HARMONIC ANALYSIS WITH NEURAL SEMI-CRF

Qiaoyu Yang **Frank Cwitkowitz** **Zhiyao Duan**
University of Rochester
{qyang15, fcwitkow}@ur.rochester.edu, zhiyao.duan@rochester.edu

ABSTRACT

Automatic harmonic analysis of symbolic music is an important and useful task for both composers and listeners. The task consists of two components: recognizing harmony labels and finding their time boundaries. Most of the previous attempts focused on the first component, while time boundaries were rarely modeled explicitly. Lack of boundary modeling in the objective function could lead to segmentation errors. In this paper, we introduce a novel approach named *Harana*, to jointly detect the labels and boundaries of harmonic regions using neural semi-CRF (conditional random field). In contrast to rule-based scores used in traditional semi-CRF, a neural score function is proposed to incorporate features with more representational power. To improve the robustness of the model to imperfect harmony profiles, we design an additional score component to penalize the match between the candidate harmony label and the absent notes in the music. Quantitative results from our experiments demonstrate that the proposed approach improves segmentation quality as well as frame-level accuracy compared to previous methods. The source code used in this paper is available on GitHub [1] .

1. INTRODUCTION

In music, harmony is the sound resulted from two or more pitches being performed together. It is the vertical aspect of music [1], and is essential for both music creation and perception. During music analysis, a harmony label is often assigned to a music segment that is harmonically coherent. Many composers use harmonic progressions to set up a musical template in which texture could then be filled [2]. For listeners, harmonic structure is a crucial mid-level representation of music that can influence the perception of other music elements such as melody and rhythm [3].

The task of harmonic analysis aims to find the correct segmentation of a music piece and to identify the corresponding label for each segmented region. These two goals are closely related. Regions with strong confidence of a candidate harmony label tend to possess the boundaries of a true segmentation [4]. On the other hand, the oracle segmentation could help the prediction of the true underlying harmony for the notes in each region [4]. Therefore, to achieve successful analysis of harmony, both of the two goals as well as their relationship should be considered.

Targeting the two indispensable components of harmonic analysis simultaneously, we propose an approach to jointly predict the boundaries and labels of harmonic regions using neural semi-Markov conditional random field (semi-CRF). It is well-known that the harmonic regions in music do not always share the same length [5]. Compared to conventional sequence labeling models, semi-CRF is more suitable for the task because it allows for various lengths among the labeled regions [6].

In the original setting of semi-CRF, a score is computed in each segmented region using the weighted sum of rule-based features [6]. However, rule-based features are bounded by pre-defined rules and might not exploit the interaction between notes and other intermediate music representations deeply enough. To solve this problem, we design a neural scoring function that first estimates the frame-level harmony distributions using a neural network and then adapts them to candidate harmony labels with an attention mechanism. The attention mechanism could make the scoring module more efficient by concentrating on sub-regions that are more harmonically related to the candidate label. In addition, an absence score is added to the scoring function to improve the robustness of the model to imperfect harmony profiles of the music. Through experiments we find that the proposed architectural components collectively yield improvement on both segmentation quality and harmony labels accuracy. We focus on MIDI-like symbolic music input in our experiments but the method could be easily adapted to audio.

In summary, our contributions include:

- Proposing the first neural semi-CRF model to jointly estimate harmony labels and their time boundaries;
- Proposing an attention-based score function to alleviate the influence of extra non-chordal notes and missing chordal notes; and
- Proposing a novel absence score to improve the robustness to imperfect harmony profiles.

2. RELATED WORKS

Due to the importance of harmony in music, a substantial amount of automatic systems have been designed for har-

[1] https://github.com/QiaoyuYang/harana

monic analysis. Early systems tended to focus on using music audio as input and apply domain knowledge from music theory. To encode the audio waveform, a time-frequency representation, or spectrogram, is usually extracted using the short-time Fourier Transform. Then, with the observation that it is the pitch class of notes rather than the absolute pitch height that affects the harmonic content, a common practice is to reduce the spectrogram to a chromagram with 12 bins corresponding to the 12 pitch classes. In the decoding stage, the chromagram can be matched to predefined chord profiles [7, 8] or made to emit explicit labels using probabilistic models such as hidden Markov model (HMM) [9–11] or CRF [11].

