AUDIO-TO-SCORE JAZZ SOLO TRANSCRIPTION WITH THE RHYTHM PERCEIVER

Ivan Shanin[1], *Xavier Riley*[2], *Simon Dixon*[1]

[1]Centre for Digital Music, Queen Mary University of London, UK
[2]Sound Patrol Inc., US

ABSTRACT

Monophonic audio-to-score transcription remains a challenging task, especially for improvised jazz solos. Prior work commonly follows a modular pattern: combining audio-to-performance-MIDI transcription, state-of-the-art beat tracking, and a post-processing algorithm that assigns metrical durations to notes (note values). This approach is prone to error accumulation at every step. Motivated by how expert transcribers work, we propose an end-to-end approach that inverts the logical order of subtasks: first, we model rhythm perception by transcribing the rhythmic structure of the melody at the beat level; then, we estimate the pitch for each note within the predicted rhythmic structures. Technically, our model introduces a rhythm-aware transformer architecture with perceiver-style cross-attention that jointly aligns frame-level acoustic features with beat-synchronous rhythmic embeddings. The proposed solution achieves state-of-the-art results on a challenging corpus of alto saxophone solos by Charlie Parker (the "Omnibook").

Index Terms— Audio-to-score transcription, rhythm perception, jazz improvisation

1. INTRODUCTION

The main goal of this work is to build a transcription system that assists the automatic analysis of improvised monophonic jazz solos, a domain that is especially challenging due to sophisticated idiomatic rhythmic concepts, rich timbral variety, advanced harmonies, and technically demanding melodic patterns. This analysis should decouple expressive techniques (rhythmic and timbral) from music-theoretical concepts (harmonic and melodic patterns). For the latter, we need an audio-to-score system that ignores expressive techniques and renders musical content in a unified, standardized score notation. Most recent progress in automatic music transcription has been made in the audio-to-performance-MIDI[1] setting [1, 2]. These systems do not provide rhythmic annotation, which obstructs computer-assisted musicological research that typically requires score-level notation. Several works address this gap with performance-MIDI-to-score solutions [3, 4]. However, modular solutions suffer from error accumulation when the performance MIDI is derived automatically from an audio recording. Another line of work develops end-to-end audio-to-score transcription systems [5, 6] that typically include a transformer decoder which autoregressively generates a tokenized musical score from audio features. This approach requires a significant amount of training data, and autoregressive generation does not guarantee a valid score.

[1]The MIDI (Musical Instrument Digital Interface) protocol captures the times at which each note starts and ends, along with its discrete pitch. Performance MIDI uses physical time (e.g. in seconds), while score MIDI interprets timing in musical units (e.g. quarter notes), as used in Western music notation.

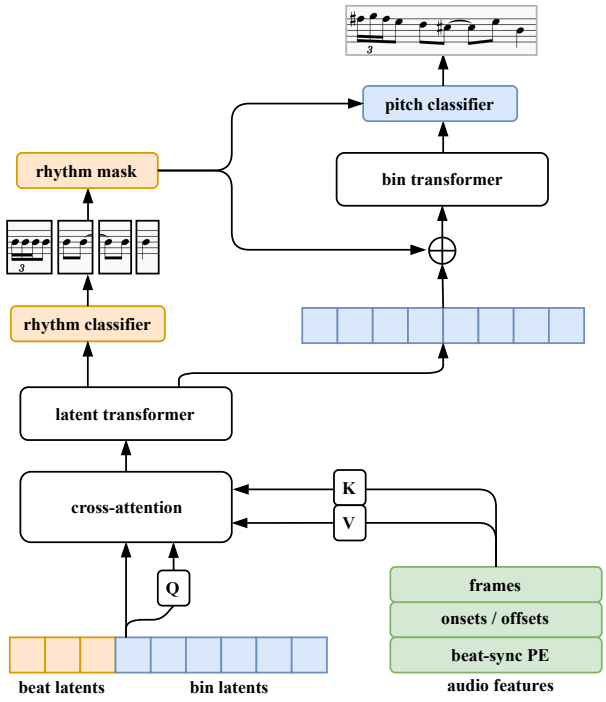

Fig. 1. Rhythm Perceiver architecture (for details see Sec. 3)

In this paper we propose a novel approach for audio-to-score transcription of jazz solos inspired by human perception. It includes a tokenization scheme based on rhythmic patterns and a transformer architecture for structured generation of monophonic musical scores. Since it explicitly models rhythmic perception in jazz and conceptually follows the Perceiver [7] family of architectures, we call it the "Rhythm Perceiver" (Fig. 1).

