Evaluating Interval-based Tokenization for Pitch Representation in Symbolic Music Analysis

Dinh-Viet-Toan Le, Louis Bigo, Mikaela Keller

Evaluating Interval-based Tokenization for Pitch Representation in Symbolic Music Analysis

Dinh-Viet-Toan Le[1], **Louis Bigo**[2], **Mikaela Keller**[1]

[1]Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille
[2]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence
dinhviettoan.le@univ-lille.fr

Abstract

Symbolic music analysis tasks are often performed by models originally developed for Natural Language Processing, such as Transformers. Such models require the input data to be represented as sequences, which is achieved through a process of *tokenization*. Tokenization strategies for symbolic music often rely on absolute MIDI values to represent pitch information. However, music research largely promotes the benefit of higher-level representations such as melodic contour and harmonic relations for which pitch intervals turn out to be more expressive than absolute pitches. In this work, we introduce a general framework for building interval-based tokenizations. By evaluating these tokenizations on three music analysis tasks, we show that such interval-based tokenizations improve model performances and facilitate their explainability.

Introduction

Originating from the field of Natural Language Processing (NLP), the term *tokenization* initially refers to the representation of a textual content in a sequential format. The field of Music Information Retrieval has then largely adopted this term to describe the process of representing symbolic music as sequences of tokens (Le et al. 2024). This sequential representation of music enables its processing through sequential models widely adopted in the field of NLP, such as Transformers. In contrast to text tokens, musical time and pitch are represented by metric values, raising the question of *absolute* versus *relative* representations. While time encoding strategies have been studied and compared (Fradet et al. 2023), traditional tokenization methods mostly use *absolute* pitch encoding, which may overlook relational aspects between notes.

Instead, music is often memorized by its melodic contour, considered as a sequence of intervals unchanged by transposition to different keys, rather than by their absolute pitches (Dowling and Fujitani 1971). Similarly, in the context of tonal music, harmony is based on the relation between the notes constituting a chord and a tonal center more than their absolute pitches. Musical *intervals* capture the relative distances between pitches, emphasizing relationships over fixed pitches, which aligns more closely with human musical perception. Applied to symbolic music representation, interval-based tokenization may provide a more intuitive approach to pitch encoding.

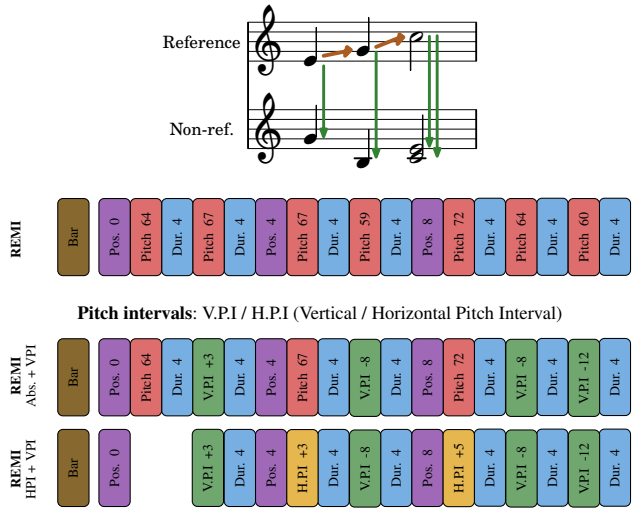

Figure 1: Representations of the sheet music based on absolute and different variants of intervalization of the REMI tokenization. (Abs.: *Absolute pitch encoding*)

In this study, we present tokenization strategies based on intervals that can be used jointly with absolute pitch encodings. Such tokenizations are shown to improve model performances in analysis tasks. This paper's contribution is twofold:

- We first propose a general framework to build interval-based tokenization strategies.
- We then show that interval-based tokenization can improve model performances on three downstream tasks. Moreover, we show that among the studied interval-based encoding strategies, optimal tokenization settings depend on the downstream task and can result in musically meaningful interpretations.

Symbolic music tokenization

Most tokenization strategies rely on absolute pitch encodings. Pitches are usually encoded based on MIDI num-

bers (Huang et al. 2019; Hsiao et al. 2021), as used in the REMI tokenization (Huang and Yang 2020) shown in Figure 1. Various alternatives relying on an absolute reference have been proposed, in particular leveraging the equivalence where a pitch can be decomposed into a `<pitch-class>` and `<octave>` token (Li, Li, and Fazekas 2023). Other pitch decompositions can result from specific instrument practice, such as the DadaGP tokenization (Sarmento et al. 2021) dedicated to guitar tablatures, in which pitches are encoded as pairs of string and fret values.

