Computational Approach to Track Beats in Improvisational Music Performance

Xianghui Xie
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium

Jared Houghtaling
Dept. of Electrical Eng. (ESAT-STADIUS)
KU Leuven
Leuven, Belgium

Katrien Foubert
Luca School of Arts
Leuven, Belgium

Toon van Waterschoot
Dept. of Electrical Eng. (ESAT-STADIUS)
KU Leuven
Leuven, Belgium

***Abstract*—Beat tracking, or identifying the temporal locations of beats in a musical recording, has a variety of applications that range from music information retrieval to machine listening. Algorithms designed to monitor the tempo of a musical recording have thus far been optimized for music with relatively stable rhythms, repetitive structures, and consistent melodies; these algorithms typically struggle to follow the free-form nature of improvisational music. Here, we present a multi-agent improvisation beat tracker (MAIBT) that addresses the challenges posed by improvisations and compare its performance with other state-of-the-art methods on a unique data set collected during improvisational music therapy sessions. This algorithm is designed for MIDI files and proceeds in four stages: (1) preprocessing to remove notes that are timid and overlapping, (2) clustering of the remaining notes and subsequent ranking of the clusters, (3) agent initialization and performance-based selection, and (4) artificial beat insertion and deletion to fill remaining beat gaps and create a comprehensive beat sequence. This particular method performs better than other generic beat-tracking approaches for music that lacks regularity; it is thus well suited to applications where unpredictability and inaccuracy are predominant, such as in music therapy improvisation.**

***Index Terms*—Beat Tracking, Improvisation, Multi-Agent, MIDI Data, Music Therapy**

I. Introduction

Beat tracking refers to the process of monitoring the tempo (*i.e.,* beats per minute) and specific timestamp of each beat in a piece of music. The method is employed within a variety of applications that include music information retrieval [1], audio editing [2], and music classification [3]. Alongside these standard applications, beat tracking also makes it possible to analyze, understand, and improve improvisational sessions in the context of music therapy, wherein music is performed spontaneously by a patient without prior preparation, and in some cases, without prior musical experience. Even in the latter scenario, the performance of the patient contains information that relates to underlying affective and communicative processes that occur during music making [4]. Music performance has, accordingly, been employed as a therapeutic tool for more than 200 years [5], but efforts to analyze these performances quantitatively are both relatively recent and relatively sparse; Hunt *et al.* first developed a computational database and audio player to assist music therapists in the year 2000 [6], [7], after which Luck *et al.* used an automated method to detect rhythmic pulsations during therapy sessions and then related those rhythms to the severity of mental retardation in patients [8]. Streeter *et al.* expanded on this work, characterizing the effects of mental illnesses on tempo and other features extracted from music therapy sessions [9], and meanwhile Erkkilä *et al.* developed a music therapy toolbox (MTTB) to assist clinicians and researchers in their analyses of improvisational sessions [10]–[13]. The MTTB, however, relies on autocorrelation to calculate pulse clarity through onset-curve periodicity, and such an approach struggles to accommodate irregularities in improvisational performance.

Fig. 1 gives an overview of a piano-based improvisational music therapy session and a process flow for analyzing data from such a session. Recently, we (Foubert *et al.*) manually assigned beat timestamps to notes played by a therapist and then used those timestamps to analyze notes played simultaneously by the patient, which was both tedious to perform on hours of musical recordings and was prone to user error [14]. Moreover, errors in beat assignments have critical consequences; metrical deviations (MD, *i.e.,* temporal deviation between patient's note and the underlying eighth-note beat of the therapist to which it is closest) of tens of milliseconds can reveal important insight into disease state. A method for tracking the beats of music therapy performances and then quantifying MD both rapidly and accurately could assist therapists in their psychiatric evaluations, and by extension, could assist patients undergoing therapy; such a task, however, is non-trivial due to the inherent irregularity of this type of musical performance.

