages contextual frames to enhance accuracy in chord prediction. (c) Decoding Model: Interprets the activations from the Conformer Block and generates the final chord sequence by decoding the structural chord components.

module, a convolution module, and a final feed-forward module. Sections III-C1, III-C2, and III-C3 introduce the self-attention, convolution, and feed-forward modules, respectively. Lastly, the process of combining these sub-blocks is explained in Section II-I-C4.

1) Multi-Headed Self-Attention (MHSA): Accurate chord recognition requires modeling both short-term context from neighboring frames and long-range dependencies across the musical piece. Traditional architectures, such as CNNs and Recurrent Neural Networks (RNNs), have been effective in capturing local and sequential patterns, respectively. However, they have inherent limitations. CNNs rely on fixed-size receptive fields, which restrict their ability to model relationships beyond a limited temporal range. RNNs are not capable of modeling longer sequences.

In contrast, self-attention mechanisms enable each element to directly attend to all others in the sequence, regardless of their distance. This allows the model to capture musically relevant patterns over varying time spans through an adaptive receptive field. Moreover, unlike CNNs and RNNs, self-attention provides direct, parallel access to global context, improving the model's ability to represent complex harmonic structures that depend on long-range relationships. These properties make self-attention particularly well-suited for large-vocabulary chord recognition tasks.

In MHSA, the input features for each time frame are divided into n_h segments, corresponding to individual attention heads. For an input matrix $\mathbf{M}$, the MHSA mechanism is defined as:

$$\text{Multihead} = \text{Concat}(h_1, \ldots, h_{n_h})W_O \tag{1}$$

where h_1 represents the output of the first attention head, and W_O is the weight matrix of the final fully connected layer that projects the concatenated outputs of all attention heads into the desired dimensional space.

The query, key, and value matrices for the j-th attention head are denoted as $Q_j = (\mathbf{M}W_Q)_j$, $K_j = (\mathbf{M}W_K)_j$, and $V_j = (\mathbf{M}W_V)_j$, respectively. These matrices are computed by projecting the input matrix $\mathbf{M}$ through fully connected layers W_Q, W_K, and W_V, ensuring that each head processes a unique representation of the input features.

The attention function for each head h_j is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_K}}\right)V \tag{2}$$

In this formulation, d_K represents the dimensionality of the key vectors. The scaling factor $\sqrt{d_K}$ is introduced to mitigate instability during training by preventing excessively large values in the dot-product operation.

The softmax activation function normalizes the attention scores into probabilities, ensuring that each key-value pair contributes appropriately. After concatenating the outputs from

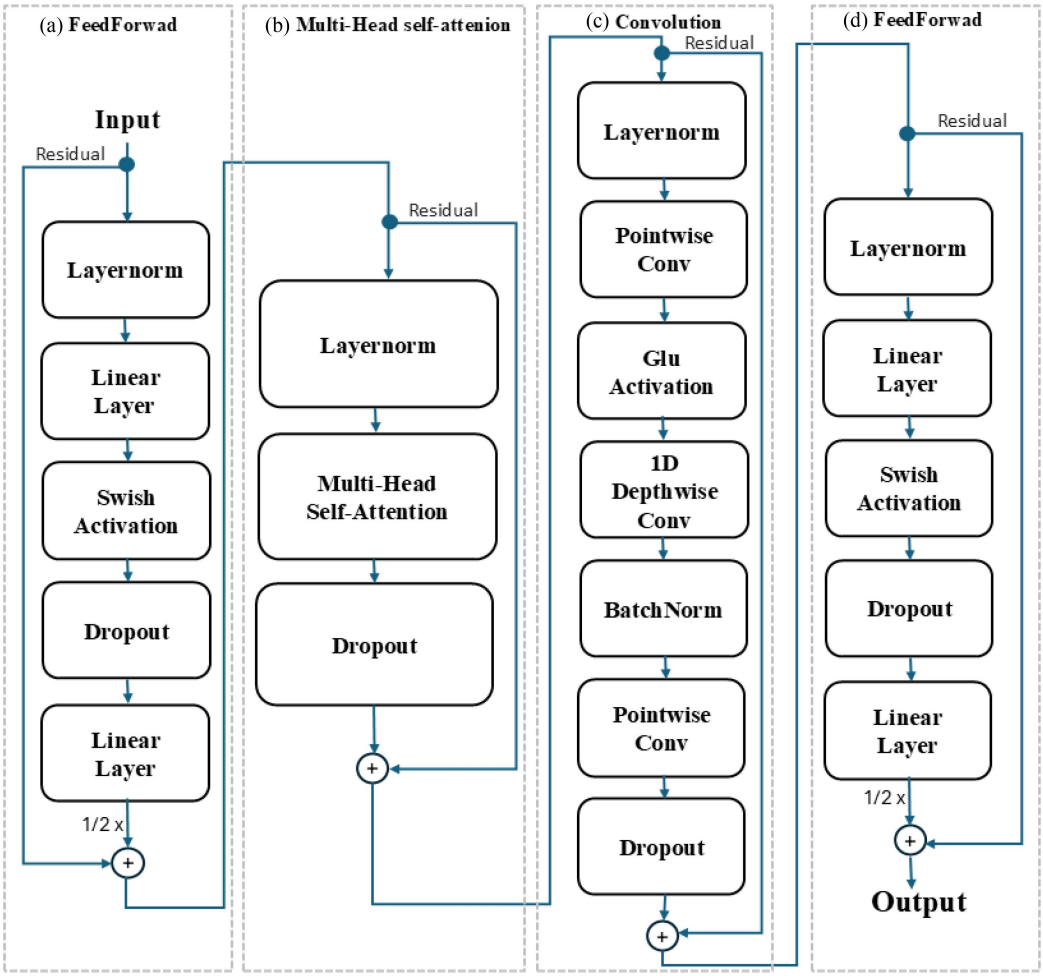

Fig. 2. Architecture Overview of Key Modules in ChordFormer Model. (a) Feed-Forward Module: Consists of a pre-normalization layer followed by a linear layer, Swish activation, and a dropout mechanism. Another linear layer projects back to the model dimensions. A residual connection enhances feature propagation. (b) Multi-Head Self-Attention Module: Incorporates multi-head self-attention with relative positional embeddings to capture global dependencies. Pre-normalization ensures stable gradient flow, and a residual connection is included to preserve input information. (c) Convolution Module: Features a pointwise convolution with an expansion factor of 2, coupled with a Gated Linear Unit (GLU) activation. This is followed by a 1-D depthwise convolution, batch normalization, and a Swish activation layer. Dropout is applied to ensure regularization, and a residual connection completes the module. (d) Feed-Forward Module: Reiterates the feed-forward structure with pre-normalization, Swish activation, and dropout for robust learning and dimensionality preservation.

all n_h attention heads, the result is processed through a fully connected layer W_O, projecting the combined output (of size $n_h \times d_{V_j}$) into the desired dimension.

To further enhance training stability, pre-norm residual connections [43], [44] and dropout are incorporated. These layers help mitigate overfitting and improve gradient flow, particularly in deeper architectures. Fig. 2(b) illustrates the multi-head self-attention block's structure, highlighting its modular design and data flow.

2) Position-Wise Convolution: To efficiently capture and utilize neighboring feature information within each time frame, a position-wise convolutional network is employed. Building on prior research [45], the convolutional module is designed to begin with a gating mechanism [46], incorporating a pointwise convolution followed by a gated linear unit (GLU). This initialization is subsequently enhanced with a one-dimensional depth-wise convolutional layer, further refining the module's capacity for feature extraction and contextual representation.

To maintain consistent feature size and sequence length, the convolutional block employs a pointwise convolution that expands the channels by a factor of 2, followed by a Gated Linear Unit (GLU) activation. By incorporating contextual information from adjacent frames, this position-wise convolutional structure effectively captures boundary transitions and smooth chord sequences. Batch normalization is applied immediately after the convolutional layers to enhance training stability. Fig. 2(c) presents a detailed scheme of the convolutional block, illustrating its structure and role within the overall framework.