With the increasing popularity of deep learning in the past decade, end-to-end models based on deep neural networks have received extensive attention [12–16]. To model the temporal evolution of music context, Boulanger-Lewandowsk et al. extracted audio features using a recurrent neural network (RNN) [17]. To better aggregate context information and learn intermediate representations with a temporal hierarchy, Zhou and Lerch used a convolutional neural network (CNN) with low-pass filters [12]. McFee and Bello further combined CNN and RNN in the feature encoder for chord recognition [13]. As a powerful attention-based architecture designed for long-term sequence modeling, transformers have also been incorporated in some recent approaches to harmonic analysis [14, 15].

While the harmonic progression or context information can be modeled with various techniques, the majority of existing methods do not directly optimize for region-level output. Some methods adopt a two-stage approach, where the first stage outputs frame-level chord labels and the second stage smooths frame-level labels with post-processing [9–11, 18–20]. However, different from other simple sequence labeling tasks such as part-of-speech tagging, a harmonic label could correspond to a region spanning multiple frames. Although temporal smoothing by HMM or CRF regresses some sporadically outliers back to the harmonic streams, these models could still suffer from segmentation errors. Masada and Bunescu relaxed the constraint on fixed-size time-span of the output prediction [21]. They used a generalized variant of CRF, semi-CRF, to jointly detect chord labels and their boundaries. However, the features to the semi-CRF are entirely rule-based, which means they are not necessarily optimal for the end task. In this work, we build on the semi-CRF framework and explore neural features and scoring techniques that are jointly optimized for the end task - harmony labeling and boundary prediction.

3. METHODS

In our proposed model, Harana, we first estimate the harmony (including root, quality, and pitch activation in this work) at the frame-level; then we aggregate the frame-level estimation into region-level segment scores based on candidate segments; finally, we use semi-CRF to find the best segmentation candidate and its corresponding labels. We focus on symbolic music input in our experiments. The following subsections describe the model in detail.

3.1 Data Representation

3.1.1 Symbolic Music Input

Given a symbolic music piece, we slice it into short frames of one eighth of a beat long. We use beat instead of note duration in order to represent the basic time unit because music with different meters may have different distributions on the note length. The pitch information in each frame is summarized with a 12-d pitch class distribution vector, which describes the normalized distribution of the duration of each pitch class in the frame. To help distinguish between harmonies with the same pitch class vector, we also include the bass note (the lowest note) in the input to the model; it is represented as a 12-d one-hot vector indicating the bass pitch class in each frame. Combining the pitch class distribution and the bass note, the input to the model is a sequence of 24-d vectors.

3.1.2 Harmony

A popular representation of music harmony in symbolic music is the Roman numeral encoding, where the full harmonic context of a label, including tonic and degree, is considered [22]. However, the combination of all the components produces 47k different harmony labels, which are intractable for a classification model with limited training data. A possible solution is to classify each harmony component independently, but this is incompatible with semi-CRF because the boundary of each component must be the same. As a compromise, we use a subset of the harmony components, root and quality, and model them jointly.

The root is represented as a 12-d one-hot vector corresponding to the 12 pitch classes. The quality is represented as a 10-d one-hot vector corresponding to 10 commonly used classes. In addition to root and quality, we use another harmony representation, the pitch class activation vector, in the neural score function. Previous works have shown its effectiveness as a label encoding for harmonic analysis [13]. These vectors are 12-d multi-hot and are circularly shifted from the pitch-class activation vectors rooted at C.

3.2 Semi-CRF

Semi-CRF is a probabilistic model for sequence labeling with a variable label-span. Given a sequence of input frames $X = \langle X_1, X_2, ..., X_N \rangle$, semi-CRF provides the conditional probability of the sequence of contiguous non-overlapping labeled segments $Y = \langle Y_1, Y_2, ..., Y_K \rangle$, where N is the number of frames and K is the number of segments. Since the labeled segments could span multiple frames, they are represented as three-dimensional tuples $Y_i = (u_i, v_i, l_i)$, where u_i, v_i and l_i respectively denote the onset, offset and label of the segment. In the context of harmonic analysis, X represents the input music frames and Y represents the harmonic regions.

