Three Metrics for Musical Chord Label Evaluation

McLeod, Andrew
Fraunhofer IDMT
Ilmenau, Germany
andrew.mcleod@idmt.fraunhofer.de

Suermondt, Xavier
EPFL
Lausanne, Switzerland
xavier.suermondt@epfl.ch

Rammos, Yannis
EPFL
Lausanne, Switzerland
yannis.rammos@epfl.ch

Herff, Steffen
Western Sydney University
Sydney, Australia
s.herff@westernsydney.edu.au

Rohrmeier, Martin A.
EPFL
Lausanne, Switzerland
martin.rohrmeier@epfl.ch

ABSTRACT

Harmony constitutes an essential aspect of a broad range of styles in Western music, and chords usually play a key role therein. Consequently, the generation or detection of chords is central to a wide range of computational models, for instance in chord estimation, chord sequence prediction, and harmonic structure detection. Such models are typically evaluated by comparing their outputs to ground-truth chord labels using a binary metric ("correct" or "incorrect"). As chord vocabularies continue to grow, binary metrics capture less information about the correctness of a given label, thus equating all labeling errors regardless of their severity. In this work, we present the `chord-eval` toolkit, which proposes three different metrics drawn, adapted, and generalized from previous work, addressing acoustic, perceptual, music-theoretical, and mechanical aspects of evaluation. We discuss use cases for which the metrics vary in appropriateness, depending on properties of the underlying music and the task at hand, and present an example of such differences.

CCS CONCEPTS

• **Information systems** → **Similarity measures**.

KEYWORDS

music, music information retrieval, harmony, chords, similarity metric

ACM Reference Format:
McLeod, Andrew, Suermondt, Xavier, Rammos, Yannis, Herff, Steffen, and Rohrmeier, Martin A.. 2022. Three Metrics for Musical Chord Label Evaluation. In *Forum for Information Retrieval Evaluation (FIRE '22), December 9–13, 2022, Kolkata, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3574318.3574335

1 INTRODUCTION

Chords play an essential role in the harmonic structure of music across a wide variety of genres and eras. As such, the understanding of chord progressions is fundamental to many tasks in Music Information Retrieval (MIR), such as music generation [3, 27], automatic accompaniment [30, 31], chord estimation [14, 16], and harmonic structure analysis [17, 18]. Note that systems designed to perform a particular task often include many components working to understand various aspects of chords—for example, the audio chord transcription system from [14] includes a chord sequence model as well as an acoustic chord estimation model, and automatic accompaniment systems include components designed to detect previous chords in addition to predicting the next.

Regardless of the task, a quantitative evaluation is almost always performed involving the comparison of the proposed system's output chord labels with some (typically human annotator-generated) ground-truth labels. Even in the case of music generation, chord prediction is often used (in addition to a listening experiment) as a proxy for generation performance [3]. Fundamentally then, across this wide range of tasks, the essence of evaluation is defining a distance function between two chord labels.Most often, this function is a simple binary accuracy metric, where the distance between two matching labels is 0 and all other distances are 1. However, there are issues with this approach: in particular the equating of all errors, varying vocabulary size, and low inter-annotator agreement.

The equating of all errors: With a binary metric, every error is penalized equally, although, depending on the task, labelling mistakes may be unequally egregious. For example, with a ground truth label of C major, C minor and A minor are clearly much closer to being correct than D♯ minor. In evaluation, such differences can be key, as two models which have the same binary accuracy may differ considerably in terms of the significance of their errors. To alleviate this issue, more granular evaluation procedures are sometimes used, which also calculate binary accuracies between *features* of each chord label (e.g. the root pitch, the bass pitch, or the chord type), either independently or in combination with each other (e.g. inversion accuracy given that the root is correct) [2, 17, 18]. While this helps, the underlying issue remains: Within a particular feature, every mistakes is penalized equally, and any interaction between feature-errors (e.g., given an incorrect root, an incorrect inversion might be better) is ignored.

Varying vocabulary size: Traditionally, a rather small vocabulary of chord labels was used, often 24 or 25 (12 pitch classes for the root, and either major or minor quality, plus sometimes a "no-chord" symbol) [14]. However, recently, as models have become more sophisticated and the availability of labeled data has increased dramatically, ever larger chord vocabularies have been utilized, incorporating spelled pitch [17], additional qualities (e.g. augmented