3) Feed-Forward Block: The feed-forward block in ChordFormer consists of two linear layers with a nonlinear activation in between. A residual connection is applied over this block as in [43], [44], followed by layer normalization. Layer normalization is also applied within the residual path and to the input before the first linear transformation. ChordFormer employs the Swish activation function [47] and incorporates dropout to improve regularization. These design choices contribute to stable training

and better generalization, supporting the model's effectiveness in complex sequence modeling for chord recognition. Fig. 2(a) illustrates the architecture of the feed-forward Network (FFN) module.

4) Conformer Block: The Conformer block builds upon the previously described modules by incorporating a unique design that places the MHSA and the Convolution module between two feed-forward modules, as shown in Fig. 1. The detailed workflow of the Conformer block is illustrated in Fig. 2. The architecture is inspired by Macaron-Net [48], and used two half-step feed-forward layers one positioned before the attention layer and the other after it.

Following the principles of Macaron-Net, the proposed ChordFormer employs residual weights that are half-step in nature within the FNN_1 modules. Moreover, the second FNN_2 module is followed by a final layer normalization phase to enhance both stability and performance. Given an input $\mathbf{Z}_i$ to the Conformer block i, the output $\mathbf{Z}_i^{(o)}$ is calculated as follows:

$$\tilde{\mathbf{Z}}_i = \mathbf{Z}_i + \frac{1}{2}\text{FNN}_1(\mathbf{Z}_i) \tag{3}$$

$$\mathbf{Z}_i^{(a)} = \tilde{\mathbf{Z}}_i + \text{MHSA}(\tilde{\mathbf{Z}}_i) \tag{4}$$

$$\mathbf{Z}_i^{(c)} = \mathbf{Z}_i^{(a)} + \text{Conv}(\mathbf{Z}_i^{(a)}) \tag{5}$$

$$\mathbf{Z}_i^{(o)} = \text{LayerNorm}\left(\mathbf{Z}_i^{(c)} + \frac{1}{2}\text{FNN}_2(\mathbf{Z}_i^{(c)})\right) \tag{6}$$

Here, FFN represents the feed-forward module, MHSA denotes the Multi-Head Self-Attention module and Conv corresponds to the Convolution module, which focuses on local feature extraction and pattern recognition, as detailed in earlier sections. The 1/2 factor corresponds to a half-step feed-forward transition. Finally, LayerNorm refers to the layer normalization technique applied at the end to stabilize training and improve overall performance. This architecture strategically allocates the feed-forward layers around the attention and convolutional modules, extending the capabilities described in the previous section. By doing so, it efficiently balances the model's effectiveness to encapsulate global dependencies and local contextual information, making it particularly well-suited for music chord recognition.

D. Loss Function

The Conformer block outputs a hidden vector of M_c components, one for each time frame t. To transform this hidden state into meaningful chord predictions, a linear unit is applied, projecting the hidden state into six separate vectors, denoted as $\mathbf{C}_m^{(t,j)}$, where $j = 1, 2, \ldots, 6$. Each vector corresponds to a distinct component of the chord sequence, as described in Section III-A, which includes components like root, triad, bass, and extensions (7^{th}, 9^{th}, 11^{th}, and 13^{th}).

A softmax function is then applied to these vectors to compute the activation probabilities for the corresponding chord components. The activation probability $\beta_m^{(t,j)}$ for the j-th component at time t is computed as:

$$\beta_m^{(t,j)} = \frac{\exp\left(\mathbf{C}_m^{(t,j)}\right)}{\sum_{m'=1}^{M_j}\exp\left(\mathbf{C}_{m'}^{(t,j)}\right)} \quad \begin{matrix}\forall j = 1, \ldots, 6, \\ \forall m = 1, \ldots, M_j\end{matrix} \tag{7}$$

where $\beta_m^{(t,j)}$ represents the activation probability of class m for the j-th chord component at frame t, $\mathbf{C}_m^{(t,j)}$ is the corresponding score for class m, and M_j is the vocabulary size of the j-th chord component. This structured approach ensures that the predicted activations are normalized and interpretable, enabling effective modeling of complex harmonic relationships within chords.

To optimize the model, a weighted cross-entropy loss is employed from [6] to account for class imbalance. Let $\mathbf{G} = \{\mathbf{G}^{(1)}, \ldots, \mathbf{G}^{(T)}\}$ be the ground-truth chord sequence, where $\mathbf{G}^{(t)}$ is the chord vector at time t, and let $g_j^{(t)}$ be the ground-truth class index for the j-th component at time t. Then, the loss function is defined as:

$$\mathcal{L} = -\sum_{t=1}^{T}\sum_{j=1}^{6}\sum_{m=1}^{M_j} w_m^{(j)}\,\mathbb{I}[m = g_j^{(t)}]\log\beta_m^{(t,j)} \tag{8}$$

where $w_m^{(j)}$ is the weight assigned to class m of the j-th component to address class imbalance, $\mathbb{I}[m = g_j^{(t)}]$ is the indicator function that equals 1 if the predicted class matches the ground-truth class and 0 otherwise, and $\beta_m^{(t,j)}$ is the activation probability predicted for the respective class. This weighted loss ensures that rare chord classes are appropriately considered during training, enabling a balanced performance across all components.

E. Addressing Data Imbalance Through Class Re-Weighting

During the training phase, the issue of class imbalance became evident, even though the model does not directly classify individual chords. Certain chord extensions are poorly represented in the training set, resulting in an imbalance with an excess of majority class samples and a shortage of minority class samples. This imbalance skews the training process, causing the model to prioritize optimizing performance for classes with larger sample sizes while disadvantaging underrepresented classes.

An additional critical factor exacerbating bias in automatic chord recognition systems is the inherent ambiguity in audio chord annotation. Even among human experts, determining the "ground truth" chord label for a specific audio segment often involves subjective interpretation. This ambiguity can result in overlapping class boundaries, generating uncertainty and reducing classification accuracy. Such ambiguity disproportionately impacts classes with lower prior probabilities, further limiting the overall performance.

To overcome these limitations, we applied a class re-weighting strategy that assigns a class-specific weight factor $w_m^{(j)}$ for each possible value of a chord component, using approach presented by Jiang et al. [6]. This re-weighting approach mitigates class imbalance by amplifying the contribution of underrepresented classes in the loss function during the training

phase. The weight factor is defined as follows:

$$w_m^{(j)} = \min\left\{\left(\frac{n_m^{(j)}}{\max_{m'} n_{m'}^{(j)}}\right)^{-\gamma}, w_{\max}\right\}, \quad (9)$$

where $n_m^{(j)}$ represents the total number of training samples associated with class m in the j-th component. The balancing factor γ, constrained to the range $0 \leq \gamma \leq 1$, regulates the degree of weighting adjustment, with smaller values resulting in more equitable weight distribution across classes. Additionally, $w_{\max}$ serves as the clamping parameter, set to $w_{\max} \geq 1$, which prevents the weight values from reaching excessively large magnitudes.

By adopting this weighting strategy, underrepresented classes with fewer samples are assigned larger weights, effectively mitigating bias in the training process. Lower values of γ yield a more uniform weight distribution, whereas higher values allow the weights to more closely reflect the sample distribution. This approach ensures that infrequent chord classes are adequately emphasized during training, thereby reducing the adverse effects of data imbalance and annotation ambiguity. The proposed ChordFormer leverages this re-weighting strategy to establish a more balanced training procedure, enhancing its robustness and accuracy across a diverse range of chord classes.