In previous work [14], we collected and labeled an experi-

This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven internal funds C2-16-00449, VES/16/032, FWO Large-scale research infrastructure "The Library of Voices" (I013218N). The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program / ERC Consolidator Grant: SONORA (no. 773268), and a postdoctoral fellowship from the Belgian American Educational Foundation (J.H.). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

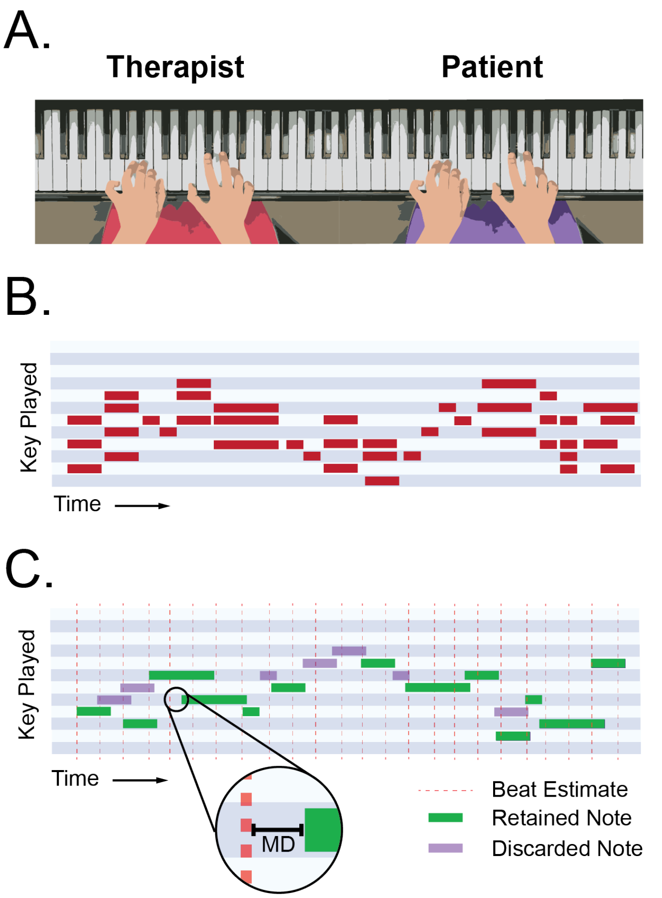

Fig. 1. Overview of music therapy improvisation. A) Schematic of a one-on-one, piano-based therapy session in which the therapist plays a specific combination of structured and unstructured accompaniments, and the patient plays in tandem. B) Example of a MIDI file containing therapist notes – tracking the instantaneous tempo of the therapist note events is the main focus of this study. C) Example of a MIDI file containing patient notes – estimated beat timestamps determined from the therapist track are used to calculate metrical deviations (MD) between the patient notes and the composition beat. These MD's can then be used to infer therapeutic progress [14].

mental dataset during music therapy sessions using a Yamaha Disklavier MPX70 Piano. This particular piano contains an array of optical sensors, each monitoring a single key, that communicate through a USB MIDI interface (Motu Midi Express 128); we recorded improvisational sessions using Logic Pro X (Max system) and then exported the resulting MIDI files [8], [9], [14]. Existing methods to track beats within MIDI files fall into four main classes: (1) multi-agent methods, which estimate tempos from beat intervals and select potential beats from note event sequence using tempo hypothesis [15], [16] (2) oscillator methods, which adjust a period and phase to synchronize with a sequence of events [17] (3) autocorrelation methods, which detect beats by first converting MIDI information into onset curves and then applying autocorrelation analysis [8], [18] and (4) probabilistic methods, which utilize statistical approaches like Monte Carlo sampling [19], Bayesian frameworks and Kalman filters [20], or Hidden Markov Models (HMM) [21] to represent temporal distributions in music pieces. The vast majority of these methods rely on repeated patterns and regularity, both which are often missing in improvisational music. Here, we present an alternate version of a multi-agent algorithm first presented by Dixon *et al.* [15] that we have modified to overcome the challenges presented by musical improvisations and that is suitable for evaluating MIDI-based music sequences collected during music therapy sessions.

II. Methodology

We provide an overview of our algorithm – the Multi-Agent Improvisation Beat Tracker (MAIBT) – in Fig. 2. The algorithm accepts MIDI data as input and outputs an estimated sequence of beat timestamps. We implemented two additional steps relative to the original algorithm [15] – namely preprocessing and beat insertion and deletion – in order to account for irregularities in improvisation, and we also modified the original protocol for calculating agent scores, all of which we describe in the following subsections.