FIRE '22, December 9–13, 2022, Kolkata, India

ACM ISBN 979-8-4007-0023-1/22/12.
https://doi.org/10.1145/3574318.3574335

and diminished triads, sus chords, and various 7th, 9th, 11th, chords) [26], and more features (e.g. bass/inversion, as well as chord tone alterations and suspensions) [16]. Thus, there is a need for any chord evaluation metric to enable a comparison between ground truth and estimated labels drawn from different vocabularies, or between models whose outputs lie in different vocabularies. For example, with a binary metric, a model which outputs only triads would be evaluated unfavorably if the ground truth also contains 7th chords, regardless of whether or not the model correctly predicts the underlying triad. One solution that has been used is to apply a "reduction" mapping all labels into a smaller vocabulary containing only a subset of used chord features (or equivalently ignoring particular features for some metrics) [5, 23]. (A somewhat similar approach is used in [12].) However, this does not fully solve the issue: In this case, a different model capable of outputting information about additional 7ths could not be rewarded for a correct additional output (and its triad-reduced labels may suffer if much of its computational power goes towards finding the correct 7th).

Low inter-annotator agreement: In some cases, ground truth chord labels can be drawn from some canonical source (e.g., as [3] used the real book). However, it is more common that human expert annotators are relied upon to produce the labels. Annotators may, however, have disagreeing judgments within a given piece. The determination of harmonic content often requires subtle assessments of the given contrapuntal, motivic, textural, and even formal context. In turn, such interpretation may require, for example, a distinction between chord and consonant non-chord tones; choices between alternative "diagonal" relations between bass and melody tones; or the recognition of a theme type (e.g. antecedent-consequent phrase) suggesting a harmonic template. Often, such decisions depend on a particular reading of the section (interpretation), and it is not uncommon for two annotators to disagree on a particular label: A recent survey on audio chord recognition reported (from [4, 13]) inter-annotator agreement on chord root to range from 76% to 94% [22]. To penalize models equally severely for every mistake could be unfair or uninformative, since expert annotators could make (or even have made) the exact same "mistake" (even calling such a case a mistake is debatable).

In addition to these three issues, given the large variety of relevant tasks, to propose a single metric that would adequately capture the intricacies of error severity for all tasks simultaneously would be tremendously difficult, if not infeasible. Indeed, the suitability of a metric for one task in no way entails its suitability for another task, as it is unclear a priori how to weight different aspects of the resulting harmony for each. For example, a metric for automatic accompaniment may focus on evaluating the similarity of the sounds of the two chords, while a metric for harmonic structure analysis may instead focus on more structural or functional aspects of chord labels.

Therefore, in this paper, rather than proposing a single "best-in-all-cases" metric, we argue for a task-specific approach to chord label evaluation. To that end, we present the `chord-eval` toolkit[1], in which (in addition to the binary metrics discussed above) three different metrics are proposed and implemented. Our toolkit supports an extremely flexible encoding of each chord (see Section 3.1), including spelled pitch and chord tone alterations. Each has been designed to focus on particular features of chords relevant to a subset of use cases, and we include in our description of each a discussion regarding which tasks it may be appropriate for.

[1] https://github.com/DCMLab/chord-eval

2 EXISTING METRICS

While some more sophisticated chordal distance (and equivalently similarity) metrics have been proposed, the most commonly used by far are still fundamentally binary. Specifically, the widely used `mir_eval` package implements many such measures across combinations of a variety of chord features, including root pitch, third quality, triad type, 7th, and inversion [25]. However, these metrics—based on the MIREX 2013 evaluation procedure, which is still in use for the current MIREX—still generally suffer from an equating of all errors discussed above.

Some metrics treat each chord label as a set of non-privileged (root and bass information is ignored) pitch classes. Enharmonic equivalence is assumed such that A♯ and B♭ are the same pitch class. The precision and recall of these sets (or other measures, e.g., [6]; unordered set comparisons in [7]) can be measured (called Chroma Precision and Chroma Recall in [23]). These are more flexible metric than the binary feature-based metrics, but ignoring root and bass information is often undesired, and there is no understanding of "how incorrect" a given pitch class is.

Tymoczko discusses several different classes of musical distance metrics (e.g., in [29]), as well as their correlation in different tonal contexts. Among them, the class of voice-leading distance metrics is most closely related to our mechanical distance, and a further comparison is drawn in Section 3.4. These metrics generally attempt to quantify the number of steps it would take to transform one chord to another following a series of voice-leading operations. Alternative implementations proposed assume both ordered and unordered pitch class sets, and represent "distance" in a variety of geometrical spaces.

For all of the above metrics, once distances between each label and the ground truth are measured, Chord Symbol Recall (CSR; [7]) is typically used to measure the quality of the labels across each piece by taking a weighted average: weighing each distance (or similarity) by the duration of the associated segment. Our proposed metrics (and indeed all of the discussed chord distance metrics) are fully compatible with CSR in this way.

