UNIVERSIDADE DO PORTO

FACULDADE DE ENGENHARIA

Towards Human-in-the-Loop Computational Rhythm Analysis in Challenging Musical Conditions

António Sá Pinto

Thesis Committee:

António Coelho, Associate Professor with Aggregation, Faculdade de Engenharia da Universidade do Porto (FEUP), Portugal, *President*;

Magdalena Fuentes, Assistant Professor of the Music and Audio Research Lab (MARL) and Integrated Design & Media (IDM), New York University, United States of America, *First Main Examiner*;

Jason Hockmann, Associate Professor of School of Computing and Digital Technology (DMT), Birmingham City University, United Kingdom, *Second Main Examiner*;

Matthew Davies, Senior Scientist at SiriusXM/Pandora, United States of America, *Main Supervisor*;

Rui Nóbrega, Auxiliar Professor, Departamento de Informática da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT-UNL), Portugal, *Voting Member*;

Aníbal Ferreira, Associate Professor, Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto (FEUP), Portugal, *Voting Member*.

Porto, 2023

Towards Human-in-the-Loop Computational Rhythm Analysis in Challenging Musical Conditions

António Humberto e Sá Pinto

In partial fulfillment of requirements for the degree of Doctor of Philosophy in Digital Media

Supervised by

Matthew E. P. Davies, Ph.D., INESC TEC

Cosupervised by

Rui Penha, Ph.D., ESMAE, Politécnico do Porto
Gilberto Bernardes, Ph.D., DEI, Faculdade de Engenharia da Universidade do Porto

Contact Information:

António Sá Pinto
Faculdade de Engenharia da Universidade do Porto

Rua Dr. Roberto Frias, s/n
4200-465 Porto
Portugal

Email: antoniosapinto@gmail.com

This thesis was typeset on an Apple® MacBook® M1 running macOS® 10.6 using the free LaTeX typesetting system, originally developed by Leslie Lamport based on TeX created by Donald Knuth. The body text is set in Palatino, a large typeface family that began as an old style serif typeface designed by Hermann Zapf and that was initially released in 1948 by the Linotype foundry. Other fonts include Sans, Latin Modern and Typewriter from the Computer Modern family, and Inconsolate. The LaTeX style was based on the ones created by Hugo Ferreira and Filipe Correia for their own PhD theses. Most charts were drawn using the `matplotlib` python library.

This work was partially funded by the FCT grant number SFRH/BD/120383/2016, with the support of the European Social Fund and of MCTES national funds.

António Humberto e Sá Pinto
Towards Human-in-the-Loop Computational Rhythm Analysis in Challenging Musical Conditions

Acknowledgements

I owe a great debt of gratitude to Matthew Davies, my primary supervisor, whose unwavering support, technical expertise, and gentle, relatable nature were truly instrumental in the successful completion of my PhD journey. Without his guidance and encouragement, this work could not have been accomplished.

I am also grateful to Rui Penha for his critical insights, availability, and responsiveness whenever I needed assistance. Gilberto Bernardes' valuable technical expertise and overall guidance were instrumental, especially during the latter part of this process.

At FEUP, my heartfelt appreciation is extended to António Coelho. His empathetic support and consistent kindness were vital during key moments of my academic journey. Marisa Silva's invaluable assistance and sympathy were crucial in ensuring that my studies ran smoothly and efficiently. Jaime Cardoso's exceptional Machine Learning course provided the joy of *deep* learning and set me on the right direction.

At INESC TEC, I am grateful to Paula Viana for her support and encouragement, as well as to my colleagues Luis Vilaça and Serkan Surun for their collaboration and camaraderie.

I would also like to acknowledge the support of my close friends and family, particularly André, Carla, Catarina, Luis, and Sérgio, whose stimulating discussions and assistance on various topics, both directly and indirectly related to my thesis, were precious.

My Toni, the smartest cat.

My parents, as always.

This page was intentionally left blank.