We make the following assumptions: (a) input audio segments are improvised solo passages where a woodwind soloist plays alongside a rhythm section (piano or guitar, bass, and drums); (b) music is in a $\frac{4}{4}$ time signature; (c) melodic and rhythmic content lies within the mainstream jazz idiom defined in the 1950s (bebop). These assumptions allow us to rely on existing high-quality beat tracking and source separation methods. Our work is motivated by the following observations about human perception: (a) expert transcribers typically recognize the rhythm first and fill in exact pitch values later; (b) melodic patterns in jazz are somewhat invariant to tempo, so that material played at one rate can often be played at double speed (called "double time") as well. Although we apply this approach to mono-

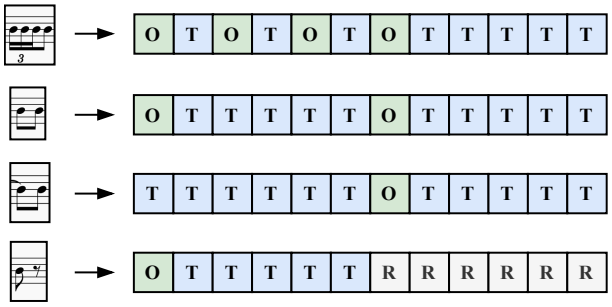

Fig. 2. Examples of beat signatures (rhythm masks) compatible with the proposed score tokenization

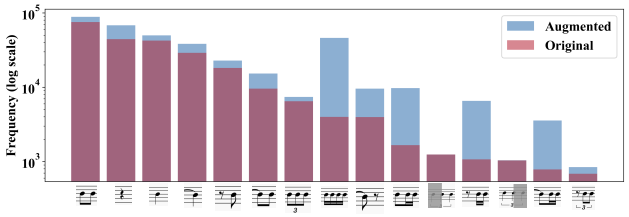

Fig. 3. Histogram of beat-signature classes extracted from Filosax (15 most frequent classes before and after augmentation). Frequencies generally follow Zipf's law. After "double-time" augmentation, beat signatures based on 16th notes become significantly more frequent (because, before augmentation, classes based on 8th notes dominate shorter note values).

phonic jazz solos, it could be adapted to other data-sparse musical styles with established rhythmic idioms.

2. RELATED WORK

The domain of improvised jazz solos is data-sparse. In the seminal "Jazzomat" project [8], 456 solos were transcribed by expert annotators, generating high-quality performance MIDI for nearly 10 hours of jazz solos (known as the "Weimar Jazz Database" or WJD). Another available dataset is Filosax [9], which contains 48 jazz standards performed by 5 tenor saxophonists over a backing track, resulting in 24 hours of audio. Importantly, this dataset contains accurate score notation of the performances, a rare and valuable resource for jazz improvisation. Finally, the Omnibook dataset [10, 11] contains a well-studied collection of 50 relatively short improvised solos by Charlie Parker, comprising roughly 1 hour of audio. Similar to Filosax, this dataset contains accurate score annotations of the performances, but the material is significantly more challenging because the rhythmic and melodic content is more advanced and the recording quality is lower. The PiJAMA [12] and Jazz Trio [13] datasets contain automatically generated performance MIDI for many solo piano and piano trio recordings, but they do not contain score annotations.

The following approaches have been applied to automatically generate score notation for jazz solos: In [8], expert annotators transcribed solos into performance MIDI, which were later converted to score notation using the proposed FlexQ algorithm, performing a grid search for the optimal rhythmic annotation that minimizes a hand-crafted error function. Another recent approach [11] used a convolutional-recurrent neural network to generate performance MIDI, which was converted to score notation using the qparse algorithm [14], solving a similar optimization problem via Weighted Tree Automata and dynamic programming. These approaches share a limitation: they are generally not suitable for widely used rhythmic jazz idioms such as swing, syncopation, or playing behind the beat.

