CURRICULUM LEARNING FOR IMBALANCED CLASSIFICATION IN LARGE VOCABULARY AUTOMATIC CHORD RECOGNITION

Luke Rowe
University of Victoria
lukerowe@uvic.ca

George Tzanetakis
University of Victoria
gtzan@cs.uvic.ca

ABSTRACT

A problem inherent to the task of large vocabulary automatic chord recognition (ACR) is that the distribution over the chord qualities typically exhibits power-law characteristics. This intrinsic imbalance makes it difficult for ACR systems to learn the rare chord qualities in a large chord vocabulary. While recent ACR systems have exploited the hierarchical relationships that exist between chord qualities, few have attempted to exploit these relationships explicitly to improve the classification of rare chord qualities.

In this paper, we propose a convolutional Transformer model for the task of ACR trained on a dataset of 1217 tracks over a large chord vocabulary consisting of 170 chord types. In order to address the class imbalance of the chord quality distribution, we incorporate the hierarchical relationships between chord qualities into a curriculum learning training scheme that gradually learns the rare and complex chord qualities in the dataset. We show that the proposed convolutional Transformer model achieves state-of-the-art performance on traditional ACR evaluation metrics. Furthermore, we show that the proposed curriculum learning training scheme outperforms existing methods in improving the classification of rare chord qualities.

1. INTRODUCTION

The task of automatic chord recognition (ACR) has been an active area of research in the field of music information retrieval (MIR) for over 20 years [1]. This task automates the process of chord sequence annotation, which can be time consuming when done manually. ACR systems have been shown to be useful in other MIR applications, as chord annotations can be used as descriptive low-level features to assist other MIR tasks, such as key detection, harmonic analysis, and even style analysis [2]. Typically, an ACR system takes as input the audio signal corresponding to a musical recording. Then, the system outputs a time-aligned sequence of chord labels describing the underlying harmonic structure of the musical recording.

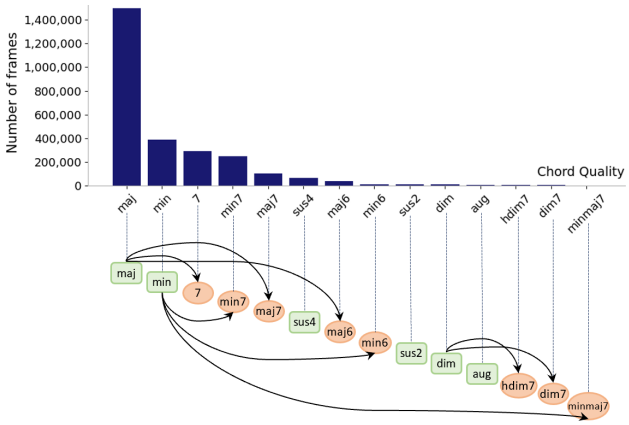


Figure 1: Top: Power-law distribution of chord qualities in BRIM (see Section 3 for details). Bottom: Hierarchy of chord qualities.

Most early ACR systems operated over a small chord vocabulary consisting of only major and minor chords and lacked the complexity needed for more complex chords. Recently, the focus has shifted to large vocabulary ACR, which includes a wider variety of chord qualities, such as augmented, diminished, sixth, seventh, and suspended chords. A critical issue with large vocabulary ACR is that the distribution over the chord qualities – and hence over the chord classes – exhibits a power-law distribution (see Figure 1). This imbalance is not specific to any particular ACR dataset but is intrinsic to large vocabulary ACR. Specifically, chord progressions seen across almost all genres of music overwhelmingly favor the major and minor chord qualities, which makes it difficult for ACR systems to learn the rare chord qualities.

Despite the complexities that arise in large vocabulary ACR, important structural relationships exist between the chord qualities in a large chord vocabulary [3]. As outlined in Figure 1, the chord qualities can be arranged into a hierarchical structure consisting of *base* chord qualities, or triads, (in rectangles) and *extended* chord qualities, or tetrads, (in ovals). Given an extended chord quality q_E and a base chord quality q_B, we say q_E *extends* q_B if the set of intervals that defines q_E is a superset of the set of intervals that defines q_B. Each extended chord quality extends a corresponding base chord quality, with the *extends* relationship indicated by an arrow in Figure 1. We hypothesize that an ACR model can better learn an extended chord quality when it has sufficiently learned its corresponding

base chord quality. The underlying intuition is that the base chord quality can be viewed as a harmonic base for the extended chord quality, and thus each extended chord quality can be interpreted as a more complex variant of its corresponding base chord quality. For example, the extended chord quality `7` can be viewed as a variant of base chord quality `maj`, where an additional `b7` interval is included.