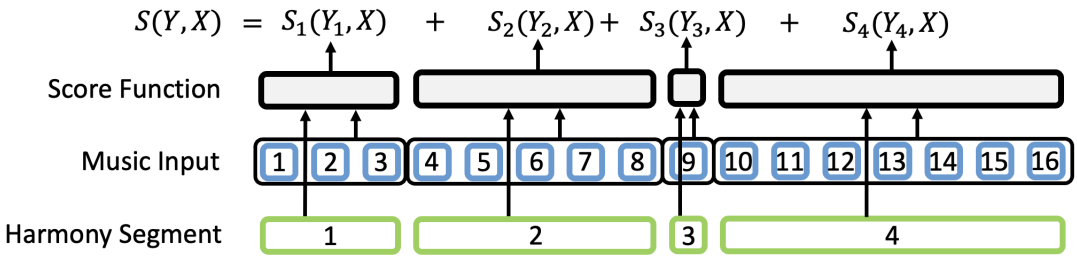

Figure 1: The semi-CRF architecture in the context of harmonic analysis. The total score is computed from music input and a set of candidate harmony segments. Numbers in the blue squares are the frame indices. Numbers in the green rectangles are the indices of candidate harmony segments.

The conditional probability given by semi-CRF takes the form of

$$P(Y|X) = \frac{e^{WF(Y,X)}}{Z(X)}, \tag{1}$$

where F is a feature vector computed from X and Y, W is a learnable weight matrix, and $Z = \sum_Y e^{WF(Y,X)}$ is a normalization factor summarizing all possible segmentation and labeling of the input sequence. In this work, we propose to generalize the weighted feature score to a neural score function $S(Y, X)$ so that

$$P(Y|X) = \frac{e^{S(Y,X)}}{Z(X)}. \tag{2}$$

With the assumption that the harmony labels are Markovian given the music input, the score function could be decomposed into the sum of segment-level scores that are dependent only on the current and the previous segments.

$$S(Y, X) = \sum_{i=1}^{K} S_i(Y_i, X; Y_{i-1}). \tag{3}$$

To simplify the notation, we treat Y_{i-1} as a parameter for the i-th segment's score function and omit it in the following sections. Figure 1 demonstrates the structure of semi-CRF in the context of music harmonic analysis.

3.3 Frame-Level Estimation

Followng Micci et al. [23], the frame-level estimation of harmony information is achieved with a DenseNet-GRU architecture. The DenseNet-GRU module is followed by fully connected layers and finally the vectors corresponding to different types of harmony information are estimated using separate linear heads. The softmax function is used to produce the class distributions of the root and the quality, whereas sigmoid is used to find the activation of each pitch class. Mathematically, the computation of frame-level harmony estimation can be formulated as

$$\begin{aligned} E(n) &= MLP(GRU(DenseNet(X_n))), \\ \hat{D}_R(n) &= \text{Softmax}(FC_R(E(n))), \\ \hat{D}_Q(n) &= \text{Softmax}(FC_Q(E(n))), \\ \hat{PC}(n) &= \text{Sigmoid}(FC_{PC}(E(n))), \end{aligned} \tag{4}$$

where X_n is the n^{th} frame of the input music. $\hat{D}_R(n)$, $\hat{D}_Q(n)$ and $\hat{PC}(n)$ represent the root distribution, quality distribution and the pitch class activations of the estimated harmony for a frame.

3.4 Attention-Based Score Function

As described in Eq. (3), the CRF model evaluates possible sequences of harmony labels and their segmentation. For each segment, i.e., a candidate harmony region, we need to aggregate the frame-level harmony information (root, quality and pitch activation) computed from Eq. (4). A simple method would be taking the average or the mode, but we note that a harmonic region is not likely to contain homogeneous harmonic content. In order to dynamically weigh the harmonic importance of each frame within a region, an attention module is proposed to focus on the frames that are most similar to the candidate harmony label. In particular, the scaled dot-product attention [24] is used:

$$A(Q, K, V) = \frac{\sum_{i=1}^{N} Q^T K_i V_i}{\sqrt{d}}, \tag{5}$$

where Q is the query vector, K is the key sequence, V is the value sequence and d is the vector size.