F. Decoding Strategy for Chord Recognition

A straightforward approach to decoding the final chord sequence from the activation probabilities involves selecting the class that has the highest probability for each component at every frame. However, this method often results in excessive transitions between adjacent chords, as it lacks a mechanism to impose penalties for immediate changes. Additionally, this approach provides limited control over the output chord vocabulary, which can lead to inconsistencies in the predictions. To overcome these challenges, ChordFormer incorporates a linear Conditional Random Field (CRF) from [6] for decoding the final chord sequence. This technique effectively balances activation probabilities with temporal smoothness, ensuring more coherent chord transitions. The CRF models the probability of a chord sequence $\mathbf{S}$ given the audio features $\mathbf{X}$ using the following formulation:

$$P(\mathbf{S} \mid \mathbf{X}) \propto \phi(\mathbf{S}^{(1)}, \mathbf{X}) \prod_{t=2}^{T} \phi(\mathbf{S}^{(t)}, \mathbf{X}) \psi(\mathbf{S}^{(t-1)}, \mathbf{S}^{(t)}). \quad (10)$$

Here, ϕ is the observation potential function that evaluates the compatibility between the audio features and the chord predictions, while ψ is the transition potential function that enforces temporal coherence by penalizing immediate transitions.

The observation potential function is defined as:

$$\phi(\mathbf{S}^{(t)}, \mathbf{X}) = \exp\left(\sum_{j=1}^{6} \sum_{m=1}^{M_j} \mathbb{I}[m = s_j^{(t)}] \log \beta_m^{(t,j)}\right). \quad (11)$$

The transition potential function that regulates the smoothness of the chord sequence is expressed as:

$$\psi(\mathbf{S}^{(t-1)}, \mathbf{S}^{(t)}) = \exp\left(-\lambda \cdot \mathbb{I}[\mathbf{S}^{(t-1)} \neq \mathbf{S}^{(t)}]\right), \quad (12)$$

where λ is a hyperparameter that controls the penalty for transitions between different chords. Higher values of λ enforce smoother transitions, reducing immediate changes in the chord sequence. We used a value of $\lambda = 30$ as in [6]. While we evaluated λ values between 2 and 35, the best results were achieved with $\lambda = 30$.

By integrating these potentials, the CRF-based decoding method provides a robust balance between prediction accuracy and temporal coherence. This method not only reduces excessive chord transitions but also allows explicit control over the output chord vocabulary, making it particularly well-suited for large-vocabulary chord recognition tasks [17].

IV. Experiments

A set of experiments have been conducted to evaluate the performance of the proposed ChordFormer model in comparison to other baseline models from the literature. This section explains how data are organized and preprocessed, describes the model's training setup and the evaluation metrics for assessing models' performance, and presents the achieved results, highlighting the differences with respect to the other models.

A. Dataset Description and Preprocessing

The evaluation of the proposed model has been carried out using the dataset by Humphrey and Bello [16], [21] comprising 1,217 songs obtained from the Isophonics, Billboard, and MARL collections. To ensure a rigorous and unbiased evaluation, we implemented a 5-fold cross-validation strategy, using the same training, validation, and testing ratio defined in [21]. Specifically, 60% of the dataset has been reserved for training, 20% for validation, and 20% for testing. Such a partitioning is consistent with prior studies, thus enabling a direct and reliable comparison of results while maintaining methodological integrity.

The CQT spectrograms were generated from the audio data using the Librosa library, configured with a sampling rate of 22,050 Hz and a hop length of 512. The frequency range extends from note C1 (inclusive) to C8 (exclusive), utilizing 36 bins per octave, resulting in a total of 252 CQT bins. After that, the spectrogram was transformed into the decibel scale using Librosa's `amplitude_to_db` function, with normalization relative to the maximum amplitude. This step enhances interpretability by representing the data on a logarithmic scale, which aligns with the human auditory perception of sound intensity. To enhance the model's robustness, data augmentation was applied to the training set using pitch-shifting techniques [6]. These pitch shifts range from -5 to $+6$ semitones, generating augmented features through direct modifications of the CQT spectrograms. The corresponding annotated chord labels were adjusted to align with the altered audio attributes. This augmentation strategy ensures a more diverse and representative training dataset while preserving precise chord annotations. By incorporating this approach, the model's ability to generalize effectively across varied musical contexts was significantly improved, advancing its performance in music chord recognition tasks.

TABLE I
CHORDFORMER PARAMETERS

Parameter	Value
input_dim	256
num_heads	16
ffn_dim	1024
num_layers	4
depthwise_conv_kernel_size	31
output_dim	100

B. Training ChordFormer and Hyperparameters

Table I summarizes the hyperparameters employed for the ChordFormer, selected through empirical tuning based on the performance on the validation set during a 5-fold cross-validation process. The input dimensionality was set to 256, and 16 attention heads were used to balance model capacity and efficiency. The feed-forward network dimension was set to 1024 to enhance representational power, while the architecture comprises 4 Conformer layers to capture hierarchical features. A depthwise convolution kernel of size 31 was chosen to effectively model local temporal patterns, and the output dimension was set at 100 to align with the structured label space. These settings yielded consistently strong performance across validation folds. The AdamW optimizer was used to train the neural network, incorporating a dynamic learning rate schedule to enhance optimization. The initial learning rate was set at 1×10^{-3} and reduced by 90% if no improvement was observed in the validation loss over five consecutive epochs. Training concludes when the learning rate drops below 1×10^{-6}, ensuring efficient computation and minimizing overfitting risks.

Every epoch involves the random extraction of a 1,000-frame segment (approximately 23.2 seconds) from every song to introduce temporal diversity. Mini-batches were formed by combining 24 such segments, effectively balancing computational efficiency with data variability to facilitate robust model training.

C. Evaluation Metrics

To facilitate a thorough comparison with existing models, the proposed ChordFormer was evaluated using the widely adopted `mir_eval` library [49], a standard tool in automatic chord recognition research. The assessment includes Root Scores, Maj-Min Scores, Seventh Scores, Thirds Scores, Triad Scores, Tetrads Scores, and MIREX Scores. These metrics collectively provide a comprehensive evaluation of chord recognition performance, capturing various aspects of chord prediction.

A key method used to measure these evaluation metrics is the Weighted Chord Symbol Recall (WCSR), proposed by Pauwels and Peeters [50], which quantifies the average frame-level accuracy of chord recognition. It is defined as

$$\text{WCSR} = \left(\frac{\sum_{i=1}^{N} d_i}{\sum_{i=1}^{N} D_i} \right) \times 100 \tag{13}$$

where N is the total number of songs in the evaluation dataset, D_i denotes the total duration of the i-th song, and d_i corresponds to the duration of correctly predicted chord frames for the i-th song. This metric enables a quantitative understanding of how well a model performs across different musical contexts, providing a robust measure of its chord recognition capabilities. We also evaluated the performance of ChordFormer by ignoring the Root across different metrics and calculating the Average Rootless Weighted Chord Symbol Recall (ARWCSR). First, WCSR of chords is calculated using (13). Afterward, the ARWCSR is computed as follows:

$$\text{ARWCSR} = \frac{1}{|C|} \sum_{c \in C} \text{WCSR}(c) \tag{14}$$

where C represents the subset of chords, which includes MajMin, Sevenths, Thirds, Triads, Tetrads, and their corresponding inversions obtained by the `mir_eval` library.

D. Evaluation Metrics for Large Vocabulary

To evaluate the model on a large vocabulary and address the challenges posed by imbalanced class, we followed the same approach proposed by Jiang et al. [6] and tested it on a vocabulary V consisting of 301 distinct chords, including Basic triads, Inverted triads, Seventh chords, Extended chords, Suspended chords, Slash chords, and a Non-chord class (N). In particular, the evaluation incorporates two critical metrics from [6]: mean frame-wise accuracy (acc_{frame}) and mean class-wise accuracy (acc_{class}). These metrics are designed to measure both the overall performance and the fairness of chord recognition across different chord classes.

The mean frame-wise accuracy (acc_{frame}) is defined as:

$$acc_{frame} = \frac{\sum_{i=1}^{n} f_i}{\sum_{i=1}^{n} F_i}, \tag{15}$$

where n represents the total number of tracks in the test set, F_i is the total number of frames in the i-th track, and f_i is the number of correctly predicted frames in the i-th track over the chord vocabulary V.

The mean class-wise accuracy (acc_{class}) is defined as:

$$acc_{class} = \frac{1}{|V|} \sum_{v \in V} \frac{\sum_{i=1}^{n} f_i^v}{\sum_{i=1}^{n} F_i^v}, \tag{16}$$

where F_i^v is the number of frames in the i-th track labeled with the ground-truth chord v, and f_i^v is the number of frames correctly predicted as v in the i-th track. Here, $|V|$ denotes the total number of unique chord classes in the vocabulary.