3 PROPOSED METRICS

3.1 Chord Encoding

In our toolkit, a chord label is fully-described by its root note, type (major, minor, minor 7, etc.), inversion (from which we can derive its bass note), and alterations, which include added tones (e.g., +9, +11), removed tones (e.g., an open fifth on C could be encoded as a C major triad with a removed E), and replaced tones (e.g., a C suspended-fourth ("sus") chord could be encoded as a C major where the E has been replaced by an F). When a pitch replaces the bass note, it becomes the bass note (e.g., a Csus4 chord in 1st inversion has a bass note of F). When the nominal bass note is removed, we treat the next chord tone (not including added tones) as the bass note (e.g., in a C major triad in 1st inversion, with an added F and a removed E, the G is treated as the bass note)Pitches

may be encoded either as neutral pitch classes (NPCs, MIDI note number modulo 12), or tonal pitch classes (TPCs, where C♯ and D♭ are distinct) [28]. This versatile protocol enables the representation of highly interpretive hearings (for example, the encoding of a first-inversion C triad as an inversion of a cadential 64 chord on a G root).

3.2 Spectral Pitch Similarity

Spectral Pitch Similarity (SPS) is a measure proposed to evaluate the similarity between the perceived pitch content of tones or chords based on relevant psychoacoustic assumptions (see [20] for more detail). In essence, it is a measure of the distance between the partials of two notes or chords. In [19], SPS (together with voice-leading distances) was proposed as a cornerstone of bottom-up psychoacoustic approaches to explain regularities in tonal-harmonic music. Indeed, SPS has been shown to be a good predictor of listeners' perception in a variety of perceptual tasks. For example, SPS carries predictive value for tonal fit responses in Krumhansl and Kessler's [15] influential probe tone data set [20]. Furthermore, the predictive power of SPS for listeners' responses also generalises to unfamiliar microtonal stimuli, further supporting the psycho-acoustic versatility of the metric [21]. On the basis of this evidence, we implement an SPS-based metric for chord evaluation tasks in which timbral perception, in the broad sense of the term as an emergent property of composite pitch events [8], is important. This could, for example, be the case for automatic accompaniment tasks, or audio transcription software.

The calculation of the SPS between tones or chords relies on a spectrogram of the audio. However, since our toolkit is designed to measure the distance between two chord *labels*, not two audio clips, we must first synthesize each label. Broadly, this process works by creating a MIDI file with the appropriate pitches, and then synthesizing the audio, and calculating the SPS between spectrograms of the synthesized chords.

By default, we generate three chords per label: in closed form with its bass note in the 3rd, 4th, and 5th octaves. Alternatively, the user has the option to specify which pitches in particular to use for the synthesis by giving a list of potential MIDI note numbers in a `pitches` parameter. If given, the notes in the synthesized chord will be drawn only from that list, where any note number from the list which is equivalent modulo 12 to a chord tone from our default process will be included in the MIDI file, and all others will be discarded. For example, a C major triad with `pitches` containing C2, E4, C5, G6, and A6 will generate a MIDI chord with the pitches C2, E4, C5, and G6. This allows the user to ensure that the voicing of a chord label corresponds to a particular musical score.

Once the pitches are known, we create the MIDI files by generating a 1 second note on each resulting pitch, starting at time 0, using the `pretty_midi` python package [24] and FluidSynth.By default, all notes are synthesized using the piano program, but this can be changed for either chord independently using a parameter. This is important to note, because from a perceptual perspective, the similarity or fit between notes or chords can be subject to timbre and, in extension, instrument choice [20]. After synthesis, we compute a spectrogram for each chord, which may be either a variable-Q transform (VQT, default), a constant-Q transform (CQT), a short-time Fourier transform, or a mel spectrogram based on the user's choice. We then take the central frame from a spectrogram of each chord, which is used to avoid any percussive noise from the chord's attack while also not straying too far into its decay, and compute their cosine similarity. We then compute the cosine distance between each pair of spectrograms (since we generate 3 spectrograms per chord label by default, there are 6 pairs, but there will be only a single pair if the `pitches` parameter is used), and return 1 minus the maximum cosine similarity as the SPS distance.

3.3 Tone-by-Tone Distance

Tone-by-tone distance treats each chord as sets of pitch classes, which may be either tonal or neutral (see Section 3.1). For each chord, we then measure the proportion of its pitch classes which are contained in the other chord. The final distance between the two chords is one minus the average of each of these two proportions. When the two pitch class sets are of equal size, the two proportions will be the same (e.g. for A minor and C major triads, the proportion for each is $\frac{2}{3}$). However, the two proportions can differ when one set is larger than the other (e.g., for A minor and C7, the proportions are $\frac{2}{3}$ and $\frac{1}{2}$ respectively, leading to a tone-by-tone distance of $1 - \frac{7}{12} = \frac{5}{12}$).