In the context of our model, the estimated frame-level harmony sequence of a candidate harmony region serves as both the key and value while the candidate region-level harmony itself is the query. Then, the candidate-informed (CI) estimation can be computed as

$$\hat{H}_{CI}(Y_i) = A(H(l_i), \hat{H}(u_i : v_i), \hat{H}(u_i : v_i)), \tag{6}$$

where l_i is the i-th candidate harmony label, and $H(l_i)$ is its harmony representation, which can be root D_R, quality D_Q or pitch class activation PC as defined in Eq. (4). Variables u_i and v_i are the first and last frames of the i^{th} harmonic region Y_i, and $\hat{H}(u_i : v_i)$ is the vector sequence of estimated frame-level harmony representations from the music input.

Now that we have a single embedding vector to summarize the harmonic content in the i-th candidate region, the score of assigning the candidate harmony label l_i to this region can be described by the similarity between the candidate harmony label embedding $H(l_i)$ and the candidate-informed music embedding $\hat{H}_{CI}(Y_i)$. Dot product is used

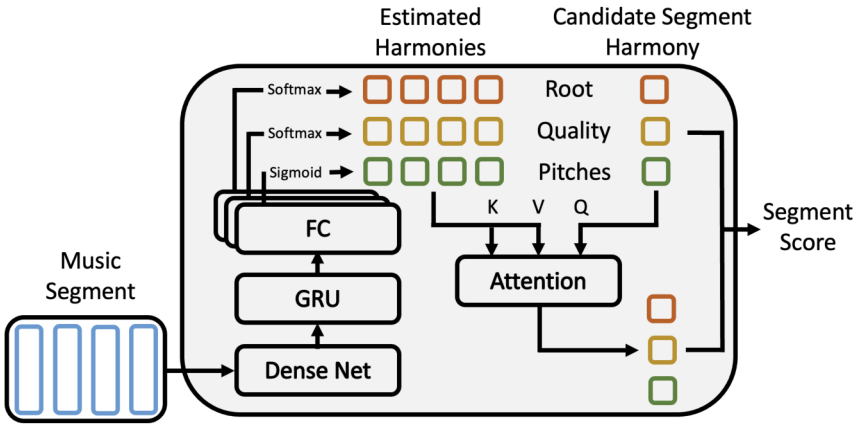

Figure 2: The proposed pipeline of the neural encoder and scoring function.

to calculate the similarity:

$$S_i^H(Y_i, X) = H(l_i)^T \hat{H}_{CI}(Y_i). \tag{7}$$

To further model the transition probability between adjacent harmony labels and enforce more inductive bias in decoding, a transition score between segments is computed:

$$S_i^T(Y_i) = T[l_{i-1}, l_i] + (v_i - u_i)T[l_i, l_i], \tag{8}$$

where T is the transition matrix containing log-probabilities of harmony transitions at the frame level. It is pre-computed from the ground-truth labels in the training data.

Combining the similarity score and the transition score, the score function of a candidate harmony region is

$$S_i(Y_i, X) = \sum_H S_i^H(Y_i, X) + \lambda S_i^T(Y_i), \tag{9}$$

where λ is a hyperparameter to balance the two score components. Figure 2 illustrates the overall structure of the neural front end and the scoring function.