Our approach shares similarities with SheetSage [15, 16]: we also use a pretrained audio encoder and a transformer encoder-only backbone. Related ideas with different tokenization schemes were explored for singing melody transcription [17] and audio-to-score systems [6]. Similar to [2], we generate the score hierarchically (first note values, then pitches), but [2] targets performance MIDI generation, whereas we generate score notation. Our original contribution is the explicit modeling of beat-level rhythmic units, which provides additional insights into the structure of improvised jazz solos.

3. METHODOLOGY

Score and Rhythm Tokenization. Recent work [15, 17] approaches score generation as token prediction on a fixed metrical grid, where tokens contain pitch and onset information. A common choice for the grid resolution is the sixteenth note, which is unable to encode triplet eighth notes, a defining element of jazz phrasing. Therefore we use a more fine-grained tokenization scheme, dividing each beat into 12 equal parts ("bins"). Each bin is assigned one of 130 token classes: 128 MIDI pitches (onset tokens), 1 rest token, and 1 tie token. The first bin of each note is an onset token; subsequent bins for that note are ties, and unvoiced bins (gaps between notes) are rests. This tokenization encodes the following monophonic note values: whole, half, dotted half, quarter, dotted quarter, triplet quarter, 8th, dotted 8th, 16th, 8th-note triplets, and 16th-note triplets. Additionally, we introduce the concept of a "beat signature": a sequence of 12 bins representing the rhythmic pattern within a beat, where the pitched onset tokens are replaced by a single generic onset token (see Fig. 2).

We motivate this by the fact that jazz has an established rhythmic idiom; to correctly define note characteristics we must know their rhythmic context. For example, 8th notes are commonly played with rhythmic displacement ("swing" feel), where instead of dividing the beat into two equal 8ths, it is divided into longer and shorter parts. The ratio between these parts (the swing ratio) is usually stable over a piece but changes with tempo and style. A common swing ratio at medium tempos is 2:1, approaching 1:1 as tempo increases [18]. A more nuanced example is a four-note pattern where the first three notes are 16th-note triplets and the last is an 8th note. This pattern can be confused with four 16th notes, especially at fast tempos. However, these rhythmic structures are not interchangeable: the first is commonly used in phrases of swung 8ths to embellish a downbeat 8th note, whereas the second is used in lines of 16ths (often referred to as a "double-time" feel). An example of both patterns appears in the melody of Charlie Parker's "Confirmation" (second beat of bar 2 and first beat of bar 21, respectively). In these examples, actual note durations depend not only on the nominal value but also on position in the rhythmic grid. This position can also be affected by expressive timing, such as playing slightly delayed or (less commonly) ahead of the beat, which is characteristic of certain jazz styles. These style-dependent rhythmic idioms define a gap between musical content and its score representation. Readers of score notation are expected to have sufficient stylistic knowledge to interpret the score accurately.

To solve the rhythm classification task, we first define a set

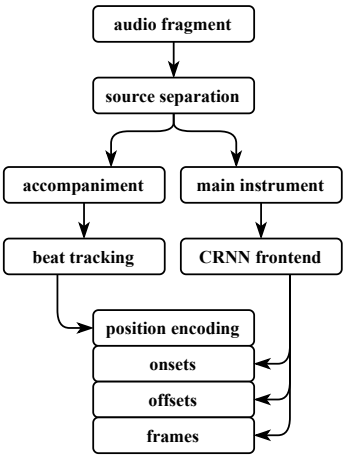

Fig. 4. Workflow of audio feature extraction

of beat-signature classes compatible with the proposed tokenization scheme. We extract these classes from the Filosax dataset as follows: for each beat, we construct a rhythm signature as a sequence of 12 bins with onset, tie, or rest tokens. If this construction is impossible due to unsupported note values, we assign a special "unsupported" class to the beat. We then compute class frequencies and assign a special "rare" class to beats with less than 30 occurrences. After this procedure we obtain 42 unique beat-signature classes (plus two special classes). We also apply a "double-time augmentation" by dividing each note value by two. This is motivated by the common practice of playing similar material at different time resolutions (the same melodic line may be played with 8ths or 16ths, depending on tempo and harmony). Figure 3 shows the histogram of the most common beat-signature classes before and after augmentation.