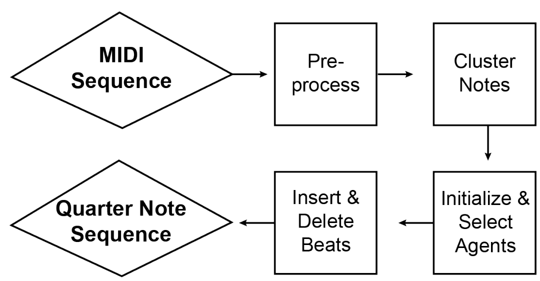

Fig. 2. Overview of the process flow of the MAIBT, beginning with a MIDI file as input and resulting in a sequence of quarter note beats.

A. Preprocessing

We introduce a preprocessing step to account for two forms of irregularity in improvisational recordings: (1) lack of defined or repetitive melodies (*i.e.*, unpredictability), and (2) inaccurate timing and playing of notes (*i.e.*, inaccuracy). Experienced musicians (here, the therapists) tend to play mistaken or overlapping notes with less velocity than beat-setting notes. Here, velocity is a quality of a note recorded in a MIDI file that describes the manner in which it was played (*i.e.,* forceful playing produces high velocities, delicate playing produces low velocities). Because we are primarily concerned with the timestamp of each beat, we can represent several simultaneous notes as the single note in that group with highest velocity. To perform this replacement, we evaluate note velocity within a local window of width w (see Algorithm 1), and obtain a pruned note sequence for an entire recording by simply sliding the window over the entire input sequence, favoring notes based on their velocities. The MIDI input sequence itself is assumed to consist of N events, the relevant features of which are collected in the columns of an $N \times 4$ matrix $\mathbf{E} = [\mathbf{e}_t, \mathbf{e}_v, \mathbf{e}_d, \mathbf{e}_s]$, with $\mathbf{e}_t$ the event onset times vector, $\mathbf{e}_v$ the event velocities vector, $\mathbf{e}_d$ the event durations vector, and $\mathbf{e}_s = \mathbf{e}_v \odot \mathbf{e}_d$ the event saliences vector (with $\odot$ the element-wise product). We assume that the MIDI events in

$\mathbf{E}$ are ordered by increasing event onset times and we denote the nth element of a vector $\mathbf{x}$ by $\mathbf{x}[n]$.

Algorithm 1 MAIBT algorithm (Part A)

procedure input $N \times 4$ MIDI event matrix $\mathbf{E}$
procedure output $\tilde{N} \times 4$ pruned event matrix $\tilde{\mathbf{E}}$
procedure PREPROCESSING($\mathbf{E}, w$)
  initialize index set $\mathcal{I}_p = \emptyset$ of events to be kept; $n \leftarrow 1$
  while $n \leq N$ **do**
    for $\mathcal{I}_w \subset \{m \geq n | \mathbf{e}_t[m] - \mathbf{e}_t[n] \leq w\}$ **do**
      $\mathcal{I}_p \leftarrow \mathcal{I}_p \cup \arg\max_{j \in \mathcal{I}_w} \mathbf{e}_v[j]$
  $\tilde{\mathbf{E}} \leftarrow \mathbf{E}[\mathcal{I}_p, :]; \tilde{N} = |\mathcal{I}_p|$

B. Note Clustering

Following preprocessing, we separate the pruned sequence into short subsequences, with the number of events, S, in each subsequence equal to 20. We then generate clusters of events to determine tempo estimates in each subsection of the recording. Here, we use "tempo" as shorthand for a beat time interval (*i.e.,* time per beat), which is the inverse of tempo as defined in music theory (*i.e.,* beats per time); our units of tempo are thus seconds. We calculate an inter-onset interval (IOI) for pairs of note events and limit the search space to the five subsequent events to improve efficiency. IOIs that have values within some defined cluster width δ are merged into one cluster defined by its average IOI (*i.e.,* centroid). Each cluster is then assigned a score based on the total number of clusters to which it is related; two clusters are related if their IOIs are near-integer multiples of one another (see Algorithm 2). The cluster score depends on a piecewise function, $f(n)$, that takes a value of $6 - n$ if $1 \leq n \leq 4$, a value of 0.2 if $5 \leq n \leq 8$, and 0 otherwise. The average IOI of each cluster represents an aggregate estimate of the tempo for that cluster, and the modified scoring system favors tempo estimates with smaller (rather than larger [15]) values.