3.5 Absence Score

In Eq. (7), the comparison between the candidate-informed music embedding $\hat{H}_{CI}$ with the candidate harmony representation H indicates the likelihood of the candidate harmony. However, this comparison may not be robust when there are many non-chordal notes or missing chordal notes in the estimation. In this case, the estimated class distributions $\hat{D}_R$ and $\hat{D}_Q$ in Eq. (4) would be relatively flat and the pitch class activation vector $\hat{PC}$ would not align well with a chord template. In other words, the neural front-end may not sufficiently suppress non-chordal notes and recognize missing chordal notes to produce class distributions discriminative enough for the semi-CRF to decode the harmony. To improve the robustness of the model to such issues, we introduce an *absence score* to allow the model to filter out pitch activations that are not active within the input music, the majority of which represent non-chordal notes that should not intersect with chordal notes of the underlying harmony. To compute the absence score, the complement of the input pitch class vector is sent to the neural front-end. That means the input to Eq. (4) is transformed by

$$X_n[1:12] = 1 - X_n[1:12]. \tag{10}$$

The harmony information estimated from the inactive music $\hat{H}^{inact}$ are then compared with the candidate harmony vectors $H(l_i)$. The similarity between them should be minimized. In summary, the absence score of a candidate harmony region is

$$AS_i^H(Y_i, X) = -H(l_i)^T \hat{H}_{CI}^{inact}(Y_i). \tag{11}$$

When the absense score is used, the complete score function becomes

$$S_i(Y_i, X) = \sum_H S_i^H(Y_i, X) + AS_i^H(Y_i, X) + \lambda S_i^T(Y_i), \tag{12}$$

3.6 Optimization

For training, both the input music frames and the ground-truth harmony label segments are provided. The goal is to update the model parameters such that the probability computed in Eq. (2) is maximized. This is equivalent to minimizing the negative log likelihood (NLL) loss:

$$\begin{aligned} NLL(\theta) &= -\log P_\theta(Y|X) \\ &= log(Z_\theta(X)) - S_\theta(Y, X), \end{aligned} \tag{13}$$

where θ are the model parameters. We then compute the gradient of the loss with respect to the parameters to train our model using gradient descent.

During inference, where only the input music frames are provided, the goal becomes finding the correct segmentation and the corresponding labels that maximize the probability $P(Y|X)$. Since the normalization factor as a sum of exponential scores stays positive, maximizing the score function $S(Y, X)$ suffices to decode the segments and labels.

In both training and inference, we used the algorithms based on dynamic programming proposed in the original semi-CRF paper to expedite the optimization process [6].

4. EXPERIMENTS

4.1 Data

A collection of datasets from various sources [22, 25–27] organized by Micchi et al. [28] is used to train and evaluate the proposed architecture. Table 1 summarizes the statistics of the data included in our experiments. MusPy [29] is used to read the compressed MusicXML files and a parser adapted from [28] is employed to handle the proposed data representations. To increase the size of the dataset and help alleviate possible data imbalance, each piece is transposed to 12 different keys. The dataset is split into disjoint subsets for training and testing with a 2:1 split.

4.2 Implementation Details

Following the original paper for faster training [30], DenseNet is implemented in three separate blocks. 1-D convolution along the time frame dimension is used in each convolutional layer. Guided by the observation that harmony changes usually occur on average at a lower frequency than the frame rate, pooling layers are added between blocks to reduce the temporal resolution of the harmony output.

To ensure continuity and completeness of harmony regions in the training samples, we force the sample boundaries to be aligned with measure boundaries. A sample is chosen as 96 frames because it is divisible by all the common measure lengths existed in the dataset. Additionally, to avoid over-sampling from music pieces with longer length, the piece index is sampled uniformly first before a music sample is selected from the piece.

The entire pipeline is implemented using PyTorch. Adam optimizer is applied with learning rate of 10^{-4} and weight decay of 10^{-2}. Dropout with rate 0.2 is added between GRU layers and after each hidden fully-connected layer to avoid over-fitting. The λ in Eq. (12) is chosen empirically to be 0.001.

4.3 Evaluation Metrics

The task of music harmony analysis is two fold: recognizing the correct labels and finding the correct segmentation corresponding to the labels. To obtain a full picture of the model performance, we used two types of evaluation metrics to assess both aspects of the task.

First, the frame-level accuracy is computed for both root and quality. The accuracy on a reduced dictionary of quality including only major and minor is also reported due to its prevalence in the literature and adequacy in many practical uses. During training, the accuracy is computed at the sample level. During inference, the result is averaged across all frames in a song.