Audio Encoder. To extract audio features we design a modular frontend consisting of source separation, beat tracking, and onset/frame recognition (Fig.4). Since we mainly work with saxophone and trumpet solos, we use an open-source source-separation system [19] trained to produce two stems: a single wind instrument and the accompaniment. In the small-ensemble jazz tradition, a soloist is typically accompanied by a "rhythm section" (piano, double bass, and drums) whose role is to outline the rhythmic and harmonic structure while interacting with the soloist. While the soloist's lines may be rhythmically sophisticated (as discussed above), the accompaniment tends to be rhythmically clearer. We exploit this in the next step, beat tracking. We apply the open-source Madmom beat tracker [20] with the following settings: (i) we apply it to the separated accompaniment rather than the original mix; (ii) we set the "transition lambda" parameter high, since we do not expect significant changes in tempo or meter within the analyzed fragment.

For onset and frame extraction we employ the strategy from [1], later used in a state-of-the-art AMT system for jazz solos [11]. It consists of several convolutional audio encoders followed by gated recurrent units (GRUs), trained to predict frame-level pianoroll and onsets from the mel spectrogram of the original audio.

Shared Position Encoder. We propose a beat-synchronized hierarchical position encoding scheme for both modalities (score tokens and audio-extracted features) with three levels of granularity. The high-level position encodes the bar index in the analyzed fragment; the medium-level position encodes the beat index within a bar; the low-level position encodes position between neighboring beats. Since both modalities share the high- and medium-level structures, we propose shared encodings for these levels, while low-level positions in the score (bin index) and audio (frame index) are encoded independently.

Let N_{bar} be the total number of bars in the analyzed fragment, N_{beat} the number of beats per bar, $N_{bin} = 12$ the number of score bins per beat, and D_{pe} the positional-encoding dimension. To encode bar and beat positions we use N_{bar} and N_{beat} trainable embedding vectors of dimension D_{pe}. To encode low-level score positions we use a set of N_{bin} trainable embedding vectors of the same dimension. We also encode the position of special tokens that define the beat signature for each beat via an additional trainable embedding vector of dimension D_{pe}, used homogeneously with bin position encodings.

For low-level audio positions we take a different approach, since the number of spectrogram frames between beats varies with tempo. For each frame f_i we compute $\omega_i \in [0, 1)$ as the time from the frame center to the previous beat, divided by the time between the previous and the next beats.

Let $\{\beta_{\ell,0}, \beta_{\ell,1}, \beta_{\ell,2}, \beta_{\ell,3}\}$ denote the beat times within bar ℓ, and $\beta_{\ell,4} = \beta_{\ell+1,0}$ denotes the first beat time of the next bar (assuming that bar ℓ is in $\frac{4}{4}$ time signature). For a frame centered at time t_i, we define the bar index ℓ_i and the beat index $b_i \in \{0, 1, 2, 3\}$ such that $t_i \in [\beta_{\ell_i,b_i},\, \beta_{\ell_i,b_i+1})$, and the normalized phase

$$\omega_i = \frac{t_i - \beta_{\ell_i,b_i}}{\beta_{\ell_i,b_i+1} - \beta_{\ell_i,b_i}} \in [0, 1)\,. \tag{1}$$

We then apply a Fourier feature mapping to ω_i to obtain a vector of dimension $2K$, which is then passed through a multi-layer perceptron to obtain a vector of dimension D_{pe}:

$$\mathbf{f_i} = \big[\sin(2\pi k\,\omega_i),\ \cos(2\pi k\,\omega_i)\big]_{k=1}^{K} \in \mathbb{R}^{2K}, \tag{2}$$

$$PE_i^{frame} = \mathrm{MLP}(\omega_i, \mathbf{f_i}) \in \mathbb{R}^{D_{pe}}. \tag{3}$$

This encoding helps distinguish sequences of different note values; for example, a sequence of triplet 8ths will have onsets near peaks of $\cos(2\pi \cdot 3\,\omega_i)$, while a sequence of 16ths will have onsets near peaks of $\cos(2\pi \cdot 4\,\omega_i)$. We set $K = 6$, as we do not expect more than 6 onsets within a single beat.