C. Agent Initialization and Selection

We initialize each agent with a starting note $e_t[j]$, tempo estimate c_i and cluster score s_i calculated during the clustering step in Section II,B; the agent then predicts the onset time of each subsequent note based on its assigned tempo. We compare and select agents using a three-scenario scoring protocol based on multiple thresholds: If (1) the temporal variation between an agent's prediction and ground-truth timestamp is less than some inner tolerance t_i, the agent retains the note and its score is updated accordingly, (2) the variation is larger than some outer threshold t_h (upper window) or t_l (lower window), the agent rejects the note, does not update its score, and continues on to the following note(s) until its temporal variation exceeds some timeout value, or (3) the variation falls between the inner and outer thresholds, the beat is accepted and score is accumulated, but we create a new identical agent that rejects this note to avoid including a false beat. In case 1, we adjust the agent's predicted instantaneous tempo $a_T(\mathbf{r})$ depending on the error relative to ground truth (Algorithm 3,

Algorithm 2 MAIBT algorithm (Part B)

procedure input $\tilde{N} \times 4$ pruned MIDI events matrix $\tilde{\mathbf{E}}$
procedure output Q IOI clusters C_i, cluster scores s_i, cardinalities k_i and centroids c_i
procedure NOTECLUSTERING($\tilde{\mathbf{E}}, \delta$)
  initialize first cluster: $c_1 \leftarrow \mathbf{e}_t[2] - \mathbf{e}_t[1]; C_1 \leftarrow \{c_1\}$
  initialize number of clusters $Q \leftarrow 1$
  for $n \leftarrow 1, \tilde{N}; m \leftarrow 1, 5$ **do**
    $\Delta e_t(n,m) = \mathbf{e}_t[n+m] - \mathbf{e}_t[n]$
    $i = \arg\min_j |\Delta e_t(n,m) - c_j|$
    if $|\Delta e_t(n,m) - c_i| < \delta$ **then**
      $C_i \leftarrow C_i \cup \{\Delta e_t(n,m)\}; k_i \leftarrow k_i + 1$
      $c_i \leftarrow ((k_i - 1)c_i + \Delta e_t(n,m))/k_i$
    else
      create new cluster $C_q \leftarrow \{\Delta e_t(n,m)\}$
      $k_q \leftarrow 1; c_q \leftarrow \Delta e_t(n,m); Q \leftarrow Q + 1$
  for $i \leftarrow 1, Q$ **do** $s_i \leftarrow 0$; **for** $q \leftarrow 1, Q; n \leftarrow 1, 8$ **do**
    if $|c_i - nc_q| < \delta$ **then** $s_i \leftarrow s_i + f(n)k_q$

Line 11). We compare tempo sequences across all agents and return the agent with highest score.

Algorithm 3 MAIBT algorithm (Part C)