	Pieces	Crotchet	Chord Annotations
BPSFH	32	23554	8615
Roman Text	82	18208	7935
Tavern	27	20673	10723
Lopez	180	31367	16666

Table 1: Summary of statistics of the datasets.

The other evaluation metric focuses on the segmentation quality of the output. We use the standard segmentation scores from the mir_eval package [31, 32]. The scores are based on directional Hamming distance and consider the overlap between the estimated harmony intervals and the ground-truth intervals. The directional Hamming distance between the set of estimated intervals $\hat{\mathcal{I}} = \{\hat{I}_i\} = \{[\hat{u}_i, \hat{v}_i]\}$ and the set of ground-truth intervals $\mathcal{I} = \{I_i\}$ is computed as the following:

$$DHD(\hat{\mathcal{I}}, \mathcal{I}) = \frac{\sum_{\hat{I}_i \in \hat{\mathcal{I}}} (|\hat{I}_i| - \max_{I_j \in \mathcal{I}} |\hat{I}_i \cap I_j|))}{\sum_{\hat{I}_i \in \hat{\mathcal{I}}} |\hat{I}_i|}. \tag{14}$$

When a harmony boundary is missing from the estimation, an estimated harmony interval overlaps with multiple ground-truth intervals, but the maximum overlap is bounded by the length of the ground-truth intervals, leaving a large portion of the estimated interval not subtracted hence a large distance value. Therefore, a large $DHD(\hat{\mathcal{I}}, \mathcal{I})$ often indicates under-segmentation, while a large $DHD(\mathcal{I}, \hat{\mathcal{I}})$ often indicates over-segmentation. To summarize the two directional distances in a single metric, the overall segmentation quality score is computed as

$$SQ = 1 - \max(DHD(\mathcal{I}, \hat{\mathcal{I}}), DHD(\hat{\mathcal{I}}, \mathcal{I})). \tag{15}$$

4.4 Baseline Models

Three baseline models are included in our experiments to demonstrate the performance improvement of our proposed method. The chosen baselines are all relevant to our model by sharing parts of the architecture. Since the neural front-end of Harana is CRNN, we first test if a plain CRNN model [23] could achieve comparable results. A second baseline model, frog [28], also relies on CRNN to extract music features. In contrast to our model, it uses a neural autoregressive distribution estimator (NADE) to decode the harmony label. At the decoding stage, it defines an order of the harmony components and iteratively predict the next component conditioned on the current component. The same output harmony categories of root and quality output are considered in the NADE decoder. A third baseline worth comparing to is the rule-based semi-CRF proposed by Masada and Bunescu [21]. It uses handcrafted rules as features to compute the segment scores in semi-CRF. For simplicity, we implemented the two most important features, chord coverage and segment purity, in our experiment. Chord coverage measures what percentage of chordal notes are covered by the music segment while segment purity describes what proportion of notes in the music segment are indeed chordal notes.

Model	Root Acc	Quality Acc	Overall Acc	Under Seg	Over Seg	Overall Seg
Harana	**0.744**	0.743	**0.651**	**0.722**	0.747	0.649
Harana - no semi-CRF	0.732	0.715	0.634	0.678	0.740	0.639
Harana - no Attention Fusing	0.741	0.738	0.650	0.716	**0.749**	0.645
Harana - no Absence Score	0.743	**0.746**	0.643	0.719	0.748	**0.650**

Table 2: The result of ablation studies summarizing the effect of removing each proposed component of the model on both frame-level accuracy and segmentation quality.

Model	Root	Quality	Majmin	Overall
CRNN	0.735	0.714	0.865	0.634
frog	0.733	0.542	0.815	0.459
RuleSCRF	0.684	0.645	0.847	0.600
Harana	**0.744**	**0.743**	**0.886**	**0.651**

Table 3: The frame-level accuracy for different models.

Model	Under Seg	Over Seg	Overall
CRNN	0.681	0.738	0.639
frog	0.681	0.724	0.624
RuleSCRF	0.666	0.741	0.625
Harana	**0.722**	**0.747**	**0.649**

Table 4: The segmentation quality for different models.