Abstract

Music Information Retrieval (MIR) is an interdisciplinary field focused on the extraction, analysis, and processing of information from various musical representations. Grounded on the automatic analysis of musical facets such as rhythm, melody, harmony, and timbre, MIR enables applications in areas like music recommendation, automated music transcription, and intelligent music composition tools. Rhythm, an integral element of music, provides a foundation for decoding music's complex relational structures and layered depth. Computational rhythm analysis is thus central to MIR research. It encompasses a wide range of tasks, such as the pivotal beat tracking, which unlocks the use of musical *time* across many MIR systems. However, conventional beat-tracking methods have struggled when dealing with complex musical features, such as expressive timing or intricate rhythmic patterns. While specialised approaches demonstrate some degree of adaptation, they do not generalise to diverse scenarios. Deep learning methods, while promising in addressing these issues, depend heavily on the availability of substantial annotated data. In scenarios requiring adaptation to user subjectivity, or where acquiring annotated data is challenging, the efficacy of beat-tracking methods lowers, thus leaving a gap in the applicability of computational rhythm analysis methods.

This thesis investigates how user-provided information can enhance computational rhythm analysis in challenging musical conditions. It initiates the exploration of human-in-the-loop strategies with the aim of fostering adaptability of current MIR techniques. By focusing on beat tracking, due to its fundamental role in rhythm analysis, our goal is to develop streamlined solutions for cases where even the most advanced methods fall short. This is achieved by utilising both high-level and low-level user inputs — namely, the user's judgement regarding the expressiveness of the musical piece and annotations of a brief excerpt — to adapt the state of the art to particularly demanding signals.

In an exploratory study, we validate the shared perception of rhythmic complexity among users as a proxy for musical expressiveness, and consequently as a key performance enhancer for beat tracking. Building upon this, we examine how high-

level user information can reparameterise a leading-edge beat-tracker, augmenting its performance to highly expressive music. We then propose a transfer learning method that finetunes the current state of the art, hereafter referred to as the baseline, to a concise user-annotated region. This method exhibits versatility across varied musical styles and offers potential solutions to the inherent limitations of previous approaches. Incorporating both user-guided contextualisation and transfer learning into a human-in-the-loop workflow, we undertake a comprehensive evaluation of our adaptive techniques. This includes examining the key customisation options available to users and their effect on performance enhancement.

Our approach outperforms the current state of the art, particularly in the challenging musical content of the *SMC* dataset, with an improvement over the baseline F-measure of almost 10 percentage points (corresponding to over 16%). However, these quantitative improvements require further interpretation due to the inherent differences between our file-specific, human-in-the-loop technique and traditional dataset-wide methods, which operate without prior exposure to specific file characteristics.

With the aim of advancing towards a user-centric evaluation framework for beat tracking, we introduce two novel metrics: the *E-Measure* and *Annotation Efficiency* (*Ae*). These metrics account for the user perspective regarding the annotation and finetuning process. The E-Measure is a variant of the F-measure focused on the annotation correction workflow and includes a shifting operation over a larger tolerance window. The Ae is defined as the relative (to the baseline) decrease in correction operations enabled by the finetuning process, normalised by the number of user annotations. Specifically, we probe the theoretical upper bound of beat tracking accuracy improvement over the *SMC* dataset. Our results show that the correct beat estimates provided by our approach surpass those of the state of the art by more than 20%. When considering the full length of the files, we can further frame this improvement in terms of gain per unit of user effort, quantifying the annotation efficiency of our approach. This is reflected in the substantial reduction of required corrections, with nearly two-thirds fewer corrections per user annotation compared to the baseline.

In the final phase, we evaluate the adaptability of our human-in-the-loop strategy across a range of musical genres and challenging instances. Our exploration encompasses various rhythm tasks, such as beat tracking, onset detection, and (indirectly) metre analysis. We apply this user-driven strategy to three distinct genres characterised by complex rhythm structures, including polyrhythms, polymetres, and polytempi. Our approach demonstrates swift adaptability, facilitating efficient use of

the state-of-the-art method and obviating the need for extensive retraining. This leads to a balanced integration of data-driven and user-centric methods, culminating in a practical and streamlined solution for computational rhythm analysis.

