BACHI: BOUNDARY-AWARE SYMBOLIC CHORD RECOGNITION THROUGH MASKED ITERATIVE DECODING ON POP AND CLASSICAL MUSIC

Mingyang Yao, Ke Chen, Shlomo Dubnov, Taylor Berg-Kirkpatrick

University of California San Diego, USA

ABSTRACT

Automatic chord recognition (ACR) via deep learning models has gradually achieved promising recognition accuracy, yet two key challenges remain. First, prior work has primarily focused on audio-domain ACR, while symbolic music (e.g., score) ACR has received limited attention due to data scarcity. Second, existing methods still overlook strategies that are aligned with human music analytical practices. To address these challenges, we make two contributions: (1) we introduce POP909-CL, an enhanced version of POP909 dataset with tempo-aligned content and human-corrected labels of chords, beats, keys, and time signatures; and (2) We propose BACHI, a symbolic chord recognition model that decomposes the task into different decision steps, namely boundary detection and iterative ranking of chord root, quality, and bass (inversion). This mechanism mirrors the human ear-training practices. Experiments demonstrate that BACHI achieves state-of-the-art chord recognition performance on both classical and pop music benchmarks, with ablation studies validating the effectiveness of each module.

***Index Terms*—** Symbolic Chord Recognition, Iterative Decoding, POP909 Annotation, Music Information Retrieval

1. INTRODUCTION

Automatic chord recognition (ACR) is a fundamental task in music information retrieval that aims to annotate music with chord labels over time. Recent breakthroughs in machine learning have shown that ACR models with deep neural networks outperform conventional approaches. Chord recognition also underpins a wide range of downstream applications, including harmonic analysis, annotation for controllable music generation, and music education.

Despite recent promising progress, symbolic chord recognition remains underexplored compared to its development in the audio domain, largely due to data scarcity. Audio-based ACR models benefit from well-established benchmarks in pop music, such as USPOP [1], RWC-Popular [2], Billboard [3], Isophonics [4], and CASD [5]. Although benchmarks for classical music are fewer, models trained on pop music have captured many learnable patterns and demonstrated broad applications in music analysis and music generation tasks.

In contrast, symbolic ACR faces the opposite situation and a more severe imbalance. Very few datasets provide accurate chord annotations except for Hooktheory [6], but it only includes chord labels and melody without full arrangements or textures, limiting its usability for ACR. For classical music, most chord-annotated symbolic datasets derive from the Roman Numeral Harmonic Analysis task. Among these, When-in-Rome [7] and the DCML corpus [8] provide the largest collections (about 1700 works in total), while others, such as the Beethoven Sonata corpus [9], TAVERN [10], and BPS [11] contain fewer than 100 works each. However, these corpora often include duplicated pieces or short excerpts, and existing approaches could only rely on subsets or combined chunks for ACR training. For example, ChordGNN [12] and AugmentedNet [13] use 300+ works (1400+ segments), while Harmony Transformer [14, 15] uses fewer than 100 works, since it outputs Roman numeral analyses rather than chord labels alone.

Another major challenge of symbolic ACR lies in methodology. Prior approaches have adopted diverse strategies to improve performance. The rule-based method [16] aggregates notes within windows to infer chords but often fails with complex harmonic progressions due to their limited reasoning capability. AugmentedNet [13] and ChordGNN [12] model structure note interactions using either convolutional or graph neural networks to strengthen the note–chord correlation in recognition. Harmony Transformer [14, 15] focuses more on the Roman numeral harmonic analysis task and introduces transformer architectures with global-context modeling, bridging long-range note dependencies. However, most approaches pay little attention to aligning the chord recognition process with the human annotation process, which is a potential factor that may be crucial for improving accuracy. Some audio-based ACR models have explored this idea. For exampe, [17] decomposes chord targets into constituent elements (root, triad, 7th) to guide recognition. Inspired by this, transferring domain knowledge from human annotation logic offers a promising direction for improving performance.