V. RESULTS AND DISCUSSION

To ensure a fair and consistent comparison, the proposed ChordFormer model and its variant ChordFormer-R (with reweighted loss) were evaluated against several baseline models using the same structural chord embeddings and training hyperparameters. The baseline models include

- CNN: a convolutional neural network designed for automatic chord recognition proposed by Korzeniowski and Widmer [15].
- CNN+BLSTM: a convolutional neural network that incorporates a bidirectional long short-term memory (BLSTM) layer to enhance sequence modeling, as proposed by Jiang et al. [6].
- BTC+CNN: a model based on Bidirectional Self-Attention, evaluated by removing the position embedding

TABLE II
COMPARISON OF WEIGHTED CHORD SYMBOL RECALL (WCSR) ACROSS DIFFERENT BASELINE MODELS ON VARIOUS METRICS

Model	Root	Thirds	MajMinor	Triads	Sevenths	Tetrads	MIREX
BTC+CNN [7]	54.28	47.94	49.00	44.67	37.99	34.01	47.94
Transformer [9]	78.55	72.91	74.24	67.75	57.42	51.46	72.23
CNN [15]	81.76	78.69	80.88	74.13	67.27	60.48	79.42
Transformer+CNN	82.40	79.40	81.67	74.88	67.84	61.04	80.22
CNN+BLSTM [6]	83.39	80.04	82.62	75.91	69.78	62.87	81.52
ChordFormer-R	83.87	80.54	81.86	76.02	69.80	63.48	82.98
ChordFormer	**84.60**	**81.76**	**84.35**	**77.69**	**72.80**	**65.75**	**83.65**

layer and integrating self-attention with CNN, as proposed by Rowe and Tzanetakis [7].

- Transformer: the original transformer model proposed by Vaswani [9], along with its combination with a CNN as a feature extractor (Transformer+CNN).

The WCSR results are reported in Table II and clearly demonstrate the superior performance of ChordFormer and ChordFormer-R. In fact, ChordFormer achieves a Root accuracy of 84.60% , MajMinor accuracy of 84.35% , and a MIREX score of 83.65% . These results highlight the Conformer block's ability to combine convolutional layers and self-attention mechanisms, enabling superior local and global feature modeling. Upon analyzing the results, we found that the frame-wise accuracy of 79.08% is primarily influenced by the MajMinor category, which has the largest sample size, achieving an accuracy of 84.35% . The Triads set, sharing common chord types like min and maj with the Sevenths set, outperforms the Sevenths set with an accuracy of 77.69% as compared with 72.80% . These findings highlight the influence of chord type distribution on recognition performance, particularly in large-vocabulary tasks. Additionally, ChordFormer-R, which incorporates a reweighted loss function to address class imbalance, shows further improvements, particularly for rare chord classes. By assigning higher weights to underrepresented components during training, ChordFormer-R ensures a more balanced optimization process, showing its robustness in handling large-vocabulary chord recognition tasks effectively. The results presented here for CNN+BLSTM differ from those reported in their paper. We obtained these results by training the model from scratch on our system using the same publicly available code and settings they provided on GitHub[1] . To ensure a fair comparison, we adopted only the model from [7], as it uses a different chord structure and data split. Additionally, we removed the positional encoding from their model because it resulted in significantly degraded performance, which is why those results are not reported here. Instead, we performed an ablation study on positional encoding using our proposed ChordFormer architecture to analyze its impact on model performance. The hyperparameters used to train all the baseline models were adopted directly from their respective original papers to ensure fair and consistent comparison.

[1] https://github.com/music-x-lab/ISMIR2019-Large-Vocabulary-Chord-Recognition

Chord inversions are crucial for accurate chord identification in real-world musical contexts. Inversions occur when the root of a chord is not the lowest note, and recognizing these variations is vital for accurate harmonic analysis. To further assess ChordFormer capability in handling complex harmonic structures, we evaluated its performance on chord inversions. Table IV compares the performance of ChordFormer with several baseline models, including CNN+BLSTM, Transformer+CNN, and Transformer, across five inversion types: Thirds_Inv, MajMin_Inv, Triads_Inv, Sevenths_Inv, and Tetrads_Inv.

The results demonstrate that ChordFormer outperforms the other models across all inversion types, with scores of 79.23 for Thirds_Inv, 81.90 for MajMin_Inv, and 71.06 for Sevenths_Inv. These results emphasize ChordFormer effectiveness in accurately recognizing inversions.

To assess the model's ability to generalize across various chord qualities, including extended chord types such as Sevenths, Tetrads, and their inversions, we conducted an evaluation by ignoring the Root. This approach allowed us to focus on the model's performance in chord quality recognition, independently of the root position, thereby emphasizing its ability to capture the harmonic chord structure.

The results presented in Table V offer a detailed comparison of ChordFormer and several baseline models. ChordFormer consistently outperforms the other models with a remarkable ARWCSR of 80.45% , highlighting its superior ability to handle chord recognition across a wide range of chord types. This success underscores the model's proficiency in identifying extended chord and inversions, even in the absence of the Root.

Interestingly, although ChordFormer achieves an impressive Root accuracy of 84.60% in the full evaluation, its performance diminishes when both Root and chord quality need to be recognized simultaneously. This underperformance suggests that while ChordFormer is adept at recognizing chord qualities when the Root is excluded, it still encounters challenges when tasked with root recognition, which affects its ability to fully capture the overall chord structure. Notably, as shown in Table II, Table IV, and Table V, this trend is not unique to ChordFormer, as other baseline models also exhibit substantially improved performance when the Root is ignored. These findings emphasize the importance of improving root recognition in future model iterations, particularly for scenarios where the Root may be ambiguous or not explicitly provided.

TABLE III
COMPARISON OF ACC_{FRAME} AND ACC_{CLASS} BETWEEN CNN+BLSTM AND CHORDFORMER MODELS

Re-weighting Value	acc_{frame} CNN+BLSTM [6]	acc_{frame} ChordFormer	acc_{class} CNN+BLSTM [6]	acc_{class} ChordFormer
No Re-weighting	0.7676	**0.7908**	0.3315	0.3934
(0.3, 10.0)	0.7659	0.7801	0.3692	0.4426
(0.5, 10.0)	0.7402	0.7772	0.3821	0.4406
(0.7, 20.0)	0.7117	0.7416	0.3823	**0.4471**
(1.0, 20.0)	0.6512	0.6994	0.3546	0.4157

TABLE IV
COMPARISON OF WCSR ACROSS DIFFERENT BASELINE MODELS ON VARIOUS INVERSIONS OF THE METRICS

Model	Thirds_Inv	MajMin_Inv	Triads_Inv	Sevenths_Inv	Tetrads_Inv
BTC+CNN [7]	46.05	47.16	42.99	36.64	32.80
Transformer [9]	70.44	71.93	65.64	55.76	49.97
CNN [15]	67.29	78.61	72.06	65.66	59.03
Transformer+CNN	77.24	79.57	72.97	66.40	59.76
CNN+BLSTM [6]	77.82	80.48	73.99	68.30	61.55
ChordFormer-R	78.25	80.67	74.00	68.83	61.95
ChordFormer	**79.23**	**81.90**	**75.53**	**71.06**	**64.27**

TABLE V
COMPARISON OF WCSR AND ARWCSR AROSS DIFFERENT BASELINE MODELS ON VARIOUS METRICS AND THEIR INVERSIONS WHEN IGNORING THE ROOT