This is thus far quite similar to the pitch class set-based metrics discussed in Section 2, with the additional allowance for using tonal pitch classes. We further generalize this understanding by including parameters that put additional importance on whether the two chords share the same root (`root_bonus`, b_R) or bass note (`bass_bonus`, b_B), two features that are often emphasized in MIR task outputs. By default these two values are both 1, putting importance on both the bass and the root of each chord. For use cases in which bass or root notes are of particular importance, positive values should be preferred. Positive values of `bass_bonus` are appropriate, for example, if the evaluation context is governed by thoroughbass or *partimento* principles, in which the bass voice has primary structural significance, and upper voices are determined in terms of intervals from the bass. On the other hand, if harmonic function is significant to the evaluation, positive values of `root_bonus` will generally provide a more relevant metric.When $b_B = b_R = 0$, the metric is similar to some of those from Section 2.

In total, this calculation is given in Equation 1. C_1 and C_2 are the pitch class sets of each chord, R is b_R if the root notes match and 0 otherwise, and B is b_B if the bass notes match and 0 otherwise. Essentially, a value of 1 (for either b_R or b_B) will measure the distance as if there was an additional root or bass note in each pitch class set (and 2 will weight it as an additional 2 notes, etc.). For example, for A minor 1st inversion and C major, the tone-by-tone distance would be $\frac{1}{3}$ by default, $\frac{1}{2}$ (slightly worse) with $b_R = 1$ and $b_B = 0$, and $\frac{1}{4}$ (slightly better) with $b_R = 0$ and $b_B = 1$.

$$avg\left(\frac{|C_1 \cap C_2| + R + B}{|C_1| + b_R + b_B}, \frac{|C_2 \cap C_1| + R + B}{|C_2| + b_R + b_B}\right) \tag{1}$$

Intuitively, tone-by-tone distance considers chords that share more pitches to be more similar. We thus expect it to be most meaningful in "pitch-class counterpoint" contexts in which binary per-note metrics ("correct" vs. "incorrect") need to be synthesized

into a granular, non-binary metric for the entire chord (e.g., for polyphonic textures). The metric is also useful whenever fine-grained evaluations are required, while avoiding aesthetically or music-theoretically contentious choices of acoustic metrics (such as SPS above) and voice-leading metrics (such as our mechanical distance below). In this sense, it may be understood as a deliberately naive proxy, substituting notational and textural differences for a more music-theoretically grounded evaluation. Additionally, practitioners who wish to prioritize the music theoretical concepts of root or bass note (as is common), and those who wish to express chord labels using TPCs (which is becoming more common) have the flexibility to do so, both features missing from previous proposals.

3.4 Mechanical Distance

Mechanical distance again treats chords as pitch class sets, and roughly corresponds to a measure of the physical distance between two chord labels as they are played on an instrument. It can be viewed as a further granularization of the tone-by-tone distance, where instead of just measuring what proportion of pitches are incorrect, we also measure how far away each erroneous note is from the target chord. One potential use case is during music performance evaluation (see, e.g., [11]), for example during instrumental lessons or for performances involving robotic musicians, where mechanical mistakes are common. On the piano, a minor misplacement of a finger can generate an erroneous tone (e.g. semitone errors). In tasks involving the automatic evaluation of such performances, the mechanical distance between an intended set of tones and the produced set of tones may be more relevant than, for instance, the spectral relationship between the generated sounds (as in SPS), and more informative than tone-by-tone distance.

It should be noted that, using default settings, this metric is quite similar to some of the voice leading distances discussed by Tymoczko (e.g., in [29])—although here, we consider the bass pitch to be "ordered" in the sense of it being non-permutable, while the other pitches are all "unordered": a novel combination to our knowledge. While it is true that mechanical distance can be seen as a generalization of this type of voice leading distance, and is indeed useful as such, we choose not to use the term "voice leading", since doing so would suggest taking into account the larger context of a chord label (local key, diatonic steps vs. semitones, etc.). It would also involve complex and often controversial music-theoretical decisions between, for example, scalar conceptions of root distance (according to which a root motion by semitone is smaller than that by step) and *Tonnetz*-based models (which typically allow only parsimonious voice-leading operations, and consider root motion by thirds or fifths to be more "proximate" than motion by step or half-step).We leave such discussion for future work.

For mechanical distance, we first calculate the *bass distance* as the distance between the bass notes of each chord, times some `bass_weight` B (1 by default). By assigning a special status to the bass, this metric integrates mechanical (ergonomic) and musical intuitiveness. Jazz pianists are known to associate chord voicings with the kinaesthetic experience of "hand positions" between outer notes, while similar intersections between thoroughbass practice and fingerings have been historically established in 18th-century keyboard treatises [1]. By default, our "mechanical distance" distance is measured in semitones (as is natural on a piano keyboard), but users may also supply any custom distance measure, provided it assigns a value to each semitone interval from 0 to 11 (e.g., a user may choose to measure distance in fifths, where C is adjacent only to F and G). This parametric freedom makes the metric adaptable to different organological constraints beyond those of keyboard instruments.