In this paper, we address the challenges of data scarcity and methodology in symbolic chord recognition. First, we propose POP909-CL, an enhanced version of the POP909 dataset [18] with human-corrected labels for chords, beats, keys, and time signatures. This resource supports not only symbolic chord recognition but also broader symbolic MIR tasks. Second, we propose BACHI, a **B**oundary-**A**ware symbolic **CH**ord recognition model with masked **I**terative decoding. It operates on beat-synchronous MIDI tokens and integrates two key components: (1) a supervised boundary detection module that predicts chord-change likelihoods and modulates encoder states via feature-wise linear modulation [19] (FiLM); and (2) a transformer decoder [20] that iteratively predicts chord root, quality, and bass in confidence order. This design mirrors human sight-singing and ear-training practices, where chord perception emerges progressively through cues whose order depends on different contexts. Our contributions are three-fold:

- We proposed POP909-CL, an enhanced version of POP909 with human-corrected labels, as a reliable resource for MIR research.
- We propose BACHI, a boundary-aware symbolic chord recognition model that incorporates cues and decision processes inspired by human ear-training to improve recognition accuracy.
- Objective evaluations demonstrate that BACHI achieves state-of-the-art performance on both classical and pop music benchmarks, with ablation studies validating the effectiveness of each module.

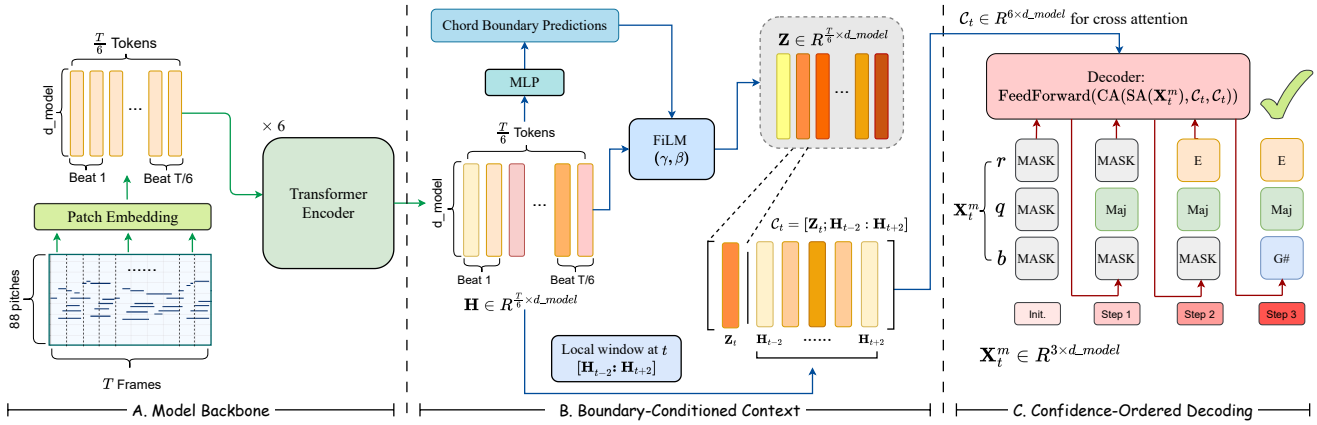

Fig. 1. The model architecture and inference mechanism of BACHI, from the model backbone (left and middle), boundary detection and conditioning (middle), and iterative decoding (right).

2. METHOD

Figure 1 illustrates the model architecture and inference mechanism of BACHI. In the following subsections, we will introduce its input and output specifications and the different recognition stages.

2.1. Input and Output

For input, BACHI encodes symbolic musical scores as piano rolls $\mathbf{P} \in \{0,1\}^{T\times D}$, where $D = 88$ is the dimensionality of pitch classes and the temporal resolution for T is 12 frames per beat. We employ a patch embedding module to convert piano rolls into continuous latent tokens. Specifically, the patch embedding module contains a 1D-CNN layer with a kernel size 6 to map the pitch class channel D to the hidden dimension of the transformer $d\_model = 512$, and reduce the temporal dimension from T to $T/6$. It is followed with a GLU activation layer to create a more normalized and compact representation.

For output and groundtruth labels, we decompose each chord label into three elements: root, quality, and bass. Bass is referred to the bass note for chord inversion (e.g., C/G is a C major chord inversion with C root, major quality, and G bass). The label sequence is processed with the same temporal resolution as input ($T/6$).

2.2. Two-stage Chord Recognition

After processing the input and output sequences, we adopt a two-stage chord recognition framework: (1) boundary detection, and (2) iterative chord element decoding using confidence ranking with masked transformers.