Each extended chord quality is *rarer* in frequency than its corresponding base chord quality as can be seen in Figure 1. Based on this observation, we introduce a curriculum learning (CL) reweighting scheme that gradually converges from the initial distribution to a balanced chord quality distribution. This way, the curriculum allows the model to learn the base chord qualities prior to learning its corresponding extended chord qualities. The proposed scheme is integrated with a convolutional Transformer model which is trained over a chord vocabulary of 170 chord types. We show that the proposed model achieves state-of-the-art performance on traditional ACR evaluation metrics. Moreover, we show that the proposed CL scheme outperforms existing methods in improving the classification of rare chord qualities.

2. RELATED WORK

2.1 Automatic Chord Recognition

Most ACR systems have two stages: feature extraction and chord sequence decoding. Arguably the most notable recent advancement in ACR is the replacement of traditional machine learning with deep learning for both stages of the ACR pipeline. For example, recent ACR systems have utilized deep neural networks [4–6], convolutional neural networks [7–11], and deep belief networks [12, 13] to produce robust feature representations that outperform earlier conventional methods. Moreover, recurrent neural networks [4, 9–14] and conditional random fields (CRFs) [8, 10, 11] have largely replaced hidden Markov models to capture the temporal dependencies in the chord sequence decoding process. Inspired by the recent success of Transformer-based models in the field of natural language processing [15–18], recent approaches have applied end-to-end Transformer-based models to the task of ACR [19] and to the related tasks of symbolic chord recognition and functional harmony recognition [20, 21].

Recent focus has shifted to the large vocabulary variant of the ACR task [3, 9, 11, 13, 14]. Since the chord quality distribution over a large chord vocabulary is extremely skewed, these systems must explicitly overcome the *imbalanced class-learning problem*, whereby model learning is biased towards the frequently-labelled classes, resulting in poor classification performance of the sparsely-labelled classes [22]. To address this problem within large vocabulary ACR, recent approaches have incorporated auxiliary training targets by decomposing chords into structured components [9, 11]. However, these structured training methods still provide limited exposure to the rare chord qualities, and thus model learning is still heavily biased towards the frequently-labelled chord qualities. Deng and Kwok addressed this problem by implementing an "even-chance" training scheme, which ensures that each chord type has an even chance of being chosen at the beginning of each training sample [14]. Jiang *et al.* combined their structured chord representation with a reweighting scheme to reduce the model learning bias induced by the imbalanced distribution of each structured component [11].

2.2 Curriculum Learning

CL was first proposed by Bengio *et al.* in [23], where they demonstrated that for certain tasks with an established difficulty metric, introducing training data from easy to hard difficulty in a deep neural network can lead to faster convergence and guide training towards better local minima. To the best of our knowledge, the only existing application of CL to the task of ACR is in [24]. In [24], McVicar *et al.* designed a curriculum to train an ACR system using ground-truth annotations with noisy alignments. CL has recently been utilized to address the imbalanced class-learning problem. In [25], Wang *et al.* proposed a CL training scheme for the task of human attribute recognition, which gradually converges from the training distribution to a balanced distribution to improve the classification performance of sparsely-labelled classes.

3. DATA PREPARATION

The ground-truth chord labels are mapped to a chord vocabulary V of 170 chords [9]. V includes chords that span all 12 pitch classes and the following 14 chord qualities Q: *maj, min, dim, aug, min6, maj6, min7, minmaj7, maj7, 7, dim7, hdim7, sus2,* and *sus4*. Additionally, V contains two extra labels: N (no chord) and X (unknown chord). Our model also integrates the structured chord representation proposed in [9], whereby each chord label is decomposed into its root, bass, and pitch structured components.

For training and evaluation, we use the dataset collected by Humphrey and Bello [9, 11, 26], which comprises of 1217 tracks from the **B**illboard, **R**WC Pop, **I**sophonics, and **M**ARL collections. We refer to this collected dataset as **BRIM**. We augment the training data using `MUDA` [27] by pitch-shifting each audio track across -6 to $+5$ semitones. To properly compare the performance on traditional ACR metrics with previous methods, we use the same 5-fold cross-validation split that is used in [9, 11, 26].