Resumo

A Recuperação de Informação Musical (MIR) é uma área científica interdisciplinar focada na extração, análise e processamento de informação a partir de variadas representações musicais. Através da análise automática de propriedades musicais como o ritmo, melodia, harmonia e timbre, esta área de investigação potencia o desenvolvimento de aplicações de recomendação musical, transcrição automatizada, assim como de ferramentas inteligentes de composição musical. O ritmo — um componente vital da música — é fundamental para a descodificação das estruturas relacionais que compõem a hierarquia musical. Por esta razão, a análise computacional do ritmo ocupa um espaço central na investigação em MIR, englobando um vasto leque de tarefas, entre as quais se inclui a extração da pulsação (*beat tracking*), fundamental para o uso computacional do tempo musical em muitos sistemas MIR. No entanto, quando expostos a características musicais complexas, tais como uma acentuada expressividade musical ou padrões rítmicos especialmente elaborados, os métodos existentes de extração da pulsação apresentam limitações. Embora existam abordagens especializadas que demonstram algum grau de adaptação, estas não são generalizáveis para cenários diversificados. Os métodos de aprendizagem profunda (*deep learning*), embora promissores, dependem da disponibilidade de grandes volumes de dados anotados. Em cenários que exigem adaptação à subjetividade do utilizador, ou onde a aquisição de dados anotados é difícil, a eficácia dos métodos de extração da pulsação musical fica comprometida, manifestando-se assim uma lacuna importante na aplicabilidade dos métodos de análise rítmica computacional.

Nesta tese, exploramos a utilização de informação fornecida pelo utilizador, de forma a potenciar a análise rítmica computacional em contextos musicais desafiantes. Mais especificamente, adotamos estratégias centradas no utilizador (*human-in-the-loop*), com o objectivo de promover a adaptabilidade das técnicas MIR atuais. Com um enfoque na extração da pulsação musical, o nosso objetivo é desenvolver soluções eficientes para os casos onde até os métodos mais avançados falham. Para tal, aproveitamos a informação contextual proveniente do utilizador, particularmente do seu julgamento acerca da expressividade da peça musical em análise, assim como de anotações da

pulsação de um breve trecho, permitindo desta forma adaptar o estado-da-arte a sinais particularmente exigentes.

Num estudo exploratório, validamos o alinhamento entre a percepção da complexidade rítmica de um grupo de utilizadores como indicador de presença de expressividade no sinal musical, e, consequentemente, como forma de aprimorar a extração da pulsação. Com base nesta premissa, estudamos a forma de utilização desta informação de alto nível como suporte para a reparameterização de um método de extração da pulsação de última geração, aumentando o seu desempenho para trechos musicais altamente expressivos. Propomos, em seguida, um método de aprendizagem por transferência (*transfer learning*) que refina o atual estado de arte através de *finetuning* de uma pequena região anotada pelo utilizador, o qual constituirá a base de referência (*baseline*) ao longo da nossa investigação. O nosso método mostra grande versatilidade em diversos estilos musicais e oferece soluções potenciais para as limitações existentes nos métodos actuais. Ao integrar a contextualização pelo utilizador e a aprendizagem por transferência num cenário em que este é parte ativa (*human-in-the-loop*), procedemos a uma avaliação exaustiva de ambas as abordagens, através da qual analisamos as principais opções de customização e adaptação disponíveis para os utilizadores, assim como o seu impacto específico no desempenho final.

De uma forma geral, a nossa estratégia permite superar o estado-da-arte de forma consistente. No entanto, no caso de conteúdo musical desafiante (amplamente presente no conjunto de dados *SMC*), este aumento é especialmente pronunciado, como revela o incremento de quase 10 pontos percentuais (ou cerca de 16%) sobre a *baseline* em termos da F-measure. No entanto, estas melhorias quantitativas requerem uma interpretação cuidada devido às diferenças inerentes entre a nossa abordagem (*human-in-the-loop*) — específica para cada ficheiro áudio —, e os métodos convencionais (como é o caso do algoritmo estado-da-arte), gerais para todo o conjunto de dados, e que operam sem exposição prévia às características específicas de cada arquivo.