Boundary Detection The input music sequence is fed into six transformer encoder blocks to produce the hidden state sequence output $\mathbf{H}$. An MLP layer is then applied to predict the chord boundary sequence $\mathbf{e}$, where boundary labels are obtained as binarized labels derived from the chord labels. The predicted boundaries are further embedded as an additional condition for the subsequent chord recognition step via feature-wise linear modulation [19] (FiLM):

$$[\boldsymbol{\gamma}_t, \boldsymbol{\beta}_t] = [\mathrm{MLP}_{\gamma_t}(\mathrm{LN}([\mathbf{H}_t; \mathbf{e}_t])), \mathrm{MLP}_{\beta_t}(\mathrm{LN}([\mathbf{H}_t; \mathbf{e}_t]))] \quad (1)$$

$$\mathbf{Z}_t = \mathrm{LN}(\mathbf{H}_t) \odot (1+\boldsymbol{\gamma}_t) + \boldsymbol{\beta}_t. \quad (2)$$

Where γ and β denote the scale and the bias from FiLM via two MLP layers and LayerNorm [21]. $\mathbf{Z}$ denotes the FiLM-conditioned latent representation from $\mathbf{H}$ and boundary predictions $\mathbf{e}$.

Iterative Decoding by Confidence The framewise representation $\mathbf{Z}_t$ is combined with a local context window $r = 2$ to form a new sequence $\mathcal{C}_t$=$[\mathbf{Z}_t, \mathbf{H}_{t-2:t+2}] \in R^{6\times d\_model}$, which aggregates boundary cues, neighboring context, and the chord features at frame t.

The final decoder of BACHI is a single-layer transformer decoder block consisting of a self-attention module and a cross-attention module. Its output is a framewise chord-element sequence $\mathbf{X}_t \in R^{3\times d\_model}$. During training, we adopt the masked transformer paradigm, where each decoder input $\mathbf{X}_t^m$ is a randomly masked version of $\mathbf{X}_t$. The model is optimized to fill the masked elements. The decoder leverages self-attention over $\mathbf{X}_t^m$ and cross-attention over $\mathcal{C}_t$, thereby integrating information from both existing chord elements (r, q, b), and the local context encoded in $\mathcal{C}_t$:

$$\mathbf{X}_t \leftarrow \mathrm{FeedForward}(\mathrm{CA}(\mathrm{SA}(\mathbf{X}_t^m), \mathcal{C}_t, \mathcal{C}_t)) \quad (3)$$

where $\mathrm{CA}(\mathrm{Q}, \mathrm{K}, \mathrm{V})$ denotes the cross attention computation. Then the logits ℓ_t^s are obtained for each chord element via separate classification heads. Note that the decoder **does not** auto-regressively predicts the output, as all masked fields are predicted simultaneously at one time.

At inference, we follow the below steps to obtain the final chord prediction:

1. initialize $\mathbf{X}_t^m$ to all `[MASK]` and send to the decoder.
2. Compute confidences of the output $c_t^s = \max(\mathrm{softmax}(\ell_t^s))$ for each unfilled $s \in \{q, r, b\}$.
3. Commit the highest-confidence element prediction.
4. Repeat until all components are filled (three iterations in total).

This results in a simple, order-agnostic procedure that adapts the prediction order to the data and mirrors human ear-training practices, by first identifying the most salient element and then progressively resolving the remaining ones.

We train the decoder jointly with the other components as an end-to-end model. Since the input to the beginning transformer encoder has the different size of the input to the final decoder, in code implementation, we conduct the reshaping along the dimensions of batch size and contextual length to support the end-to-end training.

Model / Approach	Accuracy (*per piece macro (%)*)							
	Classical Corpus				Pop909-CL			
	Root	**Quality**	**Bass**	**Full**	**Root**	**Quality**	**Bass**	**Full**
Rule-based [16]	54.6	45.8	50.5	28.4	85.9	69.7	85.8	65.0
AugmentedNet [13]	73.9	74.2	72.3	57.2	88.6	84.5	90.5	78.7
ChordGNN [12]	73.0	73.7	71.0	58.5	80.7	82.0	82.7	71.6
Harmony Transformer v2 [15]	75.9	75.7	74.9	60.3	**90.6**	86.9	**92.2**	82.2
BACHI (ours)	**78.3**	**79.7**	**77.5**	**69.0**	89.6	**87.1**	91.5	**82.8**

Table 1. Model performance on classical Corpus (DCML and WiR) and POP909-CL. Accuracies are reported *per piece macro (%)*.

3. EXPERIMENTS

3.1. Dataset and Training Setup

Classical Music We construct a classical corpus by combining When-in-Rome (WiR) [7] and DCML [8] functional-harmony repositories and make the de-duplications. Since both datasets focus on a harmonic analysis task instead of chord recognition, we convert their annotations into the absolute chord labels via *music21* [22] package and a self-written conversion script of chord quality.