Some other studies have considered an encoding based on *intervals* instead of absolute pitches. In the context of monophonic songs, successive notes can be described in relation to the previous notes through intervals. Among various other viewpoints, intervals can be used to represent symbolic music in a monophonic music genre classification task (Conklin 2013). This interval-based encoding enables the discovery of "musical words" in monophonic music (Park et al. 2024) using Byte-Pair Encoding (Sennrich, Haddow, and Birch 2016), which tend to appear distinctly in specific musical styles.

Considering intervals in the more general context of polyphonic music is less straightforward because of the distinction between *melodic* intervals, between successive notes, and *harmonic* intervals, between simultaneous notes. To the best of our knowledge, only Kermarec, Bigo, and Keller (2022) have attempted a tokenization of polyphonic pieces with pitch values decribed with intervals instead. They have extended the REMI tokenization (Huang and Yang 2020) with *pitch interval* tokens. In particular, they have implemented a *spatial pitch-interval* tokenization strategy which distinguishes between simultaneous and consecutive notes with horizontal and vertical pitch intervals. The study is however limited in several aspects. On the one hand, regarding the modeling of the tokenization strategy, horizontal pitch intervals are applied to the *skyline*[1] stream of notes, and all vertical pitch intervals are assumed to have a negative value. However, the skyline pitches might not always be the optimal reference notes to describe the whole musical content. On the other hand, the experimental framework for downstream tasks is rather simple, musical content is represented as "bag-of-tokens" and is used as input to a logistic regression, which might not take fully advantage of the interval-based tokenization strategy.

In this study, our aim is to extend this work by generalizing the process of encoding symbolic music using intervals instead of absolute pitches, which we call *intervalization*.

Intervalization

In this section, we propose a formalization of the *intervalization* process. Let $\mathbf{x}$ be a sequence of note events:

$$\mathbf{x} = \{e_1, \dots, e_T\}$$

Each note event can be written as $e_k = (p_k, t_k)$ where $p_k \in \{1, \dots, 128\}$ denotes an absolute pitch element and t_k the onset time element associated to the note.

[1]The *skyline* of a musical content includes the highest note at every instant.

Let $\mathbf{x}_{\text{ref}} \subset \mathbf{x}$ be a sub-sequence of notes, chosen as *reference*. $\mathbf{x}_{\text{ref}}$ can correspond for instance to the *bottom-line* of the musical content, the *skyline*, or the melody if available in the data:

$$\begin{aligned}\mathbf{x}_{\text{ref}} &= \{e_{\text{ref}_1}, \dots, e_{\text{ref}_\tau}\} \\ &= \{(p_{\text{ref}_1}, t_{\text{ref}_1}), \dots, (p_{\text{ref}_\tau}, t_{\text{ref}_\tau})\}\end{aligned}$$

$\mathbf{x}_{\text{ref}}$ is chosen to be a monophonic sequence (*i.e.* without simultaneous events): $t_{\text{ref}_j} \neq t_{\text{ref}_{j'}}$ for $j, j' \in \{1, \dots, \tau\}$.

The choice of $\mathbf{x}_{\text{ref}}$ induces a partition of $\mathbf{x}$ into τ subsets of events:

$$\mathbf{x} = S_1, \dots, S_\tau$$

where S_j is defined as the set of notes occurring between e_{ref_j} and $e_{\text{ref}_{j+1}}$:

$$S_j = \{e_{\text{ref}_j}\} \cup \left\{(p,t) \in \mathbf{x} \;\middle|\; \begin{array}{l} t_{\text{ref}_j} \leq t < t_{\text{ref}_{j+1}} \\ (p,t) \neq e_{\text{ref}_j} \end{array}\right\}$$