Com o objetivo maior de contribuir para a criação de uma estrutura de avaliação de extração da pulsação musical centrada no utilizador, introduzimos duas novas métricas: *E-Measure* e *Annotation Efficiency* (*Ae*). Estas métricas centram-se na perspectiva do utilizador sobre o processo de anotação e *finetuning*. A E-Measure é uma variante da F-measure focada no fluxo de trabalho de correção de anotação que inclui uma operação de deslocamento (*shift*) sobre uma janela de tolerância mais abrangente. A *Ae* é definida como a diminuição do número de correções de anotações possibilitadas pelo processo de *finetuning* (em relação à *baseline*), normalizada pelo número de anotações introduzidas pelo utilizador. Investigamos o limite teórico de melhoria na precisão da

extração da pulsação no conjunto de dados *SMC*. Os nossos resultados mostram que o número de estimativas corretas do nosso método supera as do estado-da-arte em mais de 20%. Considerando a duração total do áudio deste conjunto de dados, podemos enquadrar esta melhoria em termos de ganho por unidade de esforço do utilizador, ou seja, quantificando a eficiência da anotação do utilizador na nossa abordagem. Os resultados revelam uma redução significativa do número de correções necessárias em comparação com o algoritmo estado-da-arte, com quase 2/3 menos correções por anotação do utilizador.

Para concluir, avaliamos a adaptabilidade da nossa estratégia *human-in-the-loop* numa variedade de trechos e géneros musicais altamente complexos. Neste âmbito, além da extração da pulsação musical, alargamos a nossa abordagem a outras tarefas de análise rítmica, incluindo detecção de ataques (*onsets*), e (indiretamente) análise de métrica musical. Aplicamos o nosso método a três géneros singulares que apresentam estruturas rítmicas complexas, como polirritmos, polimétricas, e politempos. O nosso método exibe adaptibilidade de forma eficiente, permitindo a utilização do método estado-da-arte sem necessidade de repetir processos morosos de treino. Em suma, através da integração de métodos de aprendizagem computacional com técnicas centradas no utilizador, potenciamos uma solução prática e ágil para a análise computacional de ritmo.

Contents

List of Figures

List of Tables

List of Acronyms

ACF Autocorrelation Function

AI Artificial Intelligence

AL Active Learning

BLSTM Bidirectional Long Short-Term Memory

BPM Beats per Minute

CE Computational Ethnomusicology

CNN Convolutional Neural Network

DBN Dynamic Bayesian Network

HCI Human-Computer Interaction

HITL Human-in-the-Loop

HMM Hidden Markov Model

IAI Inter-Annotation-Interval

IBI Inter-Beat-Interval

IML Interactive Machine Learning

IOI Inter-Onset-Interval

LSTM Long Short-Term Memory

MIR Music Information Retrieval

MIREX Music Information Retrieval Evaluation eXchange

ML Machine Learning

PGM Probabilistic Graphical Model

RL Reinforcement Learning

RNN Recurrent Neural Network

SMC Sound and Music Computing

STFT Short-Time Fourier Transform

TCN Temporal Convolutional Network

TL Transfer Learning

UX User eXperience

XAI Explainable Artificial Intelligence

1

Introduction

This chapter presents a general introduction to the dissertation. Starting with the motivation to this work and the research objectives, we then present the main contributions of this thesis. Finally, the chapter concludes with an outline of the document's structure.

1.1 Research Context

Music Information Retrieval (MIR) is an interdisciplinary research field that encompasses a diverse range of disciplines, such as signal processing, machine learning, computer science, cognitive psychology, and music theory [Müller, 2015]. Since its inception, MIR's primary objective has been to develop computational methods for analysing and representing music, and designing systems capable of automatically extracting, processing, and organising music-related data [Downie, 2003]. These methods and

systems have been deployed in numerous areas, from music search and automatic transcription to music visualisation and interactive music solutions.