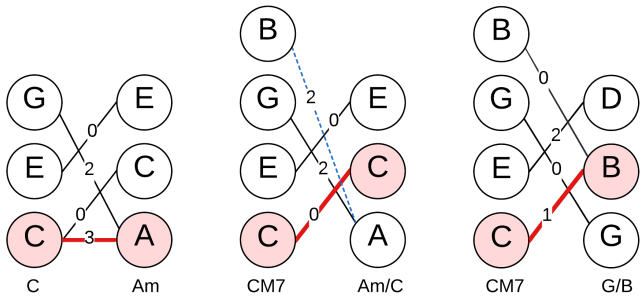

Figure 1: Example spatial distance pairings of C vs. Am (left, distance 5), CM7 vs. Am/C (center, distance 4), and CM7 vs. G/B (right, distance 3). Edge labels are semitone distance, bass notes have a shaded background, bold red edges correspond to bass distance, and the blue dashed edge connects unmatched notes.

We then create a fully-connected bipartite graph between the pitches of each chord, where each edge's weight is the distancebetween the two pitches. We calculate a minimum weight full matching of the graph [10], and take the sum of the resulting edge weights as a *matched distance* (if the bass notes are matched together in this graph, they are left out of this sum, since they are included in the bass distance). Finally, if there are any remaining unmatched pitch classes, for each unmatched pitch class in the larger chord, we take the minimum distance from it to any pitch class in the smaller chord. The mechanical distance between the two chords is the sum of these "unmatched" distances, the bass distance, and the matched distance.Through this process, we have paired every pitch class from each chord to at least one pitch class in the other chord.

Examples of these pairings can be seen in Figure 1. Here, notice that there are no unmatched edges in the CM7 vs. G/B pairing: Since the minimum weight full matching of the graph only leaves CM7's bass note C disconnected (which is already paired with G/B's B), rather than the B (as in the CM7 vs. Am/C case), there are no unmatched pitch classes.

4 EXAMPLES

To investigate each metric's results for different chord pairs, we present distance plots for each in Figure 2. Here, one chord is always a C major triad in root position, while the other varies across different roots, types, and inversions. The top plot shows SPS distance, the middle shows Tone by Tone distance, and the bottom shows mechanical distance, all with default settings.

In this context, SPS correlates somewhat with tone-by-tone, but does show more gradations for its penalties. For example, notice that every chord with root G—with the exception of diminished

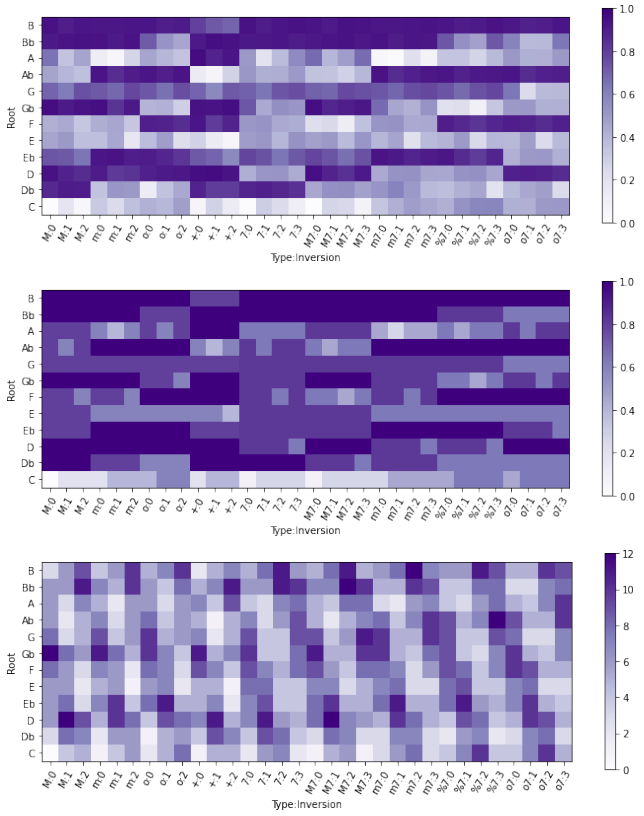

Figure 2: Selected distances (top: SPS; middle: tone-by-tone; bottom: mechanical) from a C major root position triad to various other chords with default settings (o and % signify diminished and half-diminished).