We call *intervalization* I the process of converting an absolute pitch element into a pitch interval element. The sequence $\mathbf{x}$ is thus transformed into the sequence $\mathbf{x}_{\text{relative}}$:

$$\mathbf{x}_{\text{relative}} = \left\{\mathrm{I}(e_1, \mathbf{x}_{\text{ref}}), \dots, \mathrm{I}(e_T, \mathbf{x}_{\text{ref}})\right\}$$

where, for $e = (p,t) \in S_j$,

$$\mathrm{I}(e, \mathbf{x}_{\text{ref}}) = \begin{cases} \left(\mathrm{I}_{\text{ref}}(p_{\text{ref}_j}, p_{\text{ref}_{j-1}}), t\right) & \text{if } e = e_{\text{ref}_j} \\ \left(\mathrm{I}_{\text{non-ref}}(p, p_{\text{ref}_j}), t\right) & \text{otherwise} \end{cases}$$

where, I_{ref} represents a method specifying the encoding method for reference pitch tokens, while $\mathrm{I}_{\text{non-ref}}$ represents the encoding method for non-reference tokens.

I_{ref} can be chosen as being an encoding using absolute pitches:

$$\mathrm{I}_{\text{ref}}(p_{\text{ref}_j}, p_{\text{ref}_{j-1}}) = p_{\text{ref}_j}$$

or horizontal pitch intervals (Kermarec, Bigo, and Keller 2022) (*i.e.* where each pitch is encoded as a *horizontal* interval with the previous pitch within the reference sequence):

$$\mathrm{I}_{\text{ref}}(p_{\text{ref}_j}, p_{\text{ref}_{j-1}}) = p_{\text{ref}_j} - p_{\text{ref}_{j-1}}$$

In the latter case, the first event is dropped.

Similarly, $\mathrm{I}_{\text{non-ref}}$ can be chosen as being an encoding using absolute pitches:

$$\mathrm{I}_{\text{non-ref}}(p, p_{\text{ref}_j}) = p$$

or vertical pitch intervals (*i.e.* where each pitch is encoded as a *vertical* interval in relation to the simultaneous pitch of the reference sequence):

$$\mathrm{I}_{\text{non-ref}}(p, p_{\text{ref}_j}) = p - p_{\text{ref}_j}$$

We give examples of these intervalization strategies in Figure 1.

Although the choice of $\mathbf{x}_{\text{ref}}$, I_{ref} and $\mathrm{I}_{\text{non-ref}}$ can be much larger as described further, we limit this study to the six intervalization strategies listed in Table 1 applied to the REMI tokenization strategy (Huang and Yang 2020). More precisely, we altered the original REMI tokenization into a "MIDI-score tokenization" (Chou et al. 2024), defined as

Tokenization	$\mathbf{x}_{\text{ref}}$	I_{ref}	$\mathrm{I}_{\text{non-ref}}$
REMI-absolute	–	Abs.	Abs.
REMI-abs.+VPI			
ref-melody	Melody	Abs.	V.P.I.
ref-skyline	Skyline	Abs.	V.P.I.
ref-bottom-line	Bottom-line	Abs.	V.P.I.
REMI-HPI+VPI			
ref-melody	Melody	H.P.I.	V.P.I.
ref-skyline	Skyline	H.P.I.	V.P.I.
ref-bottom-line	Bottom-line	H.P.I.	V.P.I.

Table 1: Tokenizations studied in this work, including the original REMI tokenization (REMI-absolute), and interval-based tokenizations based on REMI. (Abs.: *Absolute pitch encoding* ; V.P.I.: *Vertical Pitch Interval* ; H.P.I.: *Horizontal Pitch Interval*)

a tokenization strategy in which `<Velocity>` tokens are dropped. Indeed, the datasets used for training, as presented below, are not all directly sourced from performance data, but rather from scores or generated data, where velocities do not convey real-world information. Moreover, since we do not perform generative tasks and with the downstream tasks described below being not related to musical interpretation, we assume that velocities have a limited impact on model performances.

We perform the initial tokenization strategy using the MidiTok package (Fradet et al. 2021). We publicly release the datasets, source code, and pre-trained models available at https://algomus.fr/code/.

Evaluation on downstream tasks

In this section, we present an experimental framework that aims to evaluate the impact of intervalization on various MIR tasks. More precisely, our experiments are based on BERT models evaluated on three downstream tasks, namely era classification, start-of-phrase detection, and chord inversion identification.