Reflecting the variety of its applications, the MIR field has transformed over time, moving through distinct "ages" [Herrera-Boyer, 2018]. In the beginning, users mainly provided input and interpreted output for specific tasks, playing mostly passive roles [Schedl and Knees, 2013]. Subsequently, efforts were made to bridge the "semantic gap" between the low-level audio data and the high-level musical cognition. This shift was marked by the development of more complex algorithms and the adoption of machine learning techniques. Further evolution led to a focus on user- and context-specific factors, making MIR systems more effective and personalised. Users transitioned into more active roles, their feedback and annotations becoming essential for system performance improvement [Schedl and Flexer, 2012]. Currently, MIR research extends far beyond simple retrieval and analysis, fostering systems that amplify musical creativity. Interaction with MIR systems has become dynamic and collaborative, positioning users as integral co-creators [Herrera-Boyer and Gouyon, 2013]. Mirroring these shifts, MIR's evolution underscores its progressive turn towards user-centric systems.

While the evolution of MIR emphasises greater user involvement, the field also grows in other significant ways. It asserts its role as a research-driven field [Serra et al., 2013], steering advancements through evaluation and aligning increasingly with other scientific domains. This evolution has prompted community proposals for a potential renaming of the field. Terms such as *Music Information Research*, suggested by Herrera et al. [2009], and *Music Informatics Research*, adopted by Humphrey and Bello [2012], highlight the field's research-centric focus and its growing synergy with informatics-based methodologies. Amidst these discussions[1], one element remains uncontested - *music*, as MIR's consistent focus and object of study.

At the core of MIR lies the extraction of music information, focused on elements like timbre, melody, and harmony. Content-based analysis stands as a main line of research, aiming to unravel the various facets that shape our interaction with music. Among these, rhythm, as a vital component of music, plays a crucial role in the overall musical experience [Sachs, 1953]. Consequently, computational rhythm analysis has emerged as a fundamental research focus within MIR. This subfield is dedicated to decoding various musical facets, such as tracking the beat, recognising rhythmic patterns, and determining the metre [Dixon et al., 2003; Gouyon and Dixon, 2005; Klapuri et al.,

[1] While we acknowledge the value these discussions bring to the field's evolution, we have chosen to continue using the original term, *Music Information Retrieval*, throughout this dissertation.

2006]. Among these, algorithmic beat tracking serves as a foundational task, which involves the automatic determination of a musical signal's pulse, a computational analogue of tapping one's foot in time to music [Davies and Plumbley, 2007].

However, the relevance of beat tracking extends beyond merely emulating a facet of human perception, as it has become indispensable in numerous MIR applications that rely on parsing "musical time". Serving as an intermediate processing step within larger scale systems, beat tracking enables the beat-synchronous analysis of elements such as harmony or structure, and streamlines dynamic responses and real-time processing in live music and other interactive applications, fostering a shared *time* between musicians and computers [Stark and Plumbley, 2011; Davies et al., 2014; Vande Veire and De Bie, 2018]. Given its paradigmatic role, we embrace beat tracking as a demonstrator for addressing the current challenges and exploring potential advancements in the field.

From a technical perspective, computational approaches to musical audio beat tracking, akin to numerous MIR tasks, have undergone a significant transformation. Traditional beat tracking methods typically relied on pattern recognition and signal processing techniques to analyse audio features and identify beat locations [Goto and Muraoka, 1994]. However, these methods face difficulties when dealing with highly expressive music lacking clear percussive elements, as they are better suited to music with relatively steady tempos and distinct percussive content [Grosche et al., 2010; Holzapfel et al., 2012b]. The emergence of machine learning (ML) approaches facilitated the shift from feature design to feature learning, addressing the suboptimal and unsustainable nature of hand-crafted features [Humphrey and Bello, 2012]. Subsequently, the fundamental limitations of shallow architectures prompted a progressive transition towards increasingly deeper ones [Humphrey et al., 2013a], culminating in the dominance of deep learning in MIR [Peeters and Richard, 2021].