Model	Root	Thirds	MajorMinor	Triads	Sevenths	Tetrads	Thirds_Inv	MajMin_Inv	Triads_Inv	Sevenths_Inv	Tetrads_Inv	ARWCSR
BTC+CNN [7]	54.28	81.64	76.85	70.06	59.32	53.10	71.45	71.78	65.44	55.53	49.70	65.49
Transformer [9]	78.55	86.85	83.96	76.61	64.22	57.55	78.61	79.57	72.61	61.15	54.79	71.59
CNN [15]	81.76	90.27	88.23	80.87	72.63	65.28	82.23	84.14	77.12	69.84	62.78	77.34
Transformer+CNN	82.40	90.51	88.87	81.47	73.11	65.78	83.15	85.07	77.99	70.49	63.43	77.99
CNN+BLSTM [6]	83.39	90.53	88.77	81.58	74.42	67.04	82.72	84.98	78.11	71.88	64.76	78.48
ChordFormer-R	83.87	90.64	88.89	81.71	74.63	67.29	83.17	85.11	78.33	72.11	64.90	78.67
ChordFormer	**84.60**	**91.42**	**90.09**	**82.98**	**77.21**	**69.71**	**84.14**	**86.67**	**79.84**	**74.84**	**67.58**	**80.45**

Further performance results are provided in Table III, which reports frame-wise accuracy (acc_{frame}) and class-wise accuracy (acc_{class}) across different re-weighting configurations for CNN+BLSTM and ChordFormer models. Without re-weighting, ChordFormer consistently outperforms CNN+BLSTM, achieving a higher acc_{frame} of 0.7908 compared to CNN+BLSTM 0.7676 and demonstrating a substantial improvement in acc_{class} (0.3934 vs. 0.3315). These results suggest that ChordFormer hybrid Conformer architecture excels in harmonic modeling, effectively capturing both local and long-range dependencies within musical sequences.

As class re-weighting increases, both models exhibit a decline in acc_{frame} due to the prioritization of rare chord classes. This decline is more pronounced for CNN+BLSTM, dropping from 0.7676 to 0.6512, while ChordFormer demonstrates greater resilience, maintaining acc_{frame} at 0.6994 even at higher re-weighting values (1.0, 20.0). Conversely, acc_{class} improves across all re-weighting configurations, with ChordFormer achieving the highest acc_{class} of 0.4471 at (0.7, 20.0), compared to CNN+BLSTM 0.3823. This highlights ChordFormer superior ability to generalize across rare chord classes, a critical advantage for real-world music information retrieval (MIR) applications.

The confusion matrix of chord qualities is considered only for cases where the root is correctly predicted by the model. The matrix, reported in Fig. 4, provides a detailed visualization of ChordFormer performance with re-weighting factors ($\gamma = 0.5, w_{\max} = 10$), highlighting how these parameters affect chord quality recognition. Strong diagonal dominance reflects ChordFormer high accuracy in predicting common chord quality such as *maj*, *min*, and *7*, which are frequent in Western music. However, moderate off-diagonal intensities indicate confusion among harmonically similar chords, such as *min7*, *hdim7*, and *dim7*, or between extensions like *9* and *11*. While re-weighting strategies improve recognition for rare chord quality like *maj9*, *13*, and *sus4(b7)*, they introduce slight trade-offs in overall frame-wise accuracy due to prioritization of minority chord qualities.

To further evaluate the system's performance on a large chord vocabulary, a comprehensive chord vocabulary encompassing all chord qualities in the dataset is adopted. The class-wise accuracy of each chord component is assessed during the decoding process, offering a detailed view of ChordFormer performance across a diverse range of harmonic structures. The graph in Fig. 3 provides an in-depth analysis of chord component recall under various re-weighting configurations ($\gamma, w_{\max}$). Common chord

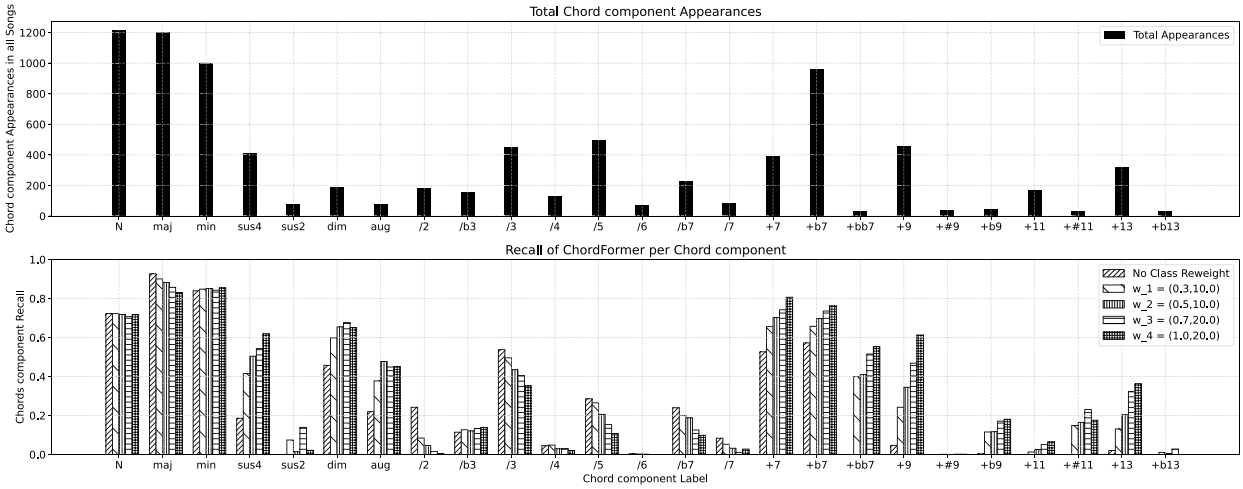

Fig. 3. Top: chord components appearance in the song dataset; Bottom: ChordFormer recall values for each chord component under varying re-weight factors $(\gamma, w_{\max})$.

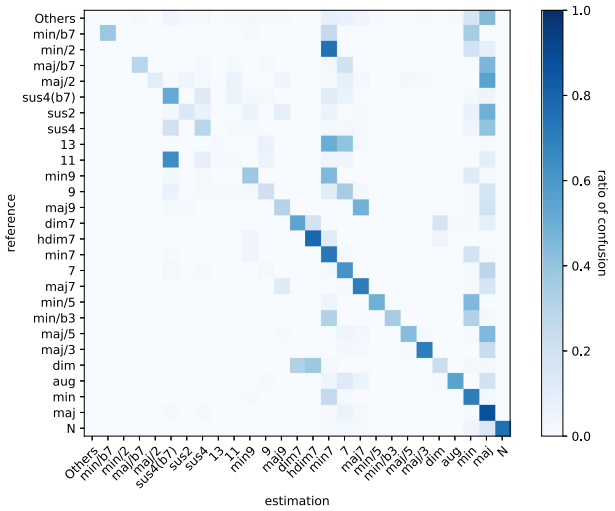

Fig. 4. Confusion matrix of ChordFormer with re-weighting factors ($\gamma = 0.5, w_{\max} = 10$).

components such as *maj* and *min* consistently achieve high recall values, indicating ChordFormer strong ability to model frequent harmonic patterns. Rare components like *dim*, *aug*, and extended chords such as *+11* and *+13* show significant improvements under re-weighting strategies, with configurations like $(0.7, 20.0)$ delivering noticeable gains.

However, trade-offs emerge, as higher re-weighting values $(1.0, 20.0)$ lead to slight reductions in recall for dominant chord components, highlighting the challenge of balancing overall accuracy with enhanced recognition of rare chords. Configurations like $(0.5, 10.0)$ and $(0.7, 20.0)$ emerge as optimal trade-offs, enhancing recall for rare chords while maintaining strong performance for frequent ones. These evaluations underscore ChordFormer robustness in managing class imbalance and handling large-vocabulary chord recognition tasks, emphasizing the importance of parameter tuning to achieve balanced performance across diverse chord classes.

VI. Ablation Studies

The ablation studies conducted on ChordFormer provide critical insights into the various components that contribute to its superior performance in large-vocabulary chord recognition. These studies examine the influence of kernel size, FFN hidden layer dimensions, self-attention heads, model variants, and positional encodings on the model's ability to handle complex harmonic structures.