7th chords—has an identical tone-by-tone distance to the C major, since they each share exactly 1 note, and never the bass. However, SPS tends to prefer 1st inversion chords (and in particular triads) where the voicing most similar (i.e. the G in both chords occurs as the third-lowest note). Likewise for chords with root E, SPS prefers them in 2nd inversion. Mechanical distance is quite distinct from the others, having a strong preference for those chords whose bass note is close to C in semitone space, gradually getting worse the larger the interval grows.

4.1 Pairwise Comparisons

Often the metrics correlate with each other. Thus we perform pairwise comparisons between the metrics to better understand cases in which each differs significantly from the others. To do so, we computed the distance—using each metric with default settings—between all pairs of chords with any NPC root; any chord type of major, minor, diminished, augmented, dominant 7th, major 7th, minor 7th, minor major 7th, diminished 7th, half-diminished 7th, augmented 7th, and augmented major 7th; any inversion; and no alterations. Then, for a given distance d, we compute its "standardized" value d' by first computing that metric's mean μ and standard deviation σ across all chords, and setting $d' = (d - \mu)/\sigma$. That is, d' now expresses d in terms of its distance away from that metric's mean, measured in standard deviations. Then, we searched for chords in which each pair of metrics disagreed the most in this standardized form.

4.1.1 SPS and Tone by Tone. Chord pairs which have a significantly lower standardized SPS distance than standardized tone-by-tone distance generally differ in both root and bass pitch. Their bass notes tend to differ by only a semitone (pitch height is a salient property for SPS), at least one other tone matches, and matching tones occur in the same place in the chord's voicing (e.g., as its upper note). Examples include Cm7 (C, E♭, G, B♭) and G♭ major, 2nd inversion (D♭, G♭, B♭); Cm, 1st inversion (E♭, G, C) and D7 (D, F♯, A, C); and Cdim, 1st inversion (E♭, G♭, C) and F major (F, A, C). Each of these pairs has a standardized SPS distance of around −1 and a standardized tone-by-tone distance around 0.3.

Intuitively, chord pairs with a much lower standardized tone-by-tone distance than SPS distance tend to match in both root and bass note, and differ by a semitone for other non-matching notes. In cases where their bass notes differ, the interval tends to be relatively large, around a major or minor third. Examples include C major (C, E, G) and C minor (C, E♭, G); C major, 1st inversion (E, G, C) and C7, 2nd inversion (G, B♭, C, E); and C minor, 1st inversion (E♭, G, C) and Cm7, 2nd inversion (G, B♭, C, E♭). Interestingly, these examples all again have a standardized SPS distance of around −1, but now a standardized tone-by-tone distance of −2.5 to −2.

4.1.2 SPS and Mechanical. Chord pairs for which the standardized mechanical distance is significantly lower than the standardized SPS distance are those where each note differs by exactly one semitone, for example C major and D♭ major, or C diminished and B diminished. This differs from the above SPS < tone-by-tone case due to the absence of any note overlap, which SPS penalizes severely. These examples have a standardized SPS distance around 1.5 and a standardized mechanical distance around −1.3. On the other hand, standardized SPS is lower than standardized mechanical distance for chord pairs whose bass notes are a tritone apart, and whose other notes are either far apart (maximizing mechanical distance) or match (minimizing SPS distance). These include, for example, C diminished (C, E♭, G♭) and G♭ diminished (enharmonic to G♭, A, C); C minor, 1st inversion (E♭, G, C) and G♭ diminished, 1st inversion (enharmonic to A, C, G♭); and C minor (C, E♭, G) and Cm7, 2nd inversion (G, B♭, C, E♭). These pairs have standardized SPS distances from −1.9 to −1.5 and standardized mechanical distances from 1.5 to 2.

4.1.3 Tone by Tone and Mechanical. Chord pairs for which the standardized mechanical distance is significantly lower than the standardized tone-by-tone distance are again those where each note differs by exactly one semitone (trivially, their tone-by-tone distance is the maximum possible 1). Excluding these, examples are similar, though the pairs now match in 1 (non-bass) tone: for example, C minor (C, E♭, G) and B major (enharmonic to B, E♭, G♭); as well as C minor and G major, 1st inversion (B, D, G). These chords have a standardized mechanical distance of around −1.7 and a standardized tone-by-tone distance of around 0.3. Those for which the standardized tone-by-tone distance is significantly lower than the standardized mechanical distance are similar to the SPS <

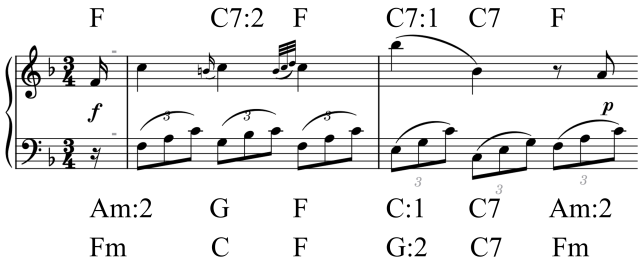

Figure 3: The first two bars from Mozart K279-2 [9], with ground truth labels above the score, and two potential estimates below. Labels are notated as "chord:inversion".