While deep learning has facilitated advancements in beat tracking methods, even the state of the art [Böck and Davies, 2020] falters when facing challenging signals, such as those with ambiguous pulses and highly-expressive timing. Consequently, there remains ample room for improvement and exploration within this domain, as researchers continue to seek more robust and adaptable solutions to address the inherent complexities of musical rhythm analysis. In this new paradigm, the availability of large and representative annotated datasets constitutes a critical element in MIR research [Peeters, 2021]. However, the need for expert annotation in challenging music scenarios, coupled with the conventional restriction on free music distribution among researchers, significantly curtails the availability of datasets. Consequently,

existing datasets are often too small and homogeneous [Salamon, 2019][2], or even fail to accurately represent real-world musical scenarios, as evidenced by the prevalent use of brief annotated excerpts rather than complete musical compositions [Hainsworth, 2004; Tzanetakis and Cook, 2002]. These limitations contribute to the challenge of developing effective, data-driven MIR methods that can generalise well across various musical genres and styles.

This dissertation is situated within the broader context of content-based MIR, at the intersection of audio signal processing and machine learning. Our research aims to contribute to surmounting current limitations, particularly by leveraging user knowledge to streamline the adaptation and applicability of prevailing data-driven approaches. In the following section, we will explore the specifics of this research and discuss how it addresses these challenges.

1.2 Motivation and Scope

Music, a universal form of human expression, holds a central position in our cultures and societies, with rhythm occupying a foundational role [Patel, 2006]. The understanding of rhythm, therefore, is fundamental to the comprehension of music. However, the complexity and diversity of rhythm present unique challenges for computational analysis. Beat tracking, as the backbone of temporal music analysis, embodies these problems.

Music inherently encompasses issues such as expressive timing and complex rhythmic patterns, which have persistently presented difficulties for beat-tracking methods [Grosche et al., 2010]. The necessity for adaptability in rhythm analysis is evident, yet the solutions are not straightforward. Traditional methods have often been found wanting in the flexibility required to adapt to rhythmic intricacies [Holzapfel et al., 2012b]. Conversely, the adaptability of deep learning methods is contingent on the availability of large and diverse datasets [Davies and Böck, 2019].

These issues become more impactful in specialised contexts that diverge from mainstream music, as in Computational Ethnomusicology [Tzanetakis et al., 2007]. The greater scarcity of annotated datasets in this area can be attributed to increased difficulties in automatic analysis and the specific cultural awareness necessary to obtain accurate annotations. These problems compound upon pre-existing issues in rhythm analysis: Western music theory has shown limitations, as some terminology and

[2] Incomparable to the size of those in leading deep learning disciplines such as Computer Vision (CV) and Natural Language Processing (NLP).

concepts do not apply universally to all music traditions [Kolinski, 1973]. Additionally, MIR systems have historically been focused on "Eurogenetic" music [Serra, 2011; Gómez et al., 2013], resulting in a bias that hinders accurate analysis and processing of diverse music data, ultimately leading to suboptimal performance. In particular, the fundamental role of rhythm across many music cultures, especially in Afro-rooted music [Kubik, 2010; Bello et al., 2015], underscores the necessity of addressing these issues. Adapting and evaluating existing models for underrepresented music traditions in MIR often calls for informed analysis [Srinivasamurthy et al., 2017], a process that frequently involves the labor-intensive task of collecting and annotating large amounts of data [Holzapfel, 2014; Nunes et al., 2015]. Such impracticality results in significant gaps in computational rhythm analysis methods for certain music traditions.