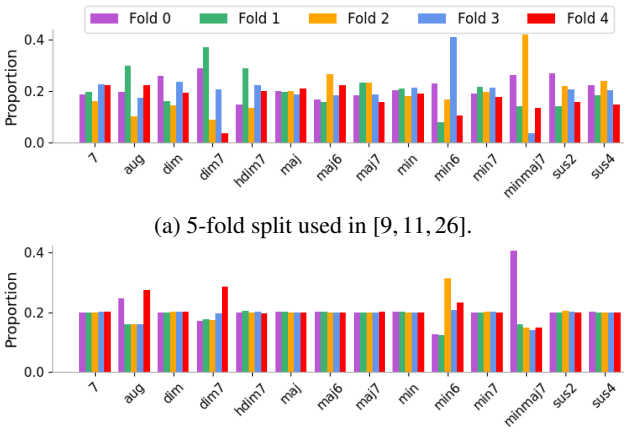
We employ a separate 5-fold cross-validation split for the imbalanced class-learning ACR experiments. Since the chord-type distribution in BRIM is extremely imbalanced, it is imperative for proper evaluation that this distribution is maintained across each fold of our 5-fold split. However, since the 5-fold split occurs at the *track* level but the distribution is measured at the *frame* level, we found stratifying over the chord types to be intractable. Therefore, we propose an approximate 5-fold stratification algorithm that instead ensures that the distribution over the *chord qualities* in BRIM is approximately maintained across each fold. The proposed algorithm (Algorithm 1) takes as input a set of *chord quality profiles* $P = \{P_t\}$, where $P_t[q]$ is the pro-

Algorithm 1: N-Fold Chord Quality Stratification

Input: Number of folds: N; set of chord qualities: Q; set of tracks T; chord quality profiles: $P = \{P_t\}$; rarest chord quality: q_r.

for $i = 0$ **to** $N - 1$ **do**
  initialize empty list *folds*$[i]$
  initialize fold profile F_i, where for each $q \in Q$, $F_i[q] = 0$
T_{sorted} = SortDescending(T, by=$P[q_r]$)
for $i = 0$ **to** $N - 1$ **do**
  append $T_{\text{sorted}}[i]$ to *folds*$[i]$
  $F_i[q] = F_i[q] + P_{T_{\text{sorted}}[i]}[q]$ for each $q \in Q$
  remove $T_{\text{sorted}}[i]$ from T
while $T \neq \emptyset$ **do**
  $q' = \underset{q \in Q}{\text{argmax}}\ \text{Var}(F_0[q], \ldots, F_{N-1}[q])$
  $t_{\text{min}} = \underset{t \in T}{\text{argmin}}\ P_t[q']$
  $i_{\text{max}} = \underset{i \in \{0,\ldots,N-1\}}{\text{argmax}}\ F_i[q']$
  append t_{min} to *folds*$[i_{\text{max}}]$
  $F_{i_{\text{max}}}[q] = F_{i_{\text{max}}}[q] + P_{t_{\text{min}}}[q]$ for each $q \in Q$
  remove t_{min} from T
  for each remaining fold $i \neq i_{\text{max}}$ **do**
    $t' = \underset{t \in T}{\text{argmin}}\ P_t[q'] + F_i[q'] - F_{i_{\text{max}}}[q']$
    append t' to *folds*$[i]$
    $F_i[q] = F_i[q] + P_{t'}[q]$ for each $q \in Q$
    remove t' from T
return *folds*

portion of q chords in track t over BRIM, for each $q \in Q$. The algorithm iteratively builds up a fold profile F_i for each fold i, where $F_i[q]$ is the proportion of q chords in fold i over BRIM, for each $q \in Q$. At each iteration, one track is added to each fold. At iteration 1, we take the 5 tracks with the highest proportion of the rarest chord quality q_r and add one track to each fold. Each subsequent iteration can be viewed as a "correction step," whereby one track is added to each fold to minimize the variance of the highest-variance chord quality over the folds. The result of the 5-fold chord quality stratification is shown in Figure 2b.



(a) 5-fold split used in [9, 11, 26].

(b) 5-fold Stratified Split.

Figure 2: 5-fold splits of BRIM. The height of each bar at chord quality q corresponds to the proportion of q chords contained in the corresponding fold.

4. METHODS

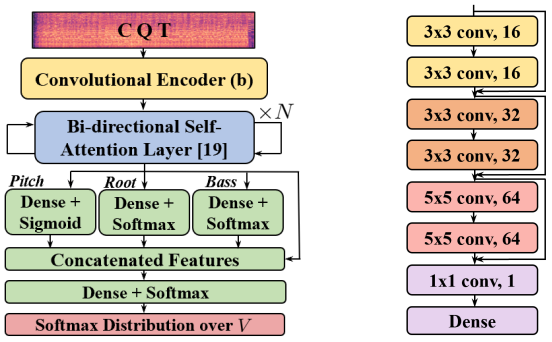
4.1 Convolutional Transformer (CT)

The proposed system uses the log-power Constant Q-Transform (CQT) spectrogram as its input feature representation. We apply a convolutional-residual encoder shown in Figure 3b to capture short-term context and induce sufficient temporal smoothing of the spectrogram. The encoder first applies batch-normalization (BN) [28] to the input CQT. We then apply a series of convolutional layers and add 3 skip connections [29] to ease the training process. Each convolutional layer is zero-padded to preserve the spatial dimension of the CQT. After each convolutional layer, a BN layer followed by a Rectified Linear Unit (ReLU) activation is applied. The output of the convolutional encoder is passed through a stack of N bi-directional self-attention layers proposed in [19]. Details of this layer can be found in [19]. We replace the absolute positional encoding used in [19] with relative positional encoding [16], which has been shown to offer better generalization capabilities by taking into account the relative positions between frames in the self-attention mechanism.