A. Convolutional Kernel Size

The analysis of different kernel sizes in the convolutional layers indicates that increasing the kernel size enhances the model's ability to capture more complex local harmonic patterns. As the kernel size increases, the model's capacity to extract features from a broader context improves, leading to better performance across various metrics. Based on the results reported in Table VI, the kernel size of 31 was chosen as the optimal size, as it provided the best performance across key metrics, balancing performance improvement with computational efficiency and avoiding unnecessary complexity. We did not further increase the kernel size, as the performance difference between the 17 and 31 kernel sizes was minimal, making it unnecessary to introduce additional complexity and computational cost.

B. FFN Hidden Layer Dimensions

When evaluating the FFN hidden layer dimensions, it became clear that larger hidden layers provide more capacity for the model to capture complex chord structures. Based on the results reported in Table VII, we observe that the 144-dimensional configuration achieved the best class accuracy (39.46%) and frame accuracy (79.08%). However, this comes at the cost of a lower performance in other metrics, such as Root accuracy and MIREX. For higher dimensions, particularly with 512 and 1024, the performance in these metrics improved, though class and frame accuracy slightly decreased. The 1024-dimensional configuration provided the best balance across all metrics, achieving

TABLE VI
WCSR OF CHORDFORMER WITH DIFFERENT KERNEL SIZES ACROSS VARIOUS METRICS

Kernel Size	Root	Thirds	MajMin	Triads	Sevenths	Tetrads	MIREX	acc_{class}	acc_{frame}
3	82.42	78.97	81.48	75.06	69.97	63.27	81.22	36.26	77.65
7	83.61	80.45	82.91	76.36	71.09	64.24	82.41	37.47	78.09
17	**84.63**	81.40	83.82	77.28	72.07	65.09	83.16	38.94	78.32
31	84.60	**81.76**	**84.35**	**77.69**	**72.80**	**65.75**	**83.65**	**39.34**	**79.08**

TABLE VII
WCSR OF CHORDFORMER WITH DIFFERENT FFN HIDDEN LAYER DIMENSIONS ACROSS VARIOUS METRICS

FFN Dim	Root	Thirds	MajMin	Triads	Sevenths	Tetrads	MIREX	acc_{class}	acc_{frame}
144	84.46	81.45	83.79	77.25	72.33	65.37	83.32	**39.46**	79.08
256	84.45	81.42	83.91	77.36	72.31	65.31	83.35	38.39	78.99
512	84.50	81.47	83.86	77.23	72.64	65.55	83.16	35.62	78.89
1024	**84.60**	**81.76**	**84.35**	**77.69**	**72.80**	**65.75**	**83.65**	39.34	**79.08**

TABLE VIII
WCSR OF CHORDFORMER WITH DIFFERENT NUMBERS OF SELF-ATTENTION HEADS ACROSS VARIOUS METRICS

Heads	Root	Thirds	MajMin	Triads	Sevenths	Tetrads	MIREX	acc_{class}	acc_{frame}
4	84.69	81.75	84.09	77.55	72.28	65.32	83.62	0.3884	0.7877
8	**84.86**	81.73	84.21	77.63	72.56	65.56	83.74	0.3926	0.7860
16	84.60	**81.76**	**84.35**	**77.69**	**72.80**	**65.75**	**83.65**	**0.3934**	**0.7908**
32	84.65	81.35	83.89	77.27	72.29	65.28	83.38	0.3853	0.7859

84.60% in Root accuracy, 84.35% in MajMin, and 83.65% in MIREX, while still maintaining competitive performance in class and frame accuracy. Therefore, 1024 was chosen as the optimal dimension, as it offered a more balanced result across all metrics compared to the other configurations.

C. Self-Attention Head

The number of self-attention heads plays a pivotal role in the model's ability to capture long-range dependencies. Increasing the number of attention heads generally improves performance, as it allows the model to focus on a greater number of dependencies in the input data. However, as the number of heads continues to increase, the performance tends to a plateau, suggesting that excessive complexity may lead to diminishing returns. This allowed us to select a number of attention heads able to balance between capturing fine-grained dependencies and avoiding overfitting.

Table VIII shows that the most balanced results were obtained when the number of heads is set to 16, where the performance across most metrics is better, though slightly lower in Root accuracy compared to smaller configurations (e.g., 4 heads).

D. Removing Different Parts of the ChordFormer

The results presented in Table IX highlight the performance of various ChordFormer model variants and the impact of removing different components of the architecture. The baseline ChordFormer model consistently achieves superior performance across all metrics, underscoring its effectiveness in large-vocabulary chord recognition.

When the Swish activation function is replaced with the ReLU, a slight reduction in performance is observed, indicating that this modification does not substantially enhance the model's ability to capture the complex harmonic structures inherent in music.

The removal of CNN layers results in a significant decline in performance across all key metrics. This underscores the essential role that CNN layers play in capturing local harmonic features, which are crucial for accurate chord recognition. Moreover, further reducing the complexity of the model by removing both CNN layers and one of the FFN layers, and replacing it with a traditional FFN architecture, results in even greater performance degradation. These findings emphasize the importance of the dual FFN layers in maintaining the model's ability to capture the broader contextual relationships in musical sequences.

Even when using a reduced configuration, such as a single FFN layer and halved residual connections, performance drops across all metrics, particularly in the model's ability to handle complex chord structures. This reinforces the notion that the good ChordFormer performance relies on its dual FFN layers, which are critical for achieving high accuracy in chord recognition tasks.

The Flat model simplifies the architecture by removing the structural chord representation and instead employs a

TABLE IX
WCSR OF CHORDFORMER COMPARISON ACROSS DIFFERENT MODELS. I) CHORDFORMER: THE BASELINE MODEL WITH CONVOLUTIONAL LAYERS AND SELF-ATTENTION. II) RELU INSTEAD OF SWISH: SWISH ACTIVATION REPLACED WITH RELU. III) NO CNN LAYER: MODEL WITHOUT CNN LAYERS. IV) CNN REMOVED, TRADITIONAL FFN: REPLACED CNN LAYERS WITH A TRADITIONAL FFN LAYER. V) 2ND HALF STEP FFN WITH RESIDUAL: ONLY SECOND HALF STEP FFN USED WITH RESIDUAL CONNECTIONS. VI) FULL RESIDUAL CONNECTIONS: FULL RESIDUAL CONNECTIONS AFTER FFN LAYERS. VII) FLAT MODEL: NO STRUCTURAL CHORD REPRESENTATION, USING A LINEAR LAYER WITH FLAT SOFTMAX.

Model	Root	Thirds	MajMin	Triads	Sevenths	Tetrads	MIREX	acc_{class}	acc_{frame}
ChordFormer	**84.60**	**81.76**	**84.35**	**77.69**	**72.80**	**65.75**	**83.65**	**0.3934**	**0.7907**
ReLU Activation	84.55	81.60	84.05	77.45	72.53	65.54	83.56	0.3898	0.7871
Without_CNN	79.94	76.48	79.10	72.67	67.56	60.89	78.48	0.3045	0.7686
Without_CNN_FNN_1	79.47	75.88	78.42	71.92	66.13	59.50	77.98	0.282	0.7539
FNN_2 1/2(Residual)	79.19	75.84	78.58	72.09	66.17	59.59	78.10	0.2858	0.7535
FFN_Full(Residual)	78.95	75.61	78.14	71.78	66.04	59.56	77.75	0.3051	0.7560
Flat	83.85	80.86	83.32	76.90	71.99	65.17	83.24	0.3912	0.7884

TABLE X
PERFORMANCE COMPARISON OF CHORDFORMER ACROSS DIFFERENT POSITIONAL ENCODING VARIANTS. I) NO: CHORDFORMER WITHOUT POSITIONAL ENCODING. II) $x + \mathrm{PE}_{\mathrm{REL}}(x)$ [51]: CHORDFORMER WITH RELATIVE POSITIONAL ENCODING. III) $x + \mathrm{PE}_{\mathrm{ABS}}(x)$ [9]: CHORDFORMER WITH ABSOLUTE POSITIONAL ENCODING. IV) $x + K \cdot \mathrm{PE}_{\mathrm{ABS}}(x)$: CHORDFORMER WITH ABSOLUTE POSITIONAL ENCODING SCALED BY A LEARNABLE SCALAR k. V) $x + \mathrm{FFN}(\mathrm{PE}_{\mathrm{ABS}}(x))$: CHORDFORMER WITH ABSOLUTE POSITIONAL ENCODING TRANSFORMED VIA A FEEDFORWARD NEURAL NETWORK.