POP909-CL The original POP909 dataset [18] consists of piano arrangements of 909 Chinese pop songs in MIDI format. Although it provides the extracted beat, chord, and key annotations, many of these contain errors due to the limitations of the rule-based extraction algorithms. In addition, the tempo varies within each sample, preventing direct conversion into score-aligned symbolic data with a fixed tempo.

To address these limitations, we introduce POP909-CL, an enhanceedd version of POP909 with **C**orrect **L**abels of chords, beats, keys, and time signatures across all 909 tracks. We recruit professional musicians to refine the annotations: starting from the rule-based extraction labels, they carefully reviewed each track and corrected erroneous labels. During this process, they also provided comments on special cases (e.g., the presence of weak attack bars), which we manually resolved to ensure consistency and accuracy.

A statistical comparison between POP909 and POP909-CL is shown below: 40.6% of start beats in POP909 are misaligned, 14.2% of key signature changes are missing, and 2.6% of time signatures are incorrect. And in POP909-CL, we correct all of them. Figure 2 provides an example of human-corrected labels versus algorithmic extraction. These refinements establish POP909-CL as a reliable pop music dataset with corrected annotations for both analytical and generative tasks. We will release POP909-CL to the music community.

Training Specification We collect 1500 unique classical and 909 pop music pieces with accurate chord labels for the training. Since we collect the correct beat labels in POP909-CL, we aligned all POP909 samples with a fixed tempo to become a music score version for ACR training. We split each dataset using 9:1 train-test splits, apply 12-key augmentation for the training sets, and only use the canonical version in the test set. We do not train a single with both classical and pop music data but two separate models for each, because the data distribution and the chord pattern are largely deviated through these two genres as described in Section 1.

Training Setting We optimize with AdamW [23] (β_1=0.9, β_2=0.98, eps=10^{-9}), and apply linear warm-up for 4000 steps (classical) and 2000 steps (POP909-CL), followed by cosine decay on the learning rate range (1e-5, 1e-4). We employ the mixed-precision training in bfloat16 and set the maximum gradient clip norm to 2.0.

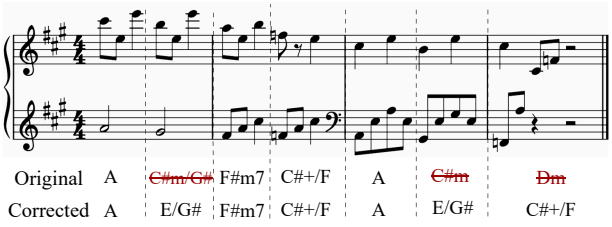

Fig. 2. The chord label comparison between the rule-based extraction (original in POP909) to human corrected ones (in POP909-CL).

3.2. Evaluation, Ablations and Baselines

We evaluate predictions using macro-accuracy over chord elements (root, quality, and bass) as well as overall chord accuracy. As baselines, we select AugmentedNet [13], ChordGNN [12], and Harmony Transformer v2 [15]. We retrain all three models on the same classical and pop datasets used for BACHI to ensure fair comparison. For Harmony Transformer v2, we modify the output format from a single target to three targets (r, q, b) to match our setup. In addition, we include the training-free rule-based method [16] as a lower-bound anchor, extracting root, quality, and bass from its chord predictions, since the chord label is its only output.

We also conduct ablation studies on four variants: (1) BACHI without boundary detection and iterative decoding, directly projecting encoder outputs to chord labels; (2) BACHI with hard boundary constraints; (3) BACHI conditioned on additional key detection, implemented by training an auxiliary MLP for key prediction and embedding its output via FiLM together with boundary information; and (4) the full BACHI model. Due to page limitations, we report results on classical datasets in the main text, with POP909-CL results provided in the supplementary website.[1].

4. RESULTS

4.1. Comprehensive Performance

Table 2.2 presents results across both classical and pop music evaluation sets. For classical music, BACHI achieves the highest scores across all metrics, with a notable improvement in full chord accuracy (69%) over prior baselines. Nevertheless, chord recognition in classical music remains highly challenging, as all models perform the full chord accuracy below 70%. This reflects the greater harmonic complexity and stylistic biases across composers and periods, highlighting the need for future work to better address these challenges.