Downstream tasks

We evaluate the impact of intervalization on three supervised downstream tasks associated with different datasets. Because we propose to evaluate the impact of intervalization when the reference is the melody, all the datasets include music with a homophonic texture, characterized by a single melody supported by an accompaniment (Benward 2018). In particular, we chose these datasets so that the melody is played by a single track during each piece. A quantitative description of the datasets is given in Table 2 (bottom). We focus on the following three tasks:

- **Start-of-phrase detection**. Following the work from Le, Bigo, and Keller (2024), the start-of-phrase detection is a sequence tagging task. The model is trained to classify each token of a sequence as being a start-of-phrase or not. We also follow their framework to build a synthesized dataset of folk tunes with generated piano accompaniment which includes start-of-phrase annotations. In particular, we consider the ESSEN dataset (Schaffrath 1995) which includes folk melodies from 47 countries with phrase annotations. The piano accompaniment is then generated by AccoMontage (Zhao and Xia 2021).
- **Chord inversion identification**. Inspired by the task of figured bass identification (Ju et al. 2020), we implement a chord inversion identification task as a sequence tagging task. The model is trained to classify each token as being part of a root position, first, second, or third inversion chord. We consider the When-in-Rome dataset which includes roman numeral labels (Gotham et al. 2023a) from which only the chord inversion characteristic is extracted. From this dataset, we only kept Bach chorales, from which we assume the melody to be the soprano voice. While the dataset does include much more data than chorales, other instrumentations, such as piano solo or orchestral pieces, do not clearly involve a melodic line or a single instrument playing the melody throughout the whole piece, which can be used as a reference in our evaluation.
- **Era classification**. This task is a binary classification task of full sequences. We consider the OpenScore Lieder dataset (Gotham and Jonas 2022), which includes voice and piano pieces by composers from 1730 to 1949. In particular, we characterize each composer by an average year derived from their birth and death year. We selected the discriminative year for this binary classification task to ensure a balanced dataset: the model is trained to classify a piece as bieng composed either before or after 1865.

Dataset	Task	# tokens (# pieces)
POP909	Pre-training	12.1M (2897)
MTC-Piano	Pre-training	12.4M (18.1k)
String quartets	Pre-training	3.2M (121)
Total	*Pre-training*	*27.8M (21.1k)*
Lieder	Era classification	2.7M (1356)
ESSEN-Piano	Phrase detection	3.4M (6926)
Bach chorales	Chord inv. ident.	204k (371)

Table 2: Description of the datasets used for pre-training and downstream tasks, namely era classification, start-of-phrase detection and chord inversion identification. The count of tokens is given in terms of REMI-absolute tokens.

Model & Pre-training

For performing these tasks, we chose to implement a Transformer encoder-only model on which a classification layer is plugged. Following MidiBERT-Piano (Chou et al. 2024), this last layer is a self-attention layer with a linear layer for the full sequence classification task and a pair of linear layers for the two tagging tasks. For each downstream task, we

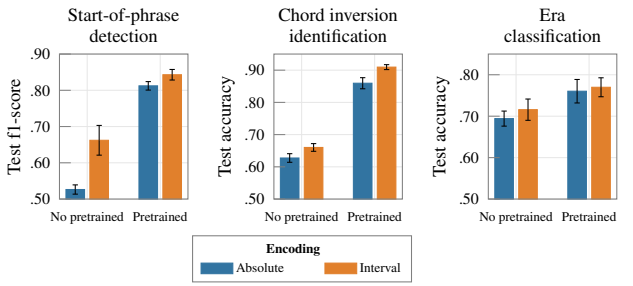

Figure 2: Performance comparison between absolute and intervalized tokenization strategies on the three downstream tasks with non pre-trained and pre-trained models. The intervalized model is based on the reference resulting in the best performance.

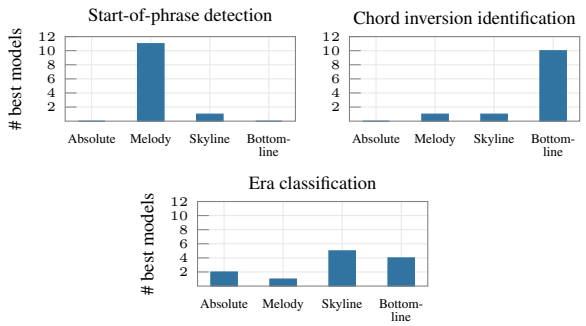

Figure 3: Count of best intervalization references when comparing intervalized models with various references and the absolute. For each task, we consider two pre-trained and two end-to-end models trained on the tokenizations shown in Table 1. This results in 12 comparisons by task, where each comparison involves three intervalization references tested against an absolute tokenization.

evaluate each intervalization strategy on both an end-to-end and a pre-trained model.