procedure input Q clusters C_i, scores s_i, centroids c_i
procedure output estimated note sequence $\mathbf{a}_t$, tempo a_T
1: **procedure** AGENTSELECT($C_i, s_i, c_i, t_i, t_l, t_h, \alpha, \gamma, S$)
2:   **for** $i \leftarrow 1, Q; j \leftarrow 1, S$ **do**
3:     $k, l \leftarrow 1$; initialize (i,j,k)th agent ($\mathbf{r} = [i,j,k]$):
4:     sequence $\mathbf{a}_t^{(\mathbf{r})} \leftarrow [\mathbf{e}_t[j]]$
5:     tempo $a_T(\mathbf{r}) \leftarrow c_i$
6:     score $a_s(\mathbf{r}) \leftarrow s_i$
7:     prediction $\hat{a}_t(\mathbf{r}, l) \leftarrow \mathbf{a}_t^{(\mathbf{r})}[l] + a_T(\mathbf{r})$
8:     **while** $\hat{a}_t(\mathbf{r}, l) \leq \mathbf{e}_t[\tilde{N}]$ **do**
9:       $p \leftarrow \arg\min_q |\hat{a}_t(\mathbf{r}, l) - \mathbf{e}_t[q]|$
10:       **if** $-t_l \leq \hat{a}_t(\mathbf{r}, l) - \mathbf{e}_t[p] \leq t_h$ **then**
11:         $a_T(\mathbf{r}) \leftarrow a_T(\mathbf{r}) + (\hat{a}_t(\mathbf{r}, l) - \mathbf{e}_t[p])/\gamma$
12:         $\Delta_t(\mathbf{r}, l) \leftarrow |\hat{a}_t(\mathbf{r}, l) - \mathbf{e}_t[p]|/(t_l + t_h)$
13:         $\mathbf{a}_t^{(\mathbf{r})} \leftarrow [\mathbf{a}_t^{(\mathbf{r})}, \hat{a}_t(\mathbf{r}, l)]$
14:         $a_s(\mathbf{r}) \leftarrow a_s(\mathbf{r}) + s_i + \alpha(1 - \Delta_t(\mathbf{r}, l))\mathbf{e}_s[p]$
15:         **if** $|\hat{a}_t(\mathbf{r}, l) - \mathbf{e}_t[p]| \geq t_i$ **then**
16:           create duplicate agent with $k \leftarrow k + 1$, not including $\hat{a}_t(\mathbf{r}, l)$, and update this agent in parallel
17:       update prediction: $\hat{a}_t(\mathbf{r}, l) \leftarrow \hat{a}_t(\mathbf{r}, l) + a_T(\mathbf{r})$
18:       $l \leftarrow l + 1$
19:   $\boldsymbol{\rho} = \arg\max_{\mathbf{r}} a_s(\mathbf{r}); \mathbf{a}_t = \mathbf{a}_t^{(\boldsymbol{\rho})}; a_T = a_T(\boldsymbol{\rho})$

D. Beat insertion and deletion

The agent with the highest score will have retained more important notes in the original sequence than any other agent, but may still lack beat timestamps due to long gaps or mistakes in the improvisation, or may have retained unimportant or misplaced notes. Consequently, we insert and delete beats to

Algorithm 4 MAIBT algorithm (Part D)

procedure input estimated note sequence $\mathbf{a}_t$, tempo a_T
procedure output comprehensive note sequence $\mathbf{c}_t$

1: **procedure** BEATINSERTIONDELETION($\mathbf{a}_t, a_T$)
2:   **while** $a_T < 0.5$ s (fast) or 0.7 s (slow) **do** $a_T \leftarrow 2a_T$
3:   **for** $i \leftarrow 1, \text{length}(\mathbf{a}_t)$ **do** $n \leftarrow \lceil (\mathbf{a}_t[i+1] - \mathbf{a}_t[i])/a_T \rceil$
4:     **if** $n - 0.2 \leq \mathbf{a}_t[i+1] - \mathbf{a}_t[i] \leq n + 0.2$ **then**
5:       **for** $m \leftarrow 1, n-1$ **do**
6:         $\mathbf{c}_t \leftarrow [\mathbf{c}_t, \mathbf{a}_t[i] + (m/n)(\mathbf{a}_t[i+1] - \mathbf{a}_t[i])]$
7:       $\mathbf{c}_t \leftarrow [\mathbf{c}_t, \mathbf{a}_t[i+1]]$
8:     **else**
9:       **for** $m \leftarrow 1, n$ **do** $\mathbf{c}_t \leftarrow [\mathbf{c}_t, \mathbf{a}_t[i] + m a_T]$