The model facilitates structured training of the root note, bass note, and pitch classes as is done in [9]. Unlike in [9], we multiply the pitch structured loss by $\gamma > 1$ to assign more priority to the pitch structured component. The model learns to jointly minimize the cross-entropy chord label loss L_{label} and the cross-entropy structured loss L_{struct}, where L_{struct} is the sum of the cross entropy losses for the pitch, root, and bass structured components. For each frame t, the model outputs a softmax distribution $\hat{y}^{(t)} \in [0, 1]^{|V|}$ over V. At evaluation time, the system predicts the label with the highest activation in $\hat{y}^{(t)}$ for each frame t. An overview of the model is shown in Figure 3.

4.2 Curriculum Learning

The idea of CL is to train the *easy* samples before the *hard* samples. Loosely, we define an easy sample as a frame where the ground-truth chord quality is a *base* chord quality, and a hard sample as a frame where the ground-truth chord quality is an *extended* chord quality. We want to ensure that for each extended chord quality, the system first sufficiently learns its corresponding base chord quality. A critical observation in Figure 1 is that each extended chord quality is rarer than its corresponding base chord quality. Based on this observation, we propose a CL reweighting scheme that gradually converges from the training distribution to a balanced chord quality distribution, similar to the scheme proposed by Wang *et al.* in [25]. The proposed scheme enables the model to put emphasis on the frequently-labelled chord qualities at the beginning of training and put increasingly more emphasis on the rare chord qualities as training converges. Since the pitch-



(a) (b)

Figure 3: Proposed CT architecture. (a) outlines the end-to-end architecture and (b) outlines the layers of the proposed convolutional encoder.

shifting data augmentation eliminates the root bias in the training data, then balancing the chord-quality distribution also balances the chord-type distribution during training.

The proposed scheme runs for E epochs. At epoch $e = 1$, the target distribution is the training distribution (*i.e.* an imbalanced chord quality distribution). As e approaches E, the system gradually modifies the target chord quality distribution to be more balanced; this is achieved by reweighting samples by placing higher weights on the frames where the ground-truth chord quality is sparsely-labelled and lower weights on the frames where the ground-truth chord quality is frequently-labelled. At epoch $e = E$, the chord quality distribution is balanced.

Let $Q' = Q \cup \{N, X\}$. Let C_q be the number of frames in the training set with chord quality $q \in Q'$ and let $q_{min} \in Q'$ be the chord quality with the least number of frames in the training set. We define the chord quality training distribution D_{train} by $D_{\text{train},q} = \frac{C_q}{C_{q_{min}}}$ for all $q \in Q'$.

Let D_e be the target chord quality distribution at epoch e. Then $D_1 = D_{\text{train}}$. During model training, the target chord quality distribution gradually transfers to a balanced distribution with the following function:

$$D_{e,q} = (D_{\text{train},q})^{g(e)} \; \forall q \in Q' \tag{1}$$

where e is the epoch number and $g(e)$ is the curriculum scheduler function. The scheduler function is a monotonically decreasing function from 1 to 0 that sets the pace of the curriculum. We experiment with three scheduler functions (visualized in Figure 4): $g(e) = 0$ (baseline, fixed balanced chord quality distribution), $g(e) = 1 - \frac{e-1}{E-1}$ (linear schedule), and $g(e) = \phi^e - \phi^E$ (convex schedule), where ϕ is a hyperparameter. Observe that for all three scheduler functions, $g(E) = 0$ and thus $D_{E,q} = 1$ for all q; *i.e.* the target chord quality distribution is balanced.

At epoch e, to facilitate training with target chord quality distribution D_e, we reweight the samples such that for chord class $i \in V$ having chord quality $q \in Q'$, the class weight w_i assigned to i in L_{label} is defined by:

$$w_i = D_{e,q} / D_{\text{train},q} \tag{2}$$

As the training set chord quality distribution is extremely imbalanced, the variation in the magnitude of the weights w_i across the different chord classes $i \in V$ is extremely large at epoch $e = E$. We hypothesize that this may cause the temporal smoothness of the output predictions to be impaired. Therefore, we train an additional CRF decoder on top of the output logits to smooth the output predictions, in the same way as [8].