We consider three datasets for pre-training for which quantitative descriptions are given in Table 2 (top):

- POP909 (Wang et al. 2020) includes Chinese pop songs with tracks annotated as "melody", "lead" (which we also consider as melody) and "piano".
- MTC-Piano (Le, Bigo, and Keller 2024) is a dataset of Dutch folk melodies extracted from the Meertens Tune Collections (Van Kranenburg et al. 2014) with piano arrangements generated by the AccoMontage model (Zhao and Xia 2021).
- The OpenScore String quartets collection (Gotham et al. 2023b) features string quartets from European composers spanning the classical to late-romantic periods. The melody is approximated as the part played by the first violin.

Using the union of these corpora as a pre-training dataset, we train a Transformer encoder-only model on an unsupervised masked language model pre-training task (Devlin et al. 2019). We pre-train seven models, one for each tokenization strategy (Table 1). The implementation of the model is based on the MidiBERT-Piano model (Chou et al. 2024). However, while the latter consists of 12 layers with 12 heads each, we use a smaller model with 3 layers and 8 heads per layer.

This configuration results in a model with 14M parameters, which is eight times lighter than MidiBERT-Piano. Thus, using our training hyperparameters, two models can fit into a single 12GB Tesla P100 GPU. The models are pre-trained until an early stopping on the validation accuracy of 10 epochs, resulting in approximately one week of pre-training for all the seven models on our hardware.

The pre-trained model weights are not frozen during the fine-tuning process. We stop the training after an early stopping of 3 epochs.

Results

We evaluate models on the above downstream tasks, with various settings regarding intervalization strategies presented in Table 1, namely, an absolute tokenization and two intervalized models with 3 references each. For each downstream task, we evaluate each tokenization strategy on both an end-to-end and a pre-trained model. Each model is trained and evaluated on three seeds of the dataset splits. Therefore, in total, 42 trainings have been performed on each downstream tasks (12 for each type of reference, 6 trainings without intervalization).

Impact of intervalization. Intervalization improves the model performance for all the tasks, both for pre-trained and end-to-end models (Figure 2). However, such improvements range from a marginal 1.2% performance increase in the case of a pre-trained model on era classification to a significant 6% increase with an end-to-end model trained on start-of-phrase detection. Moreover, our results show that pre-training models systematically outperform end-to-end models. On the three tasks, pre-trained model performances are on average 1.20 times better than their end-to-end counterparts, with variations depending on the task.

Impact of intervalization references. For each task, for each split of the dataset among the three considered seeds, we compare twice, once for the end-to-end and once for the fine-tuned, the models trained without intervalization to the three models trained using intervalized tokenizations based on one of the three types of references, for a given setting of $I_{\text{non-ref}}$ and I_{ref}. We then count the number of times that the choice of a particular reference leads to the best result among these four models. In total, we therefore proceed to 12 comparisons per task (3 splits $\times$ (2 pre-trained models + 2 end-to-end models)). Each comparison involves 4 models (3 intervalized + 1 absolute). The counts of the best models associated with their reference are shown in Figure 3.

Models trained with a melodic reference achieve the best performance in 11 of 12 comparisons for the start-of-phrase detection task. In contrast with the skyline or bottom-line references, the melody plays a musically meaningful role. Regarding musical phrases, Arnold Schoenberg stated (Schoenberg, Strang, and Stein 1999, p. 3):

Phrase endings may be marked by a [...] melodic relaxation through a drop in pitch, the use of smaller intervals and fewer notes;

For the era classification task, the choice of the intervalization reference does not show a significant impact on the model performance. Unlike the other tasks, which involve local token tagging, this task focuses on classifying entire sequences into more abstract classes. Such a higher-level task may explain why tokenization plays a less critical role in this context.

Finally, for the task of chord inversion identification, the bottom-line reference leads to the best models in 10 of the 12 comparisons. In the next paragraph, we investigate potential explanations for the effectiveness of this intervalization reference compared to the others.