create a comprehensive sequence, as shown graphically in Fig. 3, which is critical to calculate all MDs in the patient track in later analyses. In the agent selection step, we favor the agent with the highest score, but we cannot know whether the tempo of this agent stems from quarter note intervals, eighth note intervals, or intervals of other note multiples. Therefore, we define a quarter note interval threshold to convert the agent's tempo into a quarter note interval. If the estimated tempo is smaller than the threshold, it is doubled until it exceeds the threshold. In this work, we set the threshold to be 0.5s (120 BPM) for fast improvisations and 0.7s (85 BPM) for slow improvisations. After tempo conversion, we can use the estimated tempo to construct a tolerance window and subsequently prune the beat sequence. Beats are either inserted at equally spaced intervals between two accepted timestamps (Algorithm 4, Line 6), or at intervals that correspond to the agent's aggregate tempo value, a_T (Algorithm 4, Line 9). This insertion strategy reduces error accumulation due to inaccurate tempo estimation and thus can self correct.

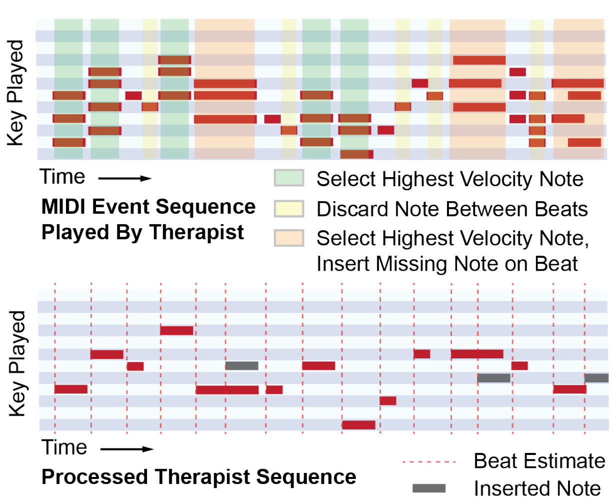

Fig. 3. Overview of the processing protocol, including preprocessing, and insertion and deletion of notes. Highlighted regions of the upper MIDI sequence correspond to different processing steps, and the lower sequence displays notes inserted and beats estimated during the process.

III. Experimental Results and Discussion

We evaluate our approach using two complementary metrics; namely, phase and period accuracy. We developed and tested MAIBT on MIDI data collected as described in [14] and Fig. 1, and compare our results with the original multi-agent beat tracking algorithm [15] as well as a more recent HMM-based approach [21].

A. Evaluation Metrics

To evaluate the performance of MAIBT, we first calculate the phase and period error, which are defined in detail by Pardo *et al.* [17]. Briefly, given a ground truth sequence of timestamps $\mathbf{g}$ and an estimated sequence $\mathbf{b}$ of timestamps, and pairing each ground truth timestamp g_i with its nearest estimate b_j, the phase error is defined as:

$$\phi_i = \frac{|b_j - g_i|}{g_{i+1} - g_i} \tag{1}$$

and the period error is defined as:

$$\tau_i = |log_2 \frac{b_{j+1} - b_j}{g_{i+1} - g_i}| \tag{2}$$

If b_j is the last timestamp in $\mathbf{b}$, then $b_j - b_{j-1}$ is used as the numerator in the period error calculation. Note that the phase error is calculated in reference to the nearest beat, which in turn only reveals the temporal difference between one estimate and one ground truth; a small phase error is not necessarily indicative of accurate timing in the entire piece. To evaluate the overall performance accurately, we must also take the period error into account.

Phase and period errors, from a single note perspective, are relatively independent. An accuracy term for each of these metrics can be obtained by applying exponential normalization across all notes:

$$acc = \frac{\sum_{i=1}^{n} e^{-\frac{error^2}{s^2}}}{(n+m)/2} \tag{3}$$

where n and m are the length of $\mathbf{g}$ and $\mathbf{b}$, respectively, and *error* is either the phase error as defined in Equation 1 or the period error as defined in Equation 2. Accuracy ranges from 0 to 1 and is equal to 1 only when the estimation $\mathbf{b}$ matches exactly with ground truth, $\mathbf{g}$. Here, we introduce a new parameter, s, in the normalization to implement a desired resolution or tolerance of beat time and tempo. Larger values of s minimize the penalty for error, and thus lead to higher overall accuracy results than smaller values of s. In addition to s, we mention other hyperparameters in Section II, and summarize their values and interpretation in Table I.