5. RESULTS

5.1 Frame-Level Accuracy

Table 3 shows the result on frame-level accuracy. It can be seen that Harana outperforms the baseline models on all the measures. The large gap between Harana and the rule-based semi-CRF model demonstrates the value of a neural score function. Without a neural front-end, the rule-based model even has weaker performance than the plain CRNN. We also notice that frog has lower accuracy than the plain CRNN model. While the autoregressive decoding in frog could help enforce coherence between harmony components, it may require the full spectrum of the harmony components including key and degree. However, only root and quality were used in our experiments. Complete harmony information is difficult to collect so we believe Harana has a greater potential to leverage larger datasets in the future.

5.2 Segmentation Quality

As shown in Table 4, Harana provides improvement on segmentation quality compared to other models. Higher under-segmentation score of Harana means there are fewer missing boundaries in the estimation. Higher over-segmentation score shows that most detected boundaries are indeed true boundaries. An interesting observation is that the rule-based semi-CRF yields the most severe under-segmentation even though it is optimized on the segmentation boundaries. The reason for this might be that rule based-features are unable to clean noises such as the non-chordal notes and missing chordal notes in the input music but directly compute features from them. The noise in the features of short regions may be confused with the intrinsic noise of longer regions.

5.3 Ablation Studies

To show the effectiveness of each component of the architecture, we conduct additional ablation studies by removing each component. Table 2 summarizes the results. We can see that the full architecture achieves the best result overall. Among the missing components, semi-CRF leads to the largest performance drop. That confirms semi-CRF is an indispensable component to capture boundary information in harmony analysis. The attention module, although also helpful, produces relatively smaller performance gain. It is expected because after the neural front-end, the frame-level estimations to be aggregated may be already harmonically coherent; The attention module only helps to focus on the most representative frames. The effect of removing the absence score is less significant. Without it, the quality accuracy and overall segmentation quality even slightly improved. The phenomenon could result from the more difficult training objective. Inactive pitch class activations of the input music are an extreme scenario of noisy harmonic information. More data and a larger neural front-end might be needed to fully leverage the advantage of the absence score [33].

6. CONCLUSIONS

In this paper, we proposed an automated approach for harmonic analysis based on neural semi-CRF to jointly segment the harmonic regions and predict the labels. We developed a neural encoder and an attention mechanism to replace the conventional rule-based score function. We further proposed an absense score to improve the model robustness to imperfect harmony profiles. Experiments showed that our proposed architecture improves the performance on both frame-level accuracy and segmentation quality. Although our experiments focused on music input of symbolic format, the architecture could be adapted to audio input by simple modifications on the neural front-end. One limitation of the semi-CRF architecture is that it has quadratic time complexity with respect to sequence length so it is difficult to train the model on very long sequences. To capture the long-term dependency of harmony progression, more efficient sequence modeling methods could be explored in the future.

7. ACKNOWLEDGEMENTS

This work is partially supported by National Science Foundation grants No. 1846184 and 2222129. Frank Cwitkowitz would like to thank the synergistic activities funded by NSF grant DGE-1922591.

8. REFERENCES

[1] S. Kostka, D. Payne, and B. Almén, *Tonal harmony*. McGraw-Hill Higher Education, 2012.

[2] S. Bennett, "The process of musical creation: Interviews with eight composers," *Journal of Research in Music Education*, vol. 24, no. 1, pp. 3–13, 1976.

[3] W. F. Thompson, "Modeling perceived relationships between melody, harmony, and key," *Perception Psychophysics*, vol. 53, no. 1, pp. 13–24, 1993.

[4] B. Pardo and W. P. Birmingham, "Algorithms for chordal analysis," *Computer Music Journal*, vol. 26, no. 2, pp. 27–49, 2002.

[5] J. Pauwels, K. O'Hanlon, E. Gómez, and M. Sandler, "20 years of automatic chord recognition from audio," in *Int. Society of Music Information Retrieval Conf.*, 2019, pp. 54–63.

[6] S. Sarawagi and W. W. Cohen, "Semi-Markov conditional random fields for information extraction," in *Conf. on Neural Information Processing Systems*, 2004.