Musical attributes reflected by the intervalization. We focus on the task of chord inversion identification and we analyze how a tokenizer with a bottom-line reference classifies each token. We study the frequency of vertical pitch tokens classified by the model as root position, first, second, or third inversion (Figure 4). This shows that particular sets of vertical pitch intervals are more prominent within specific inversions. In particular, these more common interval values match the musical definitions of chord inversions. For example, a major (resp. minor) third in combination with a fifth in relation to the bass note define a root position major chord (resp. minor chord) (Figure 4a). The presence of occurrences outside these musical definitions of major/minor/dominant chords can reflect the presence of chord extensions or note embellishments such as passing notes. Moreover, analyzing the proportions of false positives among the predictions can explain some of the model's errors. For example, for first inversions (Figure 4b), the largest numbers of false positives occur with thirds and sixths, which are intervals that also compose chords in root position and second inversion respectively.

Going further, various four-part writing principles (Peters 2016; Benward 2018) can be inferred from these frequency distributions. For example, Figures 4a and 4b show that third intervals (3m and 3M) occur more often with an additional octave than within the same octave. This aligns with voice spacing rules, which typically recommend that bass and tenor voices are not too close. Similarly, Figure 4d shows that there is no octave doubling (P8) of the bass in the case of a third inversion. That also aligns with the fact that the seventh of a dominant seventh chord should not be doubled in four-part harmony.

Conclusion & future directions

In this work, we present a framework for building interval-based tokenization strategies. These tokenizations rely on the choice of a reference within the sequence of notes, an encoding of this reference, and an encoding of non-reference notes. We study the case of tokenizations with an absolute reference and vertical pitch intervals for non-reference notes, as well as tokenizations based on horizontal and vertical pitch intervals. These references are chosen as the melody, skyline and bottom-line notes.

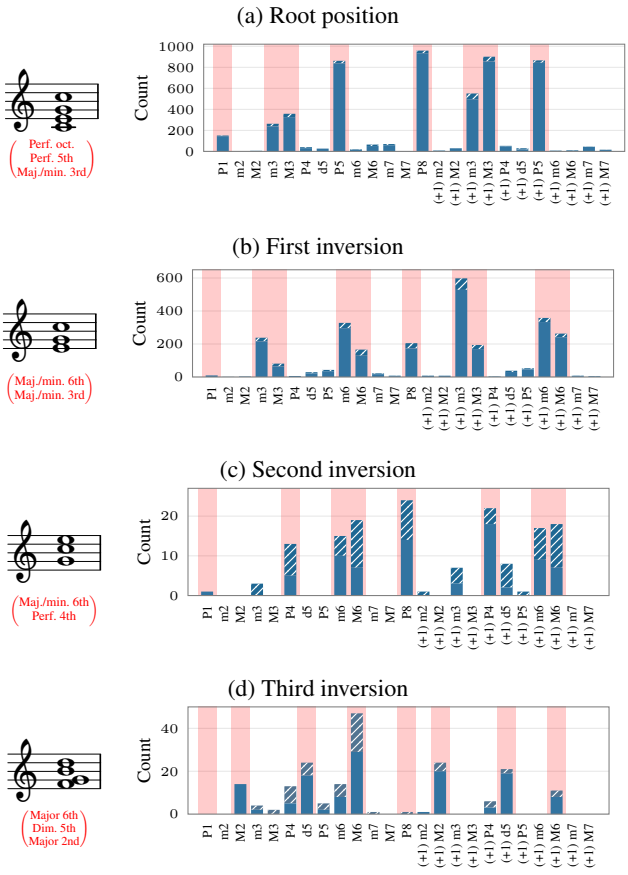

Figure 4: Histograms of vertical pitch interval tokens predicted as root position, first, second or third inversion. The tokenizer is a REMI intervalized tokenizer with the reference being the bottom-line encoded as absolute pitches and non-reference events encoded using vertical pitch intervals. The hatched part of a bar represents the proportion of false positives. Red highlights indicate the intervals that occur in each chord inversion. The notation (+1) indicates an additional octave.

By evaluating these tokenization strategies on three downstream analysis tasks, we show that intervalization improves the models performance compared to an absolute encoding. Moreover, the choice of a specific intervalization reference has an impact on specific tasks and can provide musically meaningful interpretations.

Towards further interval-based tokenization

In this work, we only studied six intervalization strategies. However, several other tokenizations based on intervals can be constructed from the formal description presented above.

We have restricted intervalization strategies to only three types of $\mathbf{x}_{\text{ref}}$. In particular, we have assumed $\mathbf{x}_{\text{ref}} \subset \mathbf{x}$. By releasing this, $\mathbf{x}_{\text{ref}}$ can be chosen as a musically meaningful reference such as a reference sequence composed only of the tonal center of a piece, which may help in tasks such as harmonic analysis.