Another clear instance of the current methods' limitations is the domain of Creative MIR [Andersen and Knees, 2016]. Within this context, beat tracking plays a crucial role in providing a shared musical *time*, enabling musically-responsive and interactive systems [Davies et al., 2013]. Exceptional precision is of utmost importance, as the quality of subsequent analyses or creative musical outcomes heavily depends on the accuracy of beat estimation. The high expectations of users [Humphrey et al., 2013b] and their specific preferences further complicate matters due to the subjective nature of music in general, and rhythm in particular. In these creative settings, addressing the user's goal or need is essential: the question "*Can the beats be accurately extracted – as envisioned by the user – for this specific piece of music?*", takes clear precedence over achieving high mean accuracy scores across existing databases.

Lastly, accuracy scores may not provide a comprehensive understanding of a system's true performance [Sturm, 2013; Davies et al., 2014], especially when relying on human-annotated ground truth [Flexer, 2014]. However, the high costs associated with expert annotation, particularly when requiring cultural-awareness as is the case of many musical traditions where current methods underperform, create a persistent bottleneck. As a result, the sustainability of MIR evaluation and development cycles that drive the advancement of the research field is challenged [Downie et al., 2010; Urbano et al., 2013]. Within this context, evaluation plays a crucial role, since it is the main method of assessing the effectiveness of MIR systems and techniques.

In response to these challenges, we pose the following research questions:

Is it possible to enhance the adaptability of these methods across different musical contexts? How can we adjust these techniques to better align with user needs and preferences? Can user-provided information unlock current challenges in computational rhythm analysis? What evaluation strategies could better assess user-centric workflows?

In this dissertation, we explore a potential solution: leveraging the knowledge of the user. We propose a user-driven approach to beat tracking, aiming to enhance the accuracy of MIR methods while broadening the array of signals they can target. This approach seeks to enhance the adaptability of MIR methods without requiring extensive model training on annotated datasets. Our thesis is that user knowledge can be leveraged for *in-situ* adaptability of leading MIR methods, thus improving robustness to challenging musical conditions.

In light of the above context and research queries, the objectives of this dissertation are the following:

- Examine the challenges of computational rhythm analysis, particularly in beat tracking, by pinpointing the musical features that create obstacles, recognising the limitations of current techniques in terms of performance and evaluation, reviewing machine learning methods that could address these issues, and examining the role of users in this landscape.
- Determine a low-dimensional parameterisation of beat tracking space and experimentally validate this user-informed mapping to ensure it provides perceptually-valid beat annotations.
- Establish a user-directed adaptation strategy for reparameterisation of an advanced beat-tracking model, aiming to improve its performance in assessing highly expressive music.
- Design an *in-situ* transfer learning approach for handling difficult beat tracking cases. This method involves exposing the advanced model to a limited set of user-annotated data.
- Establish user-centric metrics for beat tracking evaluation, focused on correctly modelling the annotation workflow and provide a measure of efficiency of our method.
- Conduct a comprehensive evaluation, comparing our proposed user-driven approach with state-of-the-art methods using established datasets. Analyse particularly challenging cases in detail to assess the advantages and potential limitations of our approach.
- Assess the applicability of our approach within the contexts of Computational Ethnomusicology and Creative MIR. Explore its performance across a variety of genres and rhythm analysis tasks with varying complexity.

This work aligns with the principles of open and reproducible research. In honour of these principles, all code generated during this research, along with any sharable

data or models, will be made openly available to the research community under open licensing terms.

In line with these objectives, we set out to address current limitations in computational rhythm analysis, with an emphasis on beat tracking.

1.3 Main Contributions

The main contributions of this dissertation are the following:

- A method for customising an advanced data-driven beat tracker, exploiting user perspective about musical timing expressiveness to enhance accuracy without requiring model retraining. The method reparameterises the algorithm's post-processing Dynamic Bayesian Network, offering improved analysis of highly expressive music. Details are given in Chapter,3.

- An *in-situ* adaptive strategy to handle subjectivity and complexity in musical audio beat tracking. Through transfer learning targeted on a brief segment of user-annotated music, we adapt the current state-of-the-art beat-tracking model [Böck and Davies, 2020] to diverse musical expressions and improve its overall performance. This contribution is presented in Chapter 4.