B. Comparison with State of the Art

We compare the performance of MAIBT with the original multi-agent beat tracking algorithm [15] and an HMM-based method [21], first using period and phase accuracy, and then using other commonly employed metrics like Cemgil [20], P-score [22], and continuity-based evaluation [23], [24] and provide the results in Table II. MAIBT outperforms the other

TABLE I
VALUE AND INTERPRETATION OF HYPERPARAMETERS

Symbol	Value	Unit	Interpretation
w	250	ms	filter window width for preprocessing
δ	80	ms	cluster width, desired resolution of tempo
t_i	50	ms	inner tolerance window width
t_h	100	ms	upper tolerance window width
t_l	80	ms	lower tolerance window width
α	5	-	balance factor for tempo score and salience
γ	10	-	tempo correction factor
s	0.1	-	resolution of phase and period errors

two beat tracking methods for improvisational recordings across all metrics. The original multi-agent method (Dixon MA) occasionally estimates accurate beat timestamps and thus performs reasonably on metrics that emphasize phase deviation (phase accuracy, Cemgil and P-score). However, Dixon MA struggles to account for inconsistent tempos and thus scores poorly on period-based metrics (period accuracy and continuities). The HMM-based method achieves comparable scores to MAIBT when the metrical level is taken into account (AMLc, AMLt) but like Dixon MA, is not designed to handle variable tempos found in improvisational recordings and, accordingly, scores poorly on period-based metrics.

TABLE II
BEAT TRACKING PERFORMANCE

Metrics	MAIBT		HMM [21]		Dixon MA [15]	
	Mean	Std.	Mean	Std.	Mean	Std.
Period acc	**0.7931**	0.1682	0.4562	0.3985	0.2499	0.1908
Phase acc	**0.7506**	0.2489	0.6379	0.1711	0.5849	0.1328
Cemgil	**0.7207**	0.2587	0.5981	0.1817	0.5790	0.1300
P-score	**0.7625**	0.2583	0.6383	0.2055	0.4983	0.1575
CMLc	**0.6191**	0.3393	0.3352	0.3445	0.3284	0.0219
CMLt	**0.7312**	0.2684	0.4529	0.3869	0.1742	0.1535
AMLc	**0.6807**	0.2985	0.5987	0.3076	0.0520	0.0202
AMLt	**0.7884**	0.1918	0.7337	0.2137	0.3487	0.1679
Average	**0.7308**	0.2540	0.5563	0.2762	0.3519	0.1218

IV. CONCLUSION

We developed the MAIBT algorithm using a dataset collected during music therapy sessions; this algorithm performed better than both the original multi-agent method and a state-of-the-art beat tracking method across multiple evaluation metrics for improvisational music. Future optimization of the algorithm will focus on reducing dependencies on predefined hyperparameters, and on applying statistical methods to analyze beat tracking results. Furthermore, we implemented two quarter note interval thresholds specific to the experimental protocol of the music therapy sessions; an advanced, fully automated iteration of the algorithm may infer the correctness of various estimated tempos through intermediate results like number of insertions and deletions, and then adjust the thresholds accordingly. The MAIBT enables rapid and accurate analyses of music therapy sessions; after processing the therapist track, estimated beat timestamps can be easily applied to assess the patient track, as shown in Fig. 1-C, after which MDs can be analyzed using statistical methods to develop diagnostic models and infer disease state.

REFERENCES

[1] P. Grosche, "Signal processing methods for beat tracking, music segmentation, and audio retrieval," Ph.D. dissertation, Saarland University, Germany, 2012.

[2] G. Griffin, Y. E. Kim, and D. Turnbull, "Beat-Sync-Mash-Coder: A web application for real-time creation of beat-synchronous music mashups," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2010, pp. 437–440, iSSN: 1520-6149.

[3] F. Gouyon and S. Dixon, "Dance music classification: A tempo-based approach," in *ISMIR*, 2004.

[4] E. Morgan, H. Gunes, and N. Bryan-Kinns, "Instrumenting the Interaction: Affective and Psychophysiological Features of Live Collaborative Musical Improvisation," *Proceedings of the International Conference on New Interfaces for Musical Expression NIME 14*, pp. 23–28, 2014.