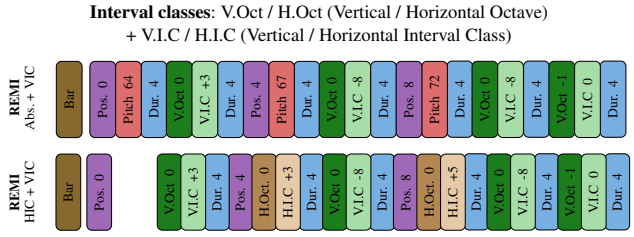

Figure 5: Examples of intervalized tokenizations based on interval classes instead of pitch intervals. (Abs.: *Absolute pitch encoding*)

In addition, we have implemented only two types of interval encodings, namely horizontal and vertical pitch intervals. However, similarly to pitches which can be encoded as `<pitch-class>` and `<octave>` (Li, Li, and Fazekas 2023), an interval can also be considered as `<octave-interval>` and `<interval-class>` as presented in Figure 5. Such an intervalization strategy may allow to directly disentangle octave relations between the notes and interval classes.

Moreover, while interval-based tokenizations can be used for analysis tasks, they cannot all be used for generation purposes, in particular for tokenizations where I_{ref} are horizontal pitch intervals because the generated sequence will not refer to a unique musical sentence. Consequently, tokenizations can be considered where absolute pitches are periodically indicated (for example, at the beginning of each bar), and with intervals representing horizontal/melodic and vertical/harmonic relationships relative to this periodic absolute reference.

Finally, we have only considered tokenizations based on REMI. However, the intervalization process can be applied to any tokenization strategy in which pitches are encoded as absolute values. Time-related tokens (*i.e.* score-based time encoding using `<Bar>` and `<Position>` as REMI, or performance-based time encoding in a MIDI-like tokenization using `<Time-Shift>` (Oore et al. 2018)) are not affected by pitch encodings. Therefore, a study comparing both time and pitch representations may show possible combinations of encodings resulting in a better modeling of symbolic music.

Ethical statement

With the choice of the presented datasets, our work inherently exhibits a bias towards a Western tonal style of music. Moreover, merging several types of genres (folk, classical, pop), as well as the construction of a dataset of folk tunes with pop accompaniment can be questioned. However, restraining the scope of this study can face issues, particularly due to the lack of large-scale annotated symbolic data required to train the presented models.

Similarly to several deep learning studies, our work may have an energy consumption impact due to the needed computation power for model development, training, and evaluation. This impact is, in particular, important for the pre-training phase of our models. Although we did not precisely monitor any hardware power consumption during this study, an approximation[2] of a one-week long pre-training on our hardware reaches a consumption of around 2 $kgCO_2$ eq. However, we try to limit this impact by focusing on small architectures, resulting in fewer parameters in the model as well as shorter training times.

References

Benward, B. 2018. *Music in theory and practice*. McGraw Hill Higher Education.

Chou, Y.-H.; Chen, I.-C.; Ching, J.; Chang, C.-J.; and Yang, Y.-H. 2024. MidiBERT-Piano: Large-scale Pre-training for Symbolic Music Classification Tasks. *Journal of Creative Music Systems*, 8(1).

Conklin, D. 2013. Multiple Viewpoint Systems for Music Classification. *Journal of New Music Research*, 42(1): 19–26.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. 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, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dowling, W. J.; and Fujitani, D. S. 1971. Contour, interval, and pitch recognition in memory for melodies. *The Journal of the Acoustical Society of America*, 49(2B): 524–531.

Fradet, N.; Briot, J.-P.; Chhel, F.; El Fallah-Seghrouchni, A.; and Gutowski, N. 2021. MidiTok: A Python package for MIDI file tokenization. In *International Society for Music Information Retrieval Conference (ISMIR), Late-Breaking Demo Session*.

Fradet, N.; Gutowski, N.; Chhel, F.; and Briot, J.-P. 2023. Impact of time and note duration tokenizations on deep learning symbolic music modeling. In *International Society for Music Information Retrieval Conference (ISMIR)*.

Gotham, M.; Micchi, G.; López, N. N.; and Sailor, M. 2023a. When in Rome: A Meta-corpus of Functional Harmony. *Transactions of the International Society for Music Information Retrieval*.