In contrast, most models achieve above 75% full chord accuracy on pop music, and the gaps among them are smaller. BACHI attains the best results in full chord and chord quality accuracy, and ranks

[1] https://andyweasley2004.github.io/BACHI/

Model Design	Accuracy (*per piece macro (%)*)			
	Root	**Quality**	**Bass**	**Full Chord**
BACHI w/o. **BD** and **ID**	**78.4**	79.5	**77.6**	66.8
BACHI w/o. **ID**	76.6	78.1	76.1	67.1
BACHI w/. key detection	77.4	78.8	76.4	67.6
BACHI	78.3	**79.7**	77.5	**69.0**

Table 2. Ablation study on our BACHI variants. The accuracy metrics are reported *per piece macro (%)*.

second in root and bass accuracy, following Harmony Transformer v2 [15]. We also observe that the rule-based method used in the original POP909 dataset achieves only 65% chord accuracy. This underscores the value of POP909-CL, which corrects approximately 35% of chord label errors in POP909.

Confusion matrices in Figure 3 further illustrate the difference between pop and classical music chord patterns. On POP909-CL, most errors involve confusion between closely related qualities (e.g., major vs. minor), while in classical music, misclassifications are distributed more broadly across qualities. These patterns confirm that pop harmony is relatively predictable and concentrated, whereas classical harmony exhibits greater variability and annotation ambiguity.

Overall, these results demonstrate both the strong performance of BACHI on classical and pop music and the contribution of POP909-CL as a reliable resource for symbolic ACR. Overall, our approach is particularly effective on challenging classical repertoire while remaining competitive on popular music, demonstrating robustness across diverse harmonic contexts.

4.2. Ablation Study

Table 4.1 presents ablation results of BACHI variants on the classical corpus, where **BD** denotes boundary detection and **ID** denotes iterative decoding. Without BD and ID, BACHI reduces to a basic transformer encoder model, achieving reasonable individual component performance (78.4% root, 79.5% quality, 77.6% bass) but lower full chord accuracy (66.8%).

Including BD alone improves results, but the absence of iterative decoding still limits performance, indicating that iterative decoding across chord elements contributes significantly to overall accuracy. Interestingly, adding key detection as an additional conditioning signal slightly decreases full-chord accuracy compared to the full BACHI, likely due to errors in key prediction propagating to chord recognition.

These findings demonstrate that the combination of boundary detection and iterative decoding substantially enhances chord recognition, while adding extra conditions, such as key, does not necessarily improve performance due to potential error accumulation in auxiliary detection tasks.

5. DISCUSSION

Our confidence-guided decoding reveals striking repertoire-specific patterns that validate our human ear-training practices. In classical pieces, the model tends to predict quality first (with the ratio 41.83%). The most frequent prediction chain is quality→root→bass (33.98%), matching analysis that infers chord type from voice-leading. In POP909-CL, the model tends to predict bass first (65.77%), as the most frequent chain is bass→root→quality (56.70%).

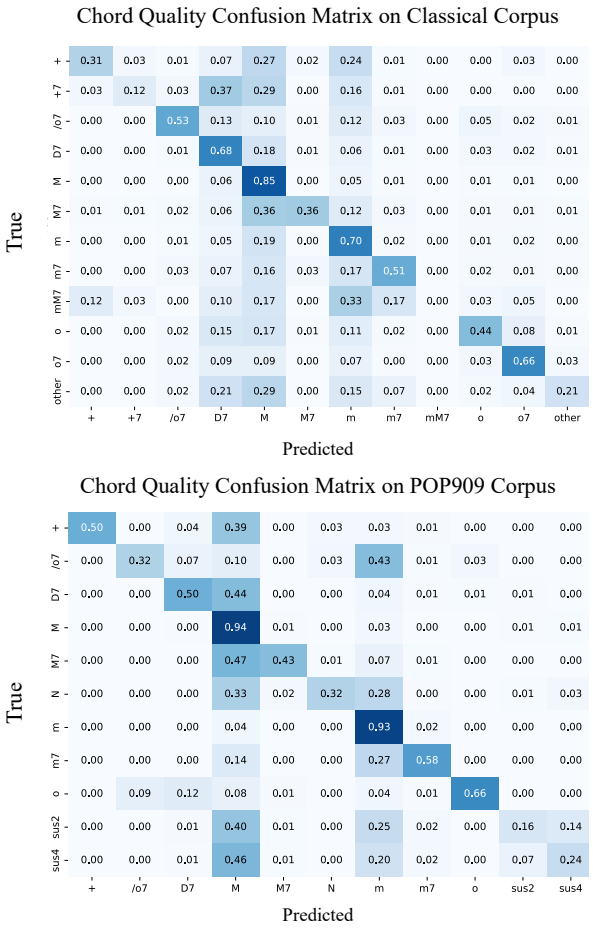

Fig. 3. Confusion matrices on chord quality in classical corpus and POP909-CL evaluation sets.