Positional Encoding	Root	Thirds	MajMin	Triads	Sevenths	Tetrads	MIREX	acc_{class}	acc_{frame}
No	**84.60**	**81.76**	**84.35**	**77.69**	**72.80**	**65.75**	**83.65**	**0.3934**	**0.7907**
$x + \mathrm{PE}_{\mathrm{rel}}(x)$	79.61	76.09	78.72	72.21	66.76	60.08	77.98	0.2957	0.7601
$x + \mathrm{PE}_{\mathrm{abs}}(x)$	81.84	78.64	81.25	74.82	69.20	62.45	80.63	0.3733	0.7782
$x + k \cdot \mathrm{PE}_{\mathrm{abs}}(x)$, k is learnable	81.52	78.39	81.18	74.69	69.41	62.57	80.51	0.3543	0.7813
$x + \mathrm{FFN}(\mathrm{PE}_{\mathrm{abs}}(x))$	83.24	80.12	82.64	76.18	71.28	64.36	82.15	0.3865	0.7893

single linear layer with a flat softmax output over all possible combinations of roots and chord types in the dictionary, resulting in 301 classes including the no-chord label. Although this model performs well compared to baseline approaches, it still underperforms relative to ChordFormer when the structural chord representation is preserved. These results confirm that the structural representation is important for the model's success, enabling it to effectively learn and distinguish chord types across a wide range of musical contexts. Notably, the performance of ChordFormer even without this structural representation highlights its inherent ability to capture both local harmonic patterns and long-range dependencies capabilities that are essential for accurate and robust chord recognition.

E. Positional Encoding

The results reported in Table X examine the impact of different positional encoding techniques on ChordFormer performance. In the baseline scenario, where no positional encoding is applied, the model achieves strong results, indicating that it effectively processes the inherent sequential and harmonic information embedded in the input data without the need for additional positional encoding. Note that the model input x is derived from the CQT, which inherently captures both logarithmic frequency and temporal information. Also, the Conformer architecture can be viewed as a transformer with implicit relative positional encoding (via convolutional layers). For these reasons, adding explicit positional embeddings has minimal impact on the model's performance. It is also observed that ChordFormer exhibits the lowest performance when relative positional embeddings, $\mathrm{PE}_{\mathrm{rel}}(x)$, computed according to the method described in [51], are directly added to the input features:

$$x_{\mathrm{output}} = x + \mathrm{PE}_{\mathrm{rel}}(x) \tag{17}$$

In contrast, the introduction of $\mathrm{PE}_{\mathrm{abs}}(x)$ using sinusoidal functions [9] as in Equation (18), yields a modest improvement across most metrics, particularly in distinguishing between chord types.

$$x_{\mathrm{output}} = x + \mathrm{PE}_{\mathrm{abs}}(x) \tag{18}$$

This encoding helps the model better understand the positional relationships between notes, leading to a slight increase in ChordFormer performance. However, this boost remains limited and does not result in a substantial improvement of the overall performance

we perform experiment with learnable or adjustable embeddings to enhance the information contained in the CQT features. While these modifications did not yield performance improvements, they provided valuable insights and new directions for future research.

Among the variants tested, the absolute positional encoding variant, defined as

$$x_{\mathrm{output}} = x + k \cdot \mathrm{PE}_{\mathrm{abs}}(x) \tag{19}$$

where the positional encoding $\mathrm{PE}_{\mathrm{abs}}(x)$ is computed using sinusoidal functions and scaled by a learnable parameter k, demonstrates consistent improvements across all evaluation metrics. This formulation allows the model to adaptively modulate the

contribution of positional information based on the CQT features, with k controlling the overall influence of the positional encoding. While this adaptive scaling provides a modest enhancement, its overall impact on performance remains relatively marginal.

The best performance across all evaluation metrics is achieved with the absolute positional encoding variant defined as

$$x_{\text{output}} = x + \text{FFN}\left(\text{PE}_{\text{abs}}(x)\right) \tag{20}$$

as shown in Table X. In this variant, sinusoidal positional embeddings $\text{PE}_{\text{abs}}(x)$ are first transformed through a feedforward neural network (FFN) layer before being added to the input vector. This approach allows the model to learn a more flexible and task-adaptive representation of positional information, resulting in improved discrimination between chord types. This integration allows the model to capture both frequency and temporal information more effectively, improving feature extraction and leading to better overall results. However, despite these enhancements, experiments involving positional encoding still fall short of the baseline model, suggesting that the CQT representation already provides sufficient positional and temporal information for effective chord recognition.

VII. Conclusion

This work introduced ChordFormer, a novel Conformer-based architecture for large-vocabulary automatic chord recognition. By integrating convolutional layers with self-attention mechanisms, ChordFormer effectively captures both local spectral features and long-range harmonic dependencies, addressing key challenges in music information retrieval. The proposed model significantly outperforms state-of-the-art methods, demonstrating superior performance in both frame-wise accuracy and class-wise accuracy, particularly in recognizing rare and complex chord types. Evaluation on inversion-aware chord metrics confirmed ChordFormer robustness, while ARWCSR analysis demonstrated its ability to generalize effectively across extended chord types when root information was excluded. Ablation studies offered deeper insight into how each component contributes to overall performance, whereas additional positional encoding provides minimal benefit due to the rich temporal–frequency locality already present in Constant-Q Transform features. These findings position ChordFormer as a scalable, interpretable, and high-performing framework for polyphonic music analysis, while highlighting avenues for improvement particularly in root recognition when both root and quality must be identified, and in designing positional encoding schemes tailored to CQT representations. Future research will explore adaptive re-weighting techniques to further optimize trade-offs between frame-wise and class-wise accuracy. Additionally, integrating self-supervised learning could enhance chord recognition by leveraging unlabeled music data. The findings of this study contribute to the ongoing development of deep learning techniques in music information retrieval, paving the way for more robust and interpretable chord recognition models in real-world applications.

References

[1] G. Durán and P. d. la Cuadra, "Transcribing lead sheet-like chord progressions of jazz recordings," *Comput. Music J.*, vol. 44, no. 4, pp. 26–42, 2020.

[2] J. d. Berardinis, A. Cangelosi, and E. Coutinho, "Measuring the structural complexity of music: From structural segmentations to the automatic evaluation of models for music generation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1963–1976, 2022.

[3] C. Weiß, H. Schreiber, and M. Müller, "Local key estimation in music recordings: A case study across songs, versions, and annotators," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2919–2932, 2020.

[4] X. Du, Z. Wang, X. Liang, H. Liang, B. Zhu, and Z. Ma, "Bytecover3: Accurate cover song identification on short queries," in *Proc. 2023 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[5] J. Pauwels, K. O'Hanlon, E. Gómez, and M. B. Sandler, "20 years of automatic chord recognition from audio," in *Proc. 20th Int. Soc. Music Inf. Retrieval Conf.* Delft, The Netherlands: International Society for Music Information Retrieval, 2019, pp. 54–63. [Online]. Available: https://zenodo.org/record/3527874

[6] J. Jiang, K. Chen, W. Li, and G. Xia, "Large-vocabulary chord transcription via chord structure decomposition," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2019, pp. 644–651.

[7] L. O. Rowe and G. Tzanetakis, "Curriculum learning for imbalanced classification in large vocabulary automatic chord recognition," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2021, pp. 586–593.

[8] J. de Berardinis, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "Choco: A chord corpus and a data transformation workflow for musical harmony knowledge graphs," *Sci. Data*, vol. 10, no. 1, 2023, Art. no. 641.

[9] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6000–6010.

[10] S. Islam et al., "A comprehensive survey on applications of transformers for deep learning tasks," *Expert Syst. Appl.*, vol. 241, 2024, Art. no. 122666.