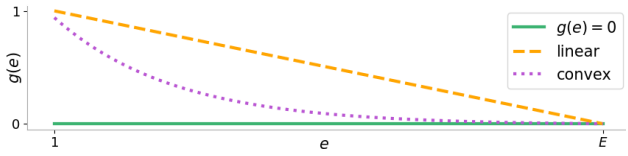


Figure 4: Proposed curriculum scheduler functions.

The proposed CL scheme differs from [25] in three critical ways. First, the training set statistics are used for computing the reweighting terms w_i, whereas [25] uses batch statistics. Using batch statistics is suboptimal for ACR since batches typically consist of a small number of sequences, and the frames of each sequence are highly interdependent. Therefore, we believe that the training statistics are more representative of the true chord quality imbalance. Second, in [25], the frequently-labelled samples are *down-sampled* by setting some samples to have weight 0 and the remaining to have weight 1. Down-sampling is ill-advised for ACR as this would disrupt the temporal coherence of the training sequence. Therefore, the proposed scheme instead employs a fully reweighted approach so that the continuity of the training sequences are preserved. Third, the proposed scheme employs a CRF decoder to smooth the output predictions at model convergence.

5. EXPERIMENTS

5.1 Model Evaluation

To compare the proposed CT model with previous methods, evaluation is conducted using `mir_eval` [30]. We obtain Weighted Chord Symbol Recall (WCSR) scores for: Root, Thirds, Triads, Sevenths, Tetrads, Maj-Min, and MIREX. We average the results of each metric across the folds, as is done in [19]. For the methods addressing the imbalanced class-learning problem, we utilize two evaluation metrics proposed in [11]: the mean frame-wise accuracy ($\text{acc}_{\text{frame}}$) and mean class-wise accuracy ($\text{acc}_{\text{class}}$) over V. $\text{acc}_{\text{frame}}$ is defined by:

$$\text{acc}_{\text{frame}} = \frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} F_i} \tag{3}$$

where n is the number tracks for evaluation, F_i is the number of frames in track i, and C_i is the number of correctly-predicted frames in track i over vocabulary V. $\text{acc}_{\text{class}}$ is defined by:

$$\text{acc}_{\text{class}} = \frac{1}{|V|} \sum_{v \in V} \frac{\sum_{i=1}^{n} C_i^v}{\sum_{i=1}^{n} F_i^v} \tag{4}$$

where F_i^v is the number of frames in track i with ground-truth chord label v and C_i^v is the number of frames in track

i that are correctly predicted as v. Further, we define an additional metric termed the mean quality-wise accuracy ($\text{acc}_{\text{quality}}$) over V to be used in the CL experiments. We define $\text{acc}_{\text{quality}}$ by:

$$\text{acc}_{\text{quality}} = \frac{1}{|Q|} \sum_{q \in Q} \frac{\sum_{i=1}^{n} C_i^q}{\sum_{i=1}^{n} F_i^q} \tag{5}$$

where F_i^q is the number of frames in track i with ground-truth chord quality q and C_i^q is the number of frames in track i that are correctly predicted as having ground-truth chord quality q. As previously outlined in [11, 14], when addressing the imbalanced class-learning problem within the task of large vocabulary ACR, we want to maximize the $\text{acc}_{\text{class}}$ while still maintaining the $\text{acc}_{\text{frame}}$.

5.2 Implementation Details

Using `librosa` [31], audio is transformed into a log-power constant-Q spectrogram spanning 6 octaves with 36 bins per octave. The sample rate is 44100 *Hz*, and the hop size is 4096. We tune the hyperparameters of the bi-directional self-attention layers (optimized to BRIM): number of self-attention layers $N = 6$, number of self-attention heads $n_h = 8$, hidden dimension $d = 512$, and dropout probability $p = 0$. The CT model is trained using the Adam optimizer [32] with initial learning rate 1e-4. The learning rate is reduced by a factor of 10 when the validation L_{label} loss does not improve after 10 consecutive epochs. Model training terminates when the validation L_{label} loss does not improve after 20 consecutive epochs. We save the model weights with the lowest validation L_{label} loss for evaluation. In each epoch, a contiguous 248-frame segment is randomly sampled from each training track, and a mini-batch consists of 32 such segments. Structured training is conducted in the same way as [9], with the exception that we set $\gamma = 7$. We set the L_{label} class weights to $w_i = 1$ for all $i \in V$.

For the CL experiments, we set $E = 90$ and $\phi = 0.95$. Since only the *chord qualities* of the 5-fold split are stratified, the imbalance in the root distribution across the folds may cause $\text{acc}_{\text{class}}$ to be an unreliable validation metric. Thus, we instead use $\text{acc}_{\text{quality}}$. The training details remain the same with a few exceptions. Namely, the learning rate is reduced by a factor of 10 when the validation $\text{acc}_{\text{quality}}$ has not improved for 10 consecutive epochs. For the linear and convex scheduler functions, we save the model weights at convergence (epoch $e = E$) for evaluation. For the baseline scheduler function $g(e) = 0$, we save the model weights with the highest validation $\text{acc}_{\text{quality}}$ for evaluation. We train the CRF using Adam with a learning rate of 1e-2. We terminate training the CRF once the validation $\text{acc}_{\text{quality}}$ stops improving. Our implementation is available at `https://github.com/RLuke22/curriculum-learning-acr`.

5.3 Methods under Comparison

We compare our proposed CT architecture with CR2S+A [9], BTC [19] and the best-performing model of [11], which we call CRNN. As BRIM is significantly larger than the dataset used to train the BTC model, the BTC model is re-trained on BRIM as described in Section 5.2. We evaluate these models using the WCSR metrics and the same 5-fold split of BRIM that is used in [9, 11, 26].

All CL experiments are trained and evaluated with the CT model. We denote the models with convex and linear scheduler functions as CT+$\text{CL}_{\text{Convex}}$ and CT+$\text{CL}_{\text{Linear}}$, respectively. The baseline model with scheduler function $g(e) = 0$ is denoted CT+$\text{CL}_{\text{Baseline}}$. We also evaluate the even-chance training scheme proposed in [14], which we call CT+EC. Specifically, we adjust the CT model training procedure so that each chord type $v \in V$ has an even chance of being selected at the beginning of each training segment. $\text{acc}_{\text{quality}}$ is used as the validation metric for the CT+EC model. Further, we evaluate the reweighting scheme of [11] on the CRNN model. We experiment with the best-reported reweighting configuration $(\gamma, w_{\text{max}}) = (0.5, 10.0)$, which we call $\text{CRNN}_{0.5,10}$. For evaluation we use the $\text{acc}_{\text{frame}}$ and $\text{acc}_{\text{class}}$ metrics using the stratified 5-fold split of BRIM.

Metric	CT	$\text{CT}_{-\text{RE}}$	$\text{CT}_{-\text{RE,S}}$	$\text{CT}_{-\text{RE,S,C}}$ (BTC) [19]	CR2S+A [9]	CRNN^{*} [11]
Root	**0.838**	0.836	0.831	0.829	0.821	0.837
Thirds	**0.809**	0.807	0.802	0.798	0.784	0.803
Triads	**0.767**	0.764	0.759	0.754	0.742	0.759
Sevenths	**0.714**	0.711	0.709	0.700	0.677	0.694
Tetrads	**0.650**	0.646	0.644	0.638	0.615	0.630
Maj-Min	**0.826**	0.823	0.819	0.813	0.802	0.822
MIREX	**0.832**	0.828	0.825	0.820	0.803	0.812

Table 1: WCSR scores averaged across 5 folds. −RE denotes removal of relative positional encoding. −S denotes removal of structured training. −C denotes removal of the convolutional encoder. *operates over a larger chord-vocabulary V' consisting of 301 chord types [11].

Method	$\text{acc}_{\text{frame}}$	$\text{acc}_{\text{class}}$
CT	**0.677**	0.347
$\text{CT+CL}_{\text{Baseline}}$	0.647	0.427
$\text{CT+CL}_{\text{Linear}}$	0.658	0.439
$\text{CT+CL}_{\text{Convex}}$	0.657	**0.449**
CT+EC [14]	0.650	0.379
$\text{CRNN}_{0.5,10}$ [11]	0.630	0.321

Table 2: $\text{acc}_{\text{frame}}$ and $\text{acc}_{\text{class}}$ scores over V for all data-balancing methods using stratified split over BRIM.

5.4 Results

The WCSR scores for all models considered are shown in Table 1. Table 1 also includes an ablation study that outlines the performance degradation with the removal of each novel component in the CT architecture; *i.e.* the convolutional encoder (C), structured training (S), and relative positional encoding (RE). Note that the CT architecture without all three novel components (and with the inclusion of global z-normalization) is equivalent to BTC [19]. Table 1 shows that the CT model outperforms existing ACR systems across all WCSR metrics. Further, the ablation study indicates that each novel component offers a gradual, yet consistent improvement to the CT model.

Table 2 shows the results of the methods that address

	Chord Quality Accuracy													
Method	maj	min	7	min7	maj7	sus4	maj6	min6	sus2	dim	aug	hdim7	dim7	minmaj7
CT	**0.801**	**0.637**	0.518	0.576	0.570	0.277	0.102	0.134	0.034	0.365	0.236	0.357	0.031	0.031
CT+CL$_{\text{Convex}}$	0.735	0.626	**0.537**	**0.595**	**0.634**	0.406	**0.275**	**0.288**	**0.261**	**0.395**	**0.517**	**0.476**	0.181	**0.229**
CT+CL$_{\text{Baseline}}$	0.727	0.618	0.512	0.581	0.604	**0.411**	0.250	0.264	0.235	0.386	0.460	0.450	**0.209**	0.151

Table 3: Chord quality accuracies of various CT models over BRIM at evaluation. Chord quality accuracy is defined as the proportion of frames where the predicted chord quality matches the ground-truth chord quality.

imbalanced class-learning including a baseline CT model (*i.e.* no reweighting). Unsurprisingly, the CT model performs the best in the acc$_{\text{frame}}$ metric. This is consistent with previous works that have shown that optimizing the class-wise accuracy typically harms the frame-wise accuracy [11,14]. The best-performing CL model CT+CL$_{\text{Convex}}$ provides substantial improvement (10.2%) in the class-wise accuracy, with only a modest degradation (2.0%) in the frame-wise accuracy. This indicates that the CL scheme considerably suppresses the learning bias in the model induced by the imbalanced chord quality distribution without significantly impairing the performance of the frequently-labelled classes. Moreover, both CL configurations CT+CL$_{\text{Convex}}$ and CT+CL$_{\text{Linear}}$ offer improvements in the acc$_{\text{frame}}$ and acc$_{\text{class}}$ metrics over the baseline CT+CL$_{\text{Baseline}}$. This indicates that by having the model sufficiently learn the base chord qualities prior to the corresponding extended chord qualities, the model better generalizes on both the frequently-labelled and sparsely-labelled chord qualities. This is further confirmed in Table 3, which shows that CT+CL$_{\text{Convex}}$ outperforms CT+CL$_{\text{Baseline}}$ in chord quality accuracy on every chord quality except for `sus4` and `dim7`.

In Figure 5, we evaluate the chord quality accuracies of the CT+CL$_{\text{Convex}}$ model at different epochs in the curriculum. Note that the `dim` base chord quality accuracy (in dark green) improves substantially from epochs 1 to 10, followed by an improvement in the `hdim7` (in red) and `dim7` (in purple) extended chord qualities from epochs 10 to 20 and 10 to 40, respectively. Similar trends can be observed for the `maj` and `min` base chord qualities. This indicates that sufficient learning of the base chord qualities leads to performance improvements in the corresponding extended chord qualities. We hypothesize that the convex scheduler function outperforms the linear scheduler function in class-wise accuracy because the extreme imbalance in the chord quality distribution warrants a faster curriculum pace at the beginning of training. As shown in Table 2, the proposed CT+CL$_{\text{Convex}}$ model convincingly outperforms previous methods in the acc$_{\text{class}}$ metric. Note that the CRNN$_{0.5,10}$ results in Table 2 differ from the results reported in [11] as we run CRNN$_{0.5,10}$ over a different chord vocabulary V than the one used in [11].

To validate the inclusion of the CRF decoder in the CL scheme, we count the number of chord changes in BRIM predicted by the CT model, the CT+CL$_{\text{Convex}}$ model, and the CT+CL$_{\text{Convex}}$ model without the CRF (denoted CT+CL$_{\text{Convex}}$-CRF). Table 4 shows that the model weights at CL convergence disrupt the smoothness of the output chord-label predictions, as evidenced by the substantially larger number of predicted chord changes by the CT+CL$_{\text{Convex}}$-CRF model.

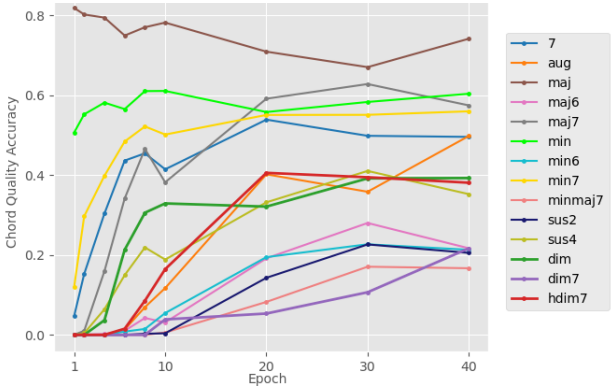


Figure 5: Chord quality accuracies of CT+CL$_{\text{Convex}}$ evaluated at different points along the curriculum.

Method	Chord Changes
CT	170,287
CT+CL$_{\text{Convex}}$	123,350
CT+CL$_{\text{Convex}}$-CRF	235,885
Ground-Truth	122,231

Table 4: Predicted chord changes over BRIM.

6. CONCLUSION

We propose a convolutional Transformer architecture for ACR and a novel CL reweighting scheme to handle the imbalanced chord quality distribution. The proposed scheme exploits the hierarchical relationships between chord qualities by gradually converging from the initial distribution to a balanced chord quality distribution. The proposed curriculum outperforms existing methods and non-CL baselines in improving the classification performance of rare chord qualities without significantly degrading the classification performance of the frequently-labelled chord qualities. Although the proposed method considerably diminishes the model-learning bias induced by the imbalanced chord quality distribution, the model still generally favors the frequently-labelled chord qualities. We believe this is primarily an issue of data scarcity. Therefore, a promising future direction to handle the imbalanced class-learning problem for ACR is to generate more annotated data either synthetically or by leveraging the vast amount of publically-available unannotated audio tracks.

7. ACKNOWLEDGEMENTS

We would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for their generous funding and support of this work. We would also like to thank Brian McFee for providing us with the BRIM dataset, Quinton Yong for his helpful insights on the stratification algorithm, and the anonymous reviewers for their insightful and constructive feedback.

8. REFERENCES

[1] T. Fujishima, "Real-time chord recognition of musical sound: A system using common lisp music," in *Proceedings of the International Computer Music Conference (ICMC)*, 1999, pp. 464–467.

[2] C. Weiß, F. Brand, and M. Müller, "Mid-level chord transition features for musical style analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 341–345.

[3] T. Carsault, J. Nika, and P. Esling, "Using musical relationships between chord labels in automatic chord extraction tasks," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 18–25.

[4] S. Sigtia, N. Boulanger-Lewandowski, and S. Dixon, "Audio chord recognition with a hybrid recurrent neural network," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 127–133.

[5] X. Zhou and A. Lerch, "Chord detection using deep learning," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 52–58.

[6] F. Korzeniowski and G. Widmer, "Feature learning for chord recognition: The deep chroma extractor," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 37–43.

[7] E. Humphrey and J. P. Bello, "Rethinking automatic chord recognition with convolutional neural networks," in *11th International Conference on Machine Learning and Applications (ICMLA)*, 2012, pp. 357–362.

[8] F. Korzeniowski and G. Widmer, "A fully convolutional deep auditory model for musical chord recognition," in *26th IEEE International Workshop on Machine Learning for Signal Processing*, 2016, pp. 1–6.

[9] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 188–194.

[10] Y. Wu and W. Li, "Automatic audio chord recognition with midi-trained deep feature and BLSTM-CRF sequence decoding model," *IEEE ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 2, pp. 355–366, 2019.

[11] J. Jiang, K. Chen, W. Li, and G. Xia, "Large-vocabulary chord transcription via chord structure decomposition," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 644–651.

[12] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, 2013, pp. 335–340.

[13] J. Deng and Y. Kwok, "A hybrid gaussian-HMM-deep learning approach for automatic chord estimation with very large vocabulary," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 812–818.

[14] ——, "Large vocabulary automatic chord estimation with an even chance training scheme," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 531–536.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017, pp. 5998–6008.

[16] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2019, pp. 2978–2988.

[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[18] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*.

[19] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A bi-directional transformer for musical chord recognition," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 620–627.

[20] T. Chen and L. Su, "Harmony transformer: Incorporating chord segmentation into harmony recognition," in

Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR), 2019, pp. 259–267.

[21] ——, "Attend to chords: Improving harmonic analysis of symbolic music using transformer-based models," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 1–13, 2021.

[22] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[23] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, 2009, pp. 41–48.

[24] M. McVicar, Y. Ni, T. D. Bie, and R. Santos-Rodriguez, "Leveraging noisy online databases for use in chord recognition," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 639–644.

[25] Y. Wang, W. Gan, J. Yang, W. Wu, and J. Yan, "Dynamic curriculum learning for imbalanced data classification," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5016–5025.

[26] E. Humphrey and J. P. Bello, "Four timely insights on automatic chord estimation," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 673–679.

[27] B. McFee, E. Humphrey, and J. P. Bello, "A software framework for musical data augmentation," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 248–254.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning, (ICML)*, 2015, pp. 448–456.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[30] C. Raffel, B. McFee, E. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. Ellis, "Mir_eval: A transparent implementation of common MIR metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference, (ISMIR)*, 2014, pp. 367–372.

[31] B. McFee, V. Lostanlen, M. McVicar, A. Metsai, S. Balke, C. Thomé, C. Raffel, D. Lee, F. Zalkow, K. Lee, O. Nieto, J. Mason, D. Ellis, R. Yamamoto, E. Battenberg, R. Bittner, K. Choi, J. Moore, Z. Wei, S. Seyfarth, P. Friesch, F. Stöter, D. Hereñú, T. Kim, M. Vollrath, and A. Weiss, "librosa/librosa: 0.7.1," Oct. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3478579

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, (ICLR)*, 2015.