Finally, the positional encoding for an audio frame (and for a score bin) is the sum of all three components:

$$PE_i^{audio} = PE_i^{frame} + PE_i^{bar} + PE_i^{beat} \in \mathbb{R}^{D_{pe}}, \tag{4}$$

$$PE_i^{score} = PE_i^{bin} + PE_i^{bar} + PE_i^{beat} \in \mathbb{R}^{D_{pe}}. \tag{5}$$

Rhythm Perceiver Architecture. The Perceiver architecture [7] is transformer-based, with an early cross-attention layer that projects inputs into a latent space (potentially better suited to the task), followed by a transformer encoder operating on the latents. For Rhythm Perceiver (Fig.1) we design the latent space as a concatenation of bin and rhythm latents, which are then processed by a shared transformer encoder. After that, rhythm and bin latents are processed by two branches. The rhythm branch consists of a linear classifier that predicts a beat-signature class for each beat. The bin branch consists of an additional transformer encoder followed by a linear classifier that predicts a pitch value for each bin. We also condition the bin branch on the rhythm branch by adding a rhythm mask to the bin latents immediately after the shared encoder.

Table 1. Ablations on the Filosax test set

Variant	Pianoroll Acc ↑	Rhythm Acc ↑	Pitch Acc ↑	Onset F1 ↑	Onset Recall ↑	Voiced Acc ↑
Perceiver	0.90	0.83	0.95	0.91	0.90	0.97
+ Rhythm Classifier	0.91	0.86	0.94	0.92	0.92	0.97
+ Mask Injection	0.91	0.87	0.95	0.92	0.92	0.97
+ DT augmentation (final)	**0.92**	**0.87**	**0.95**	**0.93**	**0.93**	**0.97**

Table 2. Audio-to-Score evaluations on the Omnibook dataset

Variant	Pianoroll Acc ↑	Rhythm Acc ↑	Onset F1 ↑	Multi-Pitch Acc ↑	MV2H ↑	Note Insertion ↓	Note Deletion ↓
CRNN + qparse [11]	0.19	0.18	0.48	N/A	N/A	23.26	54.74
Perceiver	0.51	0.48	0.81	0.64 ± 0.13	0.92 ± 0.03	17.21	61.13
Rhythm Perceiver	**0.53**	**0.53**	**0.83**	0.64 ± 0.13	0.92 ± 0.03	**20.42**	**53.36**

A rhythm mask is created from rhythm branch predictions by unfolding each predicted class into a sequence of 12 bins with onset, tie, or rest tokens. Each token is replaced with one of three learnable embedding vectors of the same dimension as the bin positional encoding. We also use this mask when computing the pitch loss: cross-entropy is calculated only at note onset positions indicated by the mask. During training we use teacher forcing, supplying the ground-truth rhythm mask instead of predictions. The model is trained end-to-end with the following loss:

$$\mathcal{L}_{\text{RP}} = \text{CE}(\hat{y}, y|r) + \lambda_{\text{rhythm}} \cdot \text{CE}(\hat{r}, r), \tag{6}$$

where $\hat{y}$ is the predicted pitch, y is the ground truth pitch, $\hat{r}$ is the predicted rhythm class, r is the ground truth rhythm class, and λ_{rhythm} weights the rhythm loss. We set $\lambda_{\text{rhythm}} = 1.0$.

At inference time we first predict rhythm classes for each beat, then read out pitch predictions at note-onset positions indicated by the rhythm mask, generating a score in the same tokenization scheme as during training. To convert this representation to MusicXML format we use standard methods from the "music21" library [21].

This two-stage approach has two benefits: (i) it mirrors human transcription practice, recognizing rhythm first and then filling in pitches; (ii) it enables structured generation of score notation, where errors have local scope and the produced score is guaranteed to be syntactically valid.

4. RESULTS AND DISCUSSION

The CRNN frontend was trained on a combination of the Filosax and WJD datasets. To match the inference domain, Filosax solo tracks were mixed with background tracks at various loudness ratios and then separated using UVR [19]. We selected 396 WJD tracks containing wind instrument solos and performed source separation as well. Onset and offset encoders are trained with a regression loss (mean squared error). The objective is to estimate the distance between each frame center and the nearest onset or offset event. To improve robustness we apply random key transpositions to avoid overfitting to common keys.