mechanical case, again including chords whose bass notes differ by a tritone and whose other notes match. Here, though chords also tend to share their root (since tone-by-tone has $b_B = 1$), thus also including many chords which differ only by inversion. For example, C diminished (C, E♭, G♭) and C diminished 7th, 2nd inversion (G♭, B♭♭, C, E♭); and C7, 1st inversion (E, G, B♭, C) and C7, 3rd inversion (B♭, C, E, G). These pairs have a standardized tone-by-tone distance around −2 and a standardized mechanical distance around 1.

4.2 A Case Study

In Figure 3, we present a musical excerpt with annotated ground truth labels and labels that could have been output by a two different chord estimation models. The models each achieve a binary CSR of 31.6. However, while they perform similarly in terms of SPS distance (20.9 for the top and 19.9 for the bottom), their other two distances differ significantly. The bottom outperforms the top in tone-by-tone 24.5 to 33.4, but the top outperforms the bottom in mechanical distance 1.47 to 2.11. This difference in preference highlights the variability in evaluation, as well as the importance of picking the appropriate metric for the task.

5 CONCLUSION

In this paper, we have presented the `chord_eval` toolkit for the evaluation of chord label accuracy. We have argued that, given the wide variety of relevant tasks and the many different potentially important aspects of harmony and chords, an appropriate metric should be chosen based on the desired use case. Our toolkit contains (in addition to the traditional binary) 3 metrics: SPS distance, based on Spectral Pitch Similarity [20] and useful for acousto-perceptual evaluation; tone-by-tone distance, which (similar to previous work) measures the proportion of correct tones in a target chord, and can be seen as a rough proxy for simple music theoretical intuitions, particularly with our novel inclusion of bass and root bonuses and tonal pitch classes; and mechanical distance, similar to existing voice leading distances, which is a further granularization of tone-by-tone, and includes a novel special handling of the bass note.

We have specifically left out a deeper (and much more complicated) consideration of the key and the holistic tonal context here, instead focusing first on the distance between a pair of chord labels in isolation. However, a complete evaluation of the harmonic labeling of a musical composition requires this context, which we intend to address in future work. Indeed, for models which attempt to output this full harmonic structure, a metric which takes the full tonal context into account would be greatly beneficial—if not essential—to continued improvement.

ACKNOWLEDGMENTS

This project was partially funded through the Swiss National Science Foundation (SNF) within the project "Distant Listening – The Development of Harmony over Three Centuries (1700–2000)" (grant no. 182811). The authors thank Claude Latour for supporting this research through the Latour Chair in Digital Musicology at EPFL.