- Building upon the previous strategies, we propose a comprehensive approach that combines both user-driven reparameterisation and *in-situ* finetuning for a streamlined, user-centric adaptation of Böck and Davies [2020] TCN-based state of the art. By simultaneously finetuning the neural network and dynamically customising the Dynamic Bayesian Network in accordance with user annotations and preferences, this combined strategy further enhances the model's adaptability to various musical contexts. This holistic approach is elaborated further in Chapter 5.

- Two novel user-centric metrics for beat-tracking evaluation, each designed with a focus on the user's perspective within an annotation workflow. The first, the *E-Measure*, adapts the well-known F-measure to explicitly model the *edit* operations that are common in beat annotation workflows (e.g., correct detections, deletions and insertions), to which we add the frequent shift operation. This allows for a more precise evaluation of the algorithm's performance in alignment with user editing activities. Our second proposed metric is the *Annotation Efficiency* (Ae), which assesses the improved performance in relation to user effort when

compared to a baseline approach. Specifically, it quantifies the reduction in correction operations achieved through the finetuning process, normalised by the number of user annotations. This metric provides a measure of the efficiency of the annotation process, offering insight into the practical benefits of our proposed methods. These contributions are presented in Chapter 4 and subsequently utilised in most evaluation exercises.

The work in this dissertation led to the development of the following open source software library:

- `beatflow`: a library which models the beat-tracking annotation workflow, providing complementary user-centric metrics and a visualisation module that graphically depicts evaluation results in terms of the correction edit operations. This software library is made available under open licensing at `https://github.com/asapsmc/beatflow`. It is presented in detail in Chapter 4 and underpins all evaluation and visualisation across this dissertation.

1.4 Publications and Research Affiliations

Some of the research presented in this thesis has been previously published in the following works:

P.1 António Sá Pinto and Matthew E. P. Davies. **Towards user-informed beat tracking of musical audio**. In *14th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pages 577-–588, 2019.

P.2 António Sá Pinto, Inês Domingues, and Matthew E. P. Davies. **Shift If You Can: Counting and Visualising Correction Operations for Beat Tracking Evaluation**. In *Extended Abstracts for the Late-Breaking Demo Session of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

P.3 António Sá Pinto and Matthew E. P. Davies. **Tapping Along to the Difficult Ones: Leveraging User-Input for Beat Tracking in Highly Expressive Musical Content**. In Richard Kronland-Martinet, Sølvi Ystad, and Mitsuko Aramaki, editors, *Perception, Representations, Image, Sound, Music - 14th International Symposium, CMMR 2019, Marseille, France, October 14-18, 2019, Revised Selected Papers, volume 12631 of Lecture Notes in Computer Science*, pages 75-–90. Springer, 2021.

P.4 António Sá Pinto, Sebastian Böck, Jaime S. Cardoso, and Matthew E. P. Davies. **User-Driven Fine-Tuning for Beat Tracking**. *Electronics*, 10(13):1518, jun 2021.

The work in section 6.1.1 contributed to the project "The Healing and Emotional Power of Music and Dance" (HELP-MD), PTDC/ART-PER/29641/2017, supported by Portuguese National Funds through the FCT—Foundation for Science and Technology, I.P.

1.5 Dissertation Outline

The remainder of this thesis is structured as follows.

We begin by laying the foundation with a comprehensive background review and domain characterisation. **Chapter 2 – Background and Related Work** provides an overview of the fundamental knowledge necessary for understanding the content of this dissertation. Rhythmic concepts, including those related to metre, beat, tempo, and timing, are presented in Section 2.1. Both musical notation and a signal-processing frame of reference are used to characterise these concepts. In Section 2.2, the task of beat tracking is presented. The classical approaches are briefly discussed, while contemporary advancements in the field are given a more in-depth examination. In addition to introducing the topic of beat tracking, the main challenges and obstacles associated with it are also highlighted. This includes considerations such as the varying contexts in which beat tracking may be applied, the specific characteristics of the audio or musical content that can impact the accuracy and effectiveness of