Rhythm Perceiver was trained only on Filosax (split: compositions 1-44 for training, 45 for validation, 46-48 for testing). We use 256-dimensional embeddings; the shared latent transformer has 6 layers and 8 attention heads; the bin transformer has 4 layers and 8 heads; dropout is 0.1 and the learning rate is 10^{-4}.

In the ablation study (Tab. 1), we evaluate the effect of explicit beat-signature modeling: (i) Perceiver without rhythm supervision (bin branch only); (ii) rhythm supervision without cross-branch conditioning (mask used only in the loss); (iii) full Rhythm Perceiver with cross-branch conditioning; (iv) Rhythm Perceiver with double-time augmentation. We report beat-level rhythm accuracy, note-level onset precision, recall, and F1; pitch accuracy for correctly predicted note onsets; and score-based pianoroll accuracy (pianorolls are generated from the predicted score and aligned with the ground truth at the same BPM). Onset precision and recall are calculated with zero tolerance.

The results show that the main source of improvement is structured prediction: in the bin-only model, it is not guaranteed that each note begins with an "onset" token, which lowers onset recall. Beat-level rhythm accuracy also improves consistently with the addition of rhythm supervision.

To compare our approach with the state of the art, we reproduce the setup from [11], using the same subset of Omnibook tracks. Edit-based metrics ("note insertion" and "note deletion") are directly comparable with [11], although we found their performance highly sensitive to the quality of beat tracking. Since beat tracking is outside the scope of this work, we divided the test set into "easy" and "hard" subsets: the "hard" subset contains tracks with fast tempos, poor recording quality, and rhythmically complex accompaniments (roughly 40% of the test set), while the "easy" subset consists of tracks with reliable automatically generated beat tracking (see Sec. 3). The remaining metrics in Tab. 2 are reported on the "easy" subset (60% of the original test set). Following [5], we also report the MV2H metric [22], along with its most relevant component for our task, "multi-pitch accuracy."

These results demonstrate a clear improvement over prior work and highlight the benefits of explicitly modeling beat-level rhythmic units, which is our main original contribution. However, the domain mismatch between Filosax and the Omnibook is evident: pianoroll and rhythm accuracies decrease substantially. This can be attributed to the fact that our model is optimized for rhythmic structures within Filosax, which primarily covers mainstream jazz. It is not designed to capture complex polyrhythms or more abstract avant-garde styles. The Charlie Parker Omnibook contains denser, more inventive, and technically challenging material than Filosax. Addressing this gap in available transcribed jazz solos, as well as extending our approach to polyphonic music, remain important directions for future work.

5. REFERENCES

[1] Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, and Yuxuan Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 3707–3717, 2021.

[2] Dichucheng Li, Yongyi Zang, and Qiuqiang Kong, "Piano transcription by hierarchical language modeling with pretrained roll-based encoders," in *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*. 2025, pp. 1–5, IEEE.

[3] Lele Liu, Qiuqiang Kong, Veronica Morfi, and Emmanouil Benetos, "Performance MIDI-to-score conversion by neural beat tracking," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, Preeti Rao, Hema A. Murthy, Ajay Srinivasamurthy, Rachel M. Bittner, Rafael Caro Repetto, Masataka Goto, Xavier Serra, and Marius Miron, Eds., 2022, pp. 395–402.

[4] Tim Beyer and Angela Dai, "End-to-end piano performance-MIDI to score conversion with transformers," in *Proceedings of the 25th International Society for Music Information Retrieval Conference, ISMIR 2024, San Francisco, California, USA and Online, November 10-14, 2024*, Blair Kaneshiro, Gautham J. Mysore, Oriol Nieto, Chris Donahue, Cheng-Zhi Anna Huang, Jin Ha Lee, Brian McFee, and Matthew C. McCallum, Eds., 2024, pp. 319–326.