Gotham, M.; Redbond, M.; Bower, B.; and Jonas, P. 2023b. The "OpenScore String Quartet" Corpus. In *Proceedings of the 10th International Conference on Digital Libraries for Musicology*, DLfM '23, 49–57. New York, NY, USA: Association for Computing Machinery. ISBN 9798400708336.

Gotham, M. R. H.; and Jonas, P. 2022. The OpenScore Lieder Corpus. In Münnich, S.; and Rizo, D., eds., *Music Encoding Conference Proceedings 2021*, 131–136. Humanities Commons. ISBN 978-84-1302-173-7.

Hsiao, W.-Y.; Liu, J.-Y.; Yeh, Y.-C.; and Yang, Y.-H. 2021. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 178–186.

[2] https://mlco2.github.io/impact

Huang, C.-Z. A.; Vaswani, A.; Uszkoreit, J.; Simon, I.; Hawthorne, C.; Shazeer, N.; Dai, A. M.; Hoffman, M. D.; Dinculescu, M.; and Eck, D. 2019. Music Transformer: Generating Music with Long-Term Structure. In *International Conference on Learning Representations (ICLR)*.

Huang, Y.-S.; and Yang, Y.-H. 2020. Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, 1180–1188. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379885.

Ju, Y.; Margot, S.; McKay, C.; Dahn, L.; and Fujinaga, I. 2020. Automatic Figured Bass Annotation Using the New Bach Chorales Figured Bass Dataset. In *Proceedings of the 21th International Society for Music Information Retrieval Conference*, 640–646.

Kermarec, M.; Bigo, L.; and Keller, M. 2022. Improving Tokenization Expressiveness With Pitch Intervals. In *International Society for Music Information Retrieval Conference (ISMIR), Late-Breaking Demo Session*.

Le, D.-V.-T.; Bigo, L.; and Keller, M. 2024. Analyzing Byte-Pair Encoding on Monophonic and Polyphonic Symbolic Music: A Focus on Musical Phrase Segmentation. In *3rd Workshop on NLP for Music and Audio (NLP4MusA)*. San Francisco, United States.

Le, D.-V.-T.; Bigo, L.; Keller, M.; and Herremans, D. 2024. Natural Language Processing Methods for Symbolic Music Generation and Information Retrieval: A Survey. arXiv:2402.17467.

Li, Y.; Li, S.; and Fazekas, G. 2023. Pitch Class and Octave-Based Pitch Embedding Training Strategies for Symbolic Music Generation. In *Proceedings of the 16th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 86–97. Tokyo, Japan: Zenodo.

Oore, S.; Simon, I.; Dieleman, S.; Eck, D.; and Simonyan, K. 2018. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, 32: 955–967.

Park, S.; Choi, E.; Kim, J.; and Nam, J. 2024. Mel2Word: A Text-Based Melody Representation for Symbolic Music Analysis. *Music & Science*, 7.

Peters, J. 2016. *Fundamentals of Writing Four-part Harmony*. CreateSpace Independent Publishing Platform. ISBN 9781536889239.

Sarmento, P.; Kumar, A.; Carr, C.; Zukowski, Z.; Barthet, M.; and Yang, Y.-H. 2021. DadaGP: A dataset of tokenized GuitarPro songs for sequence models. In *International Society for Music Information Retrieval Conference (ISMIR)*.

Schaffrath, H. 1995. The Essen Folksong Collection. In *Center for Computer Assisted Research in the Humanities*.

Schoenberg, A.; Strang, G.; and Stein, L. 1999. *Fundamentals of Musical Composition*. Faber & Faber. ISBN 9780571196586.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725. Berlin, Germany: Association for Computational Linguistics.

Van Kranenburg, P.; de Bruin, M.; Grijp, L. P.; and Wiering, F. 2014. The Meertens tune collections. *Meertens Online Reports*, 2014(1).

Wang, Z.; Chen, K.; Jiang, J.; Zhang, Y.; Xu, M.; Dai, S.; and Xia, G. 2020. POP909: A pop-song dataset for music arrangement generation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 38–45. Montreal, Canada: ISMIR.

Zhao, J.; and Xia, G. 2021. AccoMontage: Accompaniment Arrangement via Phrase Selection and Style Transfer. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021)*, 833–840.