[11] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," *Interspeech*, pp. 5036–5040, Oct. 2020, doi: 10.21437/interspeech.2020-3015.

[12] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona, "A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends," *Expert Syst. Appl.*, vol. 209, 2022, Art. no. 118190.

[13] S. Sigtia, N. Boulanger-Lewandowski, and S. Dixon, "Audio chord recognition with a hybrid recurrent neural network," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2015, pp. 127–133.

[14] E. J. Humphrey and J. P. Bello, "Rethinking automatic chord recognition with convolutional neural networks," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, 2012, vol. 2, pp. 357–362.

[15] F. Korzeniowski and G. Widmer, "A fully convolutional deep auditory model for musical chord recognition," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process.*, 2016, pp. 1–6.

[16] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 188–194.

[17] Y. Wu and W. Li, "Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF sequence decoding model," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 2, pp. 355–366, Feb. 2019.

[18] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2013, pp. 335–340.

[19] J.-Q. Deng and Y.-K. Kwok, "A hybrid Gaussian-HMM-deep learning approach for automatic chord estimation with very large vocabulary," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 812–818.

[20] J.-Q. Deng and Y.-K. Kwok, "Large vocabulary automatic chord estimation with an even chance training scheme," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 531–536.

[21] E. J. Humphrey and J. P. Bello, "Four timely insights on automatic chord estimation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2015, vol. 10, pp. 673–679.

[22] F. Takuya, "Realtime chord recognition of musical sound: Asystem using common LISP music," in *Proc. Int. Comput. Music Conf.*, Beijing, China, 1999, pp. 464–467.

[23] M. McVicar, R. Santos-Rodrıguez, Y. Ni, and T. D. Bie, "Automatic chord estimation from audio: A review of the state of the art," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 2, pp. 556–575, Feb. 2014.

[24] E. J. Humphrey, T. Cho, and J. P. Bello, "Learning a robust Tonnetz-space transform for automatic chord recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 453–456.

[25] M. Khadkevich and M. Omologo, "Reassigned spectrum-based feature extraction for GMM-based automatic chord recognition," *EURASIP J. Audio, Speech, Music Process.*, vol. 2013, pp. 1–12, 2013.
[26] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. Sagayama, "HMM-based approach for automatic chord detection using refined acoustic features," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2010, pp. 5518–5521.
[27] A. Sheh and D. P. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proc. Int. Symp. Music Inf. Retrieval*, 2003, pp. 185–191.
[28] M. Khadkevich and M. Omologo, "Use of hidden Markov models and factored language models for automatic chord recognition," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2009, pp. 561–566.
[29] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. D. Bie, "An end-to-end machine learning system for harmonic analysis of music," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 6, pp. 1771–1783, Aug. 2012.
[30] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2010, pp. 135–140.
[31] F. Korzeniowski and G. Widmer, "Feature learning for chord recognition: The deep chroma extractor," in *Proc. 17th Int. Soc. Music Inf. Retrieval Conf.*, New York City, U.S., 2016, pp. 37–43.
[32] X. Zhou and A. Lerch, "Chord detection using deep learning," in *Proc. 16th ISMIR Conf.*, 2015, vol. 53, pp. 52–58.
[33] V. Lanz, "Automatic chord recognition in audio recording," Ph.D. dissertation, Univerzita Karlova, Prague, Czechia, 2021.
[34] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A bi-directional transformer for musical chord recognition," in *Proc. 20th Int. Soc. Music Inf. Retrieval Conf.*, Delft, The Netherlands, 2019, pp. 620–627.
[35] K. O'Hanlon and M. B. Sandler, "FifthNet: Structured compact neural networks for automatic chord recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2671–2682, 2021.
[36] F. Korzeniowski and G. Widmer, "On the futility of learning complex frame-level language models for chord recognition," in *IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 91–95.
[37] Y. Wu, T. Carsault, E. Nakamura, and K. Yoshii, "Semi-supervised neural chord estimation based on a variational autoencoder with latent chord labels and features," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2956–2966, 2020.
[38] C. Li, J. Jiang, Y. Li, and L. Tian, "Large-vocabulary chord recognition based on contrastive learning and noisy student (2023)," *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 3695–3706, May 2025.
[39] J. C. Brown, "Calculation of a constant Q spectral transform," *J. Acoustical Soc. Amer.*, vol. 89, no. 1, pp. 425–434, 1991.
[40] A. Poltronieri, V. Presutti, and M. Rocamora, "Chordsync: Conformer-based alignment of chord annotations to music audio," in *Proc. Sound Music Comput. Conf.*, 2024.
[41] Q. Zhang et al., "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7829–7833.
[42] S. Karita et al., "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 449–456.
[43] Q. Wang et al., "Learning deep transformer models for machine translation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1810–1822.
[44] T. Q. Nguyen and J. Salazar, "Transformers without tears: Improving the normalization of self-attention," in *Proc. 16th Int. Conf. Spoken Lang. Transl.*, Hong Kong, 2019. [Online]. Available: https:/aclanthology.org/2019.iwslt-1.17/
[45] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention," in *Proc. 8th Int. Conf. Learn. Representations*, *Addis Ababa, Ethiopia*, 2020, pp. 26–30.
[46] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 933–941.
[47] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.
[48] Y. Lu et al., "Understanding and improving transformer from a multi-particle dynamic system point of view," in *Proc. Workshop Int. Deep Neural Models Differ. Equ.*, 2019.
[49] C. Raffel et al., "MIR_EVAL: A transparent implementation of common MIR metrics," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014, vol. 10, pp. 367–372.
[50] J. Pauwels and G. Peeters, "Evaluating automatically estimated chord sequences," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 749–753.
[51] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. 2018 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2018, pp. 464–468.

Muhamamd Waseem Akram received the bachelor's degree in computer science from the National College of Business Administration & Economics, Lahore, Pakistan, in 2014, and the master's degree in computer science from COMSATS University Islamabad, Islamabad, Pakistan, in 2019. He is currently working toward the Ph.D. degree in emerging digital technology with Scuola Superiore Sant'Anna, Pisa, Italy. His research interests include music and artificial intelligence, and the application of artificial intelligence in steel plant manufacturing.

Stefano Dettori received the master's degree in automation and robotics engineering from the University of Pisa, Pisa, Italy, and the Ph.D degree in emerging digital technologies from Scuola Superiore Sant'Anna, Pisa. He is currently an Assistant Professor with ICT-COISP (TeCIP Institute), Scuola Superiore Sant'Anna. His research interests include time series forecasting, reservoir computing, artificial intelligence applications, model predictive control and optimization, and design of decision support systems for industrial systems.

Valentina Colla (Fellow, IEEE) received the master's degree in telecommunication engineering from the University of Pisa, Pisa, Italy, in 1994, and the Ph.D. degree in industrial and information engineering from Scuola Superiore Sant'Anna, Pisa, in 1998. She has participated in 83 projects funded by the European Union, 14 with the role of Project Coordinator. She is currently an Associate Professor with Scuola Superiore Sant'Anna. She has authored more than 400 papers in her research interests which include Artificial Intelligence-based solutions for modelling, simulation, monitoring, and optimization tasks in industrial contexts, with a particular focus on metallurgical and manufacturing industry.

Giorgio Carlo Buttazzo (Fellow, IEEE) received the degree in electronic engineering from the University of Pisa, Pisa, Italy, the M.S. degree in computer science from the University of Pennsylvania, Philadelphia, PA, USA, and the Ph.D. degree in computer engineering from the Scuola Superiore Sant'Anna, Pisa. He is currently a Full Professor of computer engineering with Scuola Superiore Sant'Anna. He has authored seven books on real-time systems and more than 300 articles in the field of real-time systems, robotics, and neural networks. He was the recipient of the 13 Best Paper Awards. He has been the Editor-in-Chief of *Real-Time Systems* and Associate Editor for *ACM Transactions on Cyber-Physical Systems*.

Open Access provided by 'Scuola Superiore "S.Anna" di Studi Universitari e di Perfezionamento' within the CRUI-CARE Agreement