[5] Juan Carlos Martinez-Sevilla, María Alfaro-Contreras, Jose J. Valero-Mas, and Jorge Calvo-Zaragoza, "Insights into end-to-end audio-to-score transcription with real recordings: A case study with saxophone works," in *24th Annual Conference of the International Speech Communication Association, Interspeech 2023, Dublin, Ireland, August 20-24, 2023*, Naomi Harte, Julie Carson-Berndsen, and Gareth Jones, Eds. 2023, pp. 2793–2797, ISCA.

[6] María Alfaro-Contreras, Antonio Ríos-Vila, Jose J. Valero-Mas, and Jorge Calvo-Zaragoza, "A transformer approach for polyphonic audio-to-score transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*. 2024, pp. 706–710, IEEE.

[7] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira, "Perceiver: General perception with iterative attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4651–4664.

[8] Martin Pfleiderer, Klaus Frieler, Jakob Abeßer, Wolf-Georg Zaddach, and Benjamin Burkhart, *Inside the Jazzomat: New Perspectives for Jazz Research*, Schott Campus, 2017.

[9] Dave Foster and Simon Dixon, "Filosax: A dataset of annotated jazz saxophone recordings," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, Jin Ha Lee, Alexander Lerch, Zhiyao Duan, Juhan Nam, Preeti Rao, Peter van Kranenburg, and Ajay Srinivasamurthy, Eds., 2021, pp. 205–212.

[10] Ken Déguernel, Emmanuel Vincent, and Gérard Assayag, "Using multidimensional sequences for improvisation in the OMax paradigm," in *13th Sound and Music Computing Conference*, 2016.

[11] Xavier Riley and Simon Dixon, "Reconstructing the Charlie Parker Omnibook using an audio-to-score automatic transcription pipeline," in *Sound and Music Computing Conference (SMC)*, 2024, pp. 546–552.

[12] Drew Edwards, Simon Dixon, and Emmanouil Benetos, "PiJAMA: Piano jazz with automatic MIDI annotations," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 6, no. 1, pp. 89–102, 2023.

[13] Huw Cheston, Joshua L. Schlichting, Ian Cross, and Peter M. C. Harrison, "Jazz trio database: Automated annotation of jazz piano trio recordings processed using audio source separation," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 7, no. 1, 2024.

[14] Francesco Foscarin, Florent Jacquemard, Philippe Rigaux, and Masahiko Sakai, "A parse-based framework for coupled rhythm quantization and score structuring," in *Mathematics and Computation in Music - 7th International Conference, MCM 2019, Madrid, Spain, June 18-21, 2019, Proceedings*, Mariana Montiel, Francisco Gómez-Martin, and Octavio Alberto Agustín-Aquino, Eds. 2019, vol. 11502 of *Lecture Notes in Computer Science*, pp. 248–260, Springer.

[15] Rodrigo Castellon, Chris Donahue, and Percy Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, Jin Ha Lee, Alexander Lerch, Zhiyao Duan, Juhan Nam, Preeti Rao, Peter van Kranenburg, and Ajay Srinivasamurthy, Eds., 2021, pp. 88–96.

[16] Chris Donahue and Percy Liang, "Sheet sage: Lead sheets from music audio," *Proc. ISMIR Late-Breaking and Demo*, 2021.

[17] Leekyung Kim, Sungwook Jeon, Wan Heo, and Jonghun Park, "Note-level singing melody transcription for time-aligned musical score generation," *CoRR*, vol. abs/2502.12438, 2025.

[18] A. Friberg and A. Sundström, "Swing ratios and ensemble timing in jazz performance: Evidence for a common rhythmic pattern," *Music Perception*, vol. 19, no. 3, pp. 333–349, 2002.

[19] "Ultimate vocal remover GUI," `https://github.com/Anjok07/ultimatevocalremovergui`, 2023, Version 5.6; MIT License.

[20] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer, "madmom: A new Python audio and music signal processing library," in *Proceedings of the 24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, 10 2016, pp. 1174–1178.

[21] Michael Scott Cuthbert and Christopher Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," in *11th International Society for Music Information Retrieval Conference*, 2010, pp. 637–642.

[22] Andrew McLeod and Mark Steedman, "Evaluating automatic polyphonic music transcription," in *19th International Society for Music Information Retrieval Conference*, 2018, pp. 42–49.