This is consistent with bass-led cues in pop. These genre-variant orders indicate that the model internalizes musician-like heuristics, supporting our hypothesis that human-mimicking decision paths benefit symbolic ACR and improve upon fixed-order decoding.

6. CONCLUSION

In this research, we presented BACHI, a boundary-aware transformer for symbolic ACR, combining patch embedding, FiLM-based boundary conditioning, and confidence-guided masked-filling decoding. Our model achieves substantial improvements over prior baselines, and we contribute new human-annotated chord labels for the POP909-CL dataset to support evaluation and generation tasks. Analysis further shows that confidence-ordered decoding adapts to genre-specific patterns, with classical music favoring quality-first prediction and popular music favoring bass-first prediction, remarking the value of flexible decoding strategies under various uncertainty levels in multiple tasks.

Our boundary-aware framework and POP909-CL annotations provide a foundation for advancing generative systems, enabling more harmonically coherent and structurally aware generation under chord conditions. Beyond applications in generation, learned confidence patterns also offer new insights into music theory by highlighting repertoire-specific harmonic tendencies. We will release our code, trained models, and POP909-CL annotations soon.

7. REFERENCES

[1] Daniel P. W. Ellis, Adam Berenzweig, and Brian Whitman, "The "uspop2002" Pop Music Data Set," 2003.

[2] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, "Rwc music database: Popular, classical, and jazz music databases," in *Proc. ISMIR*, 2002.

[3] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga, "An expert ground-truth set for audio chord recognition and music analysis," in *Proc. ISMIR*, 2011.

[4] Isophonics / Centre for Digital Music (Queen Mary University of London), "Isophonics reference annotations (beatles, queen, Zweieck, *etc.*)," 2005.

[5] Hendrik Vincent Koops, W. Bas de Haas, and Anja Volk, "Annotator subjectivity in harmony annotations of popular music," *Journal of New Music Research*, 2019.

[6] "HookTheory," `https://www.hooktheory.com/` [Accessed: (September 1, 2023)].

[7] Mark R H Gotham, Rainer Kleinertz, Christof Weiss, Meinard Müller, and Stephanie Klauk, "What if the 'when' implies the 'what'?: Human harmonic analysis datasets clarify the relative role of the separate steps in automatic tonal analysis," in *Proc. ISMIR*, 2021.

[8] DCMLab, "Dcmlab github repositories," Accessed: 2025-09-15.

[9] Johannes Hentschel, Yannis Rammos, Markus Neuwirth, and Martin Rohrmeier, "Ludwig van beethoven – piano sonatas (a corpus of annotated scores)," 2025.

[10] Johanna Devaney, Colton Arthur, Nathaniel Condit-Schultz, and Kariin Nisula, "Theme and variation encodings with roman numerals (tavern): A new data set for symbolic music analysis," in *Proc. ISMIR*, 2015.

[11] Tzu-Pei Chen and Li Su, "Functional harmony recognition of symbolic music data with multi-task rnn," in *Proc. ISMIR*, 2018.

[12] Emmanouil Karystinaios and Gerhard Widmer, "Roman numeral analysis with graph neural networks: Onset-wise predictions from note-wise features," in *Proc. ISMIR*, 2023.

[13] Néstor Nápoles López, Mark Gotham, and Ichiro Fujinaga, "AugmentedNet: A Roman Numeral Analysis Network with Synthetic Training Examples and Additional Tonal Tasks," in *Proc. ISMIR*, 2021.

[14] Tsung-Ping Chen and Li Su, "Harmony transformer: Incorporating chord segmentation into harmony recognition," in *Proc. ISMIR*, 2019.

[15] Tsung-Ping Chen and Li Su, "Attend to chords: Improving harmonic analysis of symbolic music using transformer-based models," in *Trans. Int. Soc. Music. Inf. Retr.*, 2021.

[16] Shuqi Dai, Huan Zhang, and Roger B. Dannenberg, "Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music," *CoRR*, vol. abs/2010.07518, 2020.

[17] Junyan Jiang, Ke Chen, and Wei Li andGus Xia, "Large-vocabulary chord transcription via chord structure decomposition," in *Proc. ISMIR*, 2019.

[18] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, and Gus Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proc. ISMIR*, 2020.

[19] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. AAAI*, 2018.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017.

[21] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[22] Michael Scott Cuthbert and Christopher Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," in *Proc. ISMIR*, 2010.

[23] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.