[7] T. Fujishima, "Real-time chord recognition of musical sound: A system using common lisp music," in *Int. Computer Music Conf.*, 1999.

[8] C. Harte and M. Sandler, "Automatic chord identifcation using a quantised chromagram," in *Audio Engineering Society Convention*, 2005.

[9] A. Sheh and D. P. Ellis, "Chord segmentation and recognition using em-trained hidden Markov models," in *Int. Society of Music Information Retrieval Conf.*, 2003, pp. 185–191.

[10] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals," in *Int. Society of Music Information Retrieval Conf.*, 2005, pp. 304–311.

[11] J. A. Burgoyne, L. Pugin, C. Kereliuk, and I. Fujinaga, "A cross-validated study of modelling strategies for automatic chord recognition in audio," in *Int. Society of Music Information Retrieval Conf.*, 2007, pp. 251–254.

[12] X. Zhou and A. Lerch, "Chord detection using deep learning," in *Int. Society of Music Information Retrieval Conf.*, 2015.

[13] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition," in *Int. Society of Music Information Retrieval Conf.*, 2017, pp. 188–194.

[14] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A bi-directional transformer for musical chord recognition," in *Int. Society of Music Information Retrieval Conf.*, 2019.

[15] T.-P. Chen and L. Su, "Harmony transformer: Incorporating chord segmentation into harmony recognition," in *Int. Society of Music Information Retrieval Conf.*, 2019.

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

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

[18] F. Korzeniowski and G. Widmer, "A fully convolutional deep auditory model for musical chord recognition," in *Int. Workshop on Machine Learning for Signal Processing*, 2016, pp. 1–6.

[19] Y. Wu and W. Li, "Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF sequence decoding model," in *Int. Workshop on Machine Learning for Signal Processing*, 2019, p. 355–366.

[20] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A bi-directional transformer for musical chord recognition," in *Int. Society of Music Information Retrieval Conf.*, 2019.

[21] K. Masada and R. C. Bunescu, "Chord recognition in symbolic music using semi-Markov conditional random fields," in *Int. Society of Music Information Retrieval Conf.*, 2017, pp. 272–278.

[22] D. Tymoczko, M. Gotham, M. S. Cuthbert, and C. Ariza, "The romantext format: A flexible and standard method for representing roman numeral analyses," in *Int. Society of Music Information Retrieval Conf.*, 2019.

[23] G. Micchi, M. Gotham, and M. Giraud, "Not all roads lead to rome: Pitch representation and model architecture for automatic harmonic analysis," *Trans. of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 42–54, 2020.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," in *Conf. on Neural Information Processing Systems*, 2017.

[25] T.-P. Chen and L. Su, "Functional harmony recognition of symbolic music data with multi-task recurrent neural networks," in *Int. Society of Music Information Retrieval Conf.*, 2018, pp. 90–97.

[26] J. Devaney, C. Arthur, N. Condit-Schultz, and K. Nisula, "Theme and variation encodings with roman numerals (tavern): A new data set for symbolic music analysis," in *Int. Society of Music Information Retrieval Conf.*, 2015.

[27] N. N. López, *Automatic harmonic analysis of classical string quartets from symbolic score*. Doctoral dissertation, Universitat Pompeu Fabra, 2017.

[28] G. Micchi, K. Kosta, G. Medeot, and P. Chanquion, "A deep learning method for enforcing coherence in automatic chord recognition," in *Int. Society of Music Information Retrieval Conf.*, 2017, pp. 443–451.

[29] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "Muspy: A toolkit for symbolic music generation," in *Int. Society of Music Information Retrieval Conf.*, 2020.

[30] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[31] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *Int. Society of Music Information Retrieval Conf.*, 2014, pp. 367–372.

[32] C. Harte, *Towards automatic extraction of harmony information from music signals*. Doctoral dissertation, Queen Mary University of London, 2010.

[33] J. Clarysse, J. Hörrmann, and F. Yang, "Why adversarial training can hurt robust accuracy," in *Int. Conf. on Learning Representations*, 2023.