[5] E. A. Atlee, "Inaugural essay on the influence of music in the cure of diseases, an," Ph.D. dissertation, University of Pennsylvania., 1804.

[6] A. Hunt, R. Kirk, M. Abbotson, and R. Abbotson, "Music therapy and electronic technology," in *Proceedings of the 26th Euromicro Conference. EUROMICRO 2000. Informatics: Inventing the Future*, vol. 2, Sep. 2000, pp. 362–367 vol.2, iSSN: 1089-6503.

[7] N. Letule, "An assessment model for the musical material produced during the course of music therapy," in *Casa Baubo. 2º Seminario Internacional de Jazz y Musicoterapia*, 2016.

[8] G. Luck, K. Riikkilä, O. Lartillot, J. Erkkilä, P. Toiviainen, A. Mäkelä, K. Pyhäluoto, H. Raine, L. Verkila, and J. Värri, "Exploring Relationships between Level of Mental Retardation and Features of Music Therapy Improvisations: A Computational Approach," *Nordic Journal of Music Therapy*, vol. 15, no. 1, pp. 30–48, Jan. 2006.

[9] E. Streeter, M. E. P. Davies, J. D. Reiss, A. Hunt, R. Caley, and C. Roberts, "Computer aided music therapy evaluation: Testing the Music Therapy Logbook prototype 1 system," *The Arts in Psychotherapy*, vol. 39, no. 1, pp. 1–10, Feb. 2012.

[10] P. Toiviainen, "Real-Time Recognition of Improvisations with Adaptive Oscillators and a Recursive Bayesian Classifier," *Journal of New Music Research*, vol. 30, no. 2, pp. 137–147, Jun. 2001.

[11] P. Toiviainen and J. S. Snyder, "Tapping to Bach: Resonance-Based Modeling of Pulse," *Music Perception: An Interdisciplinary Journal*, vol. 21, no. 1, pp. 43–80, Sep. 2003.

[12] O. Lartillot, "Intelligent music systems in music therapy," *Music Therapy Today*, vol. 5, Nov. 2004.

[13] T. Eerola and P. Toiviainen, *MIDI Toolbox: MATLAB Tools for Music Research*. Jyväskylä, Finland: University of Jyväskylä, 2004.

[14] K. Foubert, T. Collins, and J. De Backer, "Impaired Maintenance of Interpersonal Synchronization in Musical Improvisations of Patients with Borderline Personality Disorder," *Frontiers in Psychology*, vol. 8, Apr. 2017.

[15] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal Of New Music Research*, vol. 30, no. 1, 2001.

[16] S. Dixon., "A Lightweight Multi-agent Musical Beat Tracking System," in *PRICAI 2000 Topics in Artificial Intelligence*, ser. Lecture Notes in Computer Science, R. Mizoguchi and J. Slaney, Eds. Berlin, Heidelberg: Springer, 2000, pp. 778–788.

[17] B. Pardo, "Tempo Tracking with a Single Oscillator," in *ISMIR*, 2004.

[18] H. Grohganz, M. Clausen, and M. Müller, "Estimating musical time information from performed MIDI files," in *ISMIR*, 2014, pp. 35–40.

[19] A. T. Cemgil and B. Kappen, "Monte Carlo Methods for Tempo Tracking and Rhythm Quantization," *Journal of Artificial Intelligence Research*, vol. 18, pp. 45–81, Jan. 2003, arXiv: 1106.4863.

[20] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, "On tempo tracking: Tempogram Representation and Kalman filtering," *Journal of New Music Research*, vol. 29, no. 4, pp. 259–273, Dec. 2000.

[21] A. McLeod and M. Steedman, "Meter Detection and Alignment of MIDI Performance," in *ISMIR*, 2018, p. 7.

[22] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri, "Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms," *Journal of New Music Research*, vol. 36, no. 1, pp. 1–16, Mar. 2007.

[23] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 342–355, Jan. 2006.

[24] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation Methods for Musical Audio Beat Tracking Algorithms," *Queen Mary University of London, Tech. Rep.*, vol. C4DM-09-06, p. 17, 2009.