REFERENCES

[1] Carl Philipp Emanuel Bach. 1951 [1797]. *Essay on the True Art of Playing Keyboard Instruments.* W. W. Norton, New York.
[2] Tsung Ping Chen and Li Su. 2018. Functional harmony recognition of symbolic music data with multi-task recurrent neural networks. In *ISMIR.* 90–97.
[3] Keunwoo Choi, George Fazekas, and Mark Sandler. 2016. Text-based LSTM networks for automatic music composition. *arXiv preprint arXiv:1604.05358* (2016).
[4] Trevor De Clercq and David Temperley. 2011. A corpus analysis of rock harmony. *Popular Music* 30, 1 (2011), 47–70.
[5] Junqi Deng and Yu-Kwong Kwok. 2018. Large vocabulary automatic chord estimation using bidirectional long short-term memory recurrent neural network with even chance training. *Journal of New Music Research* 47, 1 (2018), 53–67.
[6] Johanna Devaney. 2021. Beyond chord vocabularies: Exploiting pitch-relationships in a chord estimation metric. In *ISMIR Late Breaking Demo Session.*
[7] Christopher Harte. 2010. *Towards automatic extraction of harmony information from music signals.* Ph. D. Dissertation. Queen Mary University of London.
[8] Robert Hasegawa. 2019. Timbre as Harmony—Harmony as Timbre. In *The Oxford Handbook of Timbre.* Oxford University Press, 525–551.
[9] Johannes Hentschel, Markus Neuwirth, and Martin Rohrmeier. 2021. The Annotated Mozart Sonatas: Score, Harmony, and Cadence. *Transactions of the International Society for Music Information Retrieval* 4, 1 (2021), 1–14. https://doi.org/10.5334/tismir.63
[10] Richard M Karp. 1980. An algorithm to solve the m×n assignment problem in expected time $O(mn \log n)$. *Networks* 10, 2 (1980), 143–152.
[11] Hyon Kim., Pedro Ramoneda., Marius Miron., and Xavier Serra. 2022. An Overview of Automatic Piano Performance Assessment within the Music Education Context. In *Proceedings of the 14th International Conference on Computer Supported Education - Volume 1: CSME,.* INSTICC, SciTePress, 465–474. https://doi.org/10.5220/0011137600003182
[12] Katherine M Kinnaird and Brian McFee. 2021. Automatic Hierarchy Expansion for Improved Structure and Chord Evaluation. *Transactions of the International Society for Music Information Retrieval* 4, 1 (2021).
[13] Hendrik Vincent Koops, W Bas De Haas, John Ashley Burgoyne, Jeroen Bransen, Anna Kent-Muller, and Anja Volk. 2019. Annotator subjectivity in harmony annotations of popular music. *Journal of New Music Research* 48, 3 (2019), 232–252.
[14] Filip Korzeniowski and Gerhard Widmer. 2018. Improved chord recognition by combining duration and harmonic language models. In *ISMIR.*
[15] Carol L Krumhansl and Edward J Kessler. 1982. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological review* 89, 4 (1982), 334.
[16] Brian McFee and Juan Pablo Bello. 2017. Structured training for large-vocabulary chord recognition. In *ISMIR.*
[17] Andrew McLeod and Martin Rohrmeier. 2021. A Modular System for the Harmonic Analysis of Musical Scores using a large vocabulary. In *ISMIR.* 435–442.
[18] Gianluca Micchi, Mark Gotham, and Mathieu Giraud. 2020. Not all roads lead to Rome: Pitch representation and model architecture for automatic harmonic analysis. *Transactions of the International Society for Music Information Retrieval (TISMIR)* 3, 1 (2020), 42–54.
[19] Andrew J Milne. 2010. Tonal music theory: A psychoacoustic explanation?. In *Proceedings of the 11th International Conference on Music Perception and Cognition (ICMPC11).*
[20] Andrew J Milne, Robin Laney, and David B Sharp. 2015. A spectral pitch class model of the probe tone data and scalic tonality. *Music Perception: An Interdisciplinary Journal* 32, 4 (2015), 364–393.
[21] Andrew J Milne, Robin Laney, and David B Sharp. 2016. Testing a spectral model of tonal affinity with microtonal melodies and inharmonic spectra. *Musicae Scientiae* 20, 4 (2016), 465–494.
[22] Johan Pauwels, Ken O'Hanlon, Emilia Gomez, and Mark B. Sandler. 2019. 20 Years of Automatic Chord Recognition from Audio. In *ISMIR.* 54–63. https://doi.org/10.5281/zenodo.3527739

[23] Johan Pauwels and Geoffroy Peeters. 2013. Evaluating automatically estimated chord sequences. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 749–753.
[24] Colin Raffel and Daniel P. W. Ellis. 2014. Intuitive Analysis, Creation and Manipulation of MIDI Data with pretty_midi. In *ISMIR Late Breaking and Demo Papers*.
[25] Colin Raffel, Brian Mcfee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel P. W. Ellis, Colin Raffel, Brian Mcfee, and Eric J. Humphrey. 2014. mir_eval: A Transparent Implementation of Common MIR Metrics. In *ISMIR*.
[26] Luke O Rowe and George Tzanetakis. 2021. Curriculum Learning for Imbalanced Classification in Large Vocabulary Automatic Chord Recognition. In *ISMIR*. 586–593. https://doi.org/10.5281/zenodo.5624463
[27] Hao Hao Tan. 2019. ChordAL: A Chord-Based Approach for Music Generation using Bi-LSTMs.. In *ICCC*. 364–365.
[28] David Temperley. 2000. The line of fifths. *Music Analysis* 19, 3 (2000), 289–319.
[29] Dmitri Tymoczko. 2009. Three conceptions of musical distance. In *International Conference on Mathematics and Computation in Music*. Springer, 258–272.
[30] Wei Yang, Ping Sun, Yi Zhang, and Ying Zhang. 2019. CLSTMS: A Combination of Two LSTM Models to Generate Chords Accompaniment for Symbolic Melody. In *International Conference on High Performance Big Data and Intelligent Systems*. 176–180. https://doi.org/10.1109/HPBDIS.2019.8735487
[31] Yin-Cheng Yeh, Wen-Yi Hsiao, Satoru Fukayama, Tetsuro Kitahara, Benjamin Genchel, Hao-Min Liu, Hao-Wen Dong, Yian Chen, Terence Leong, and Yi-Hsuan Yang. 2021. Automatic melody harmonization with triad chords: A comparative study. *Journal of New Music Research* 50, 1 (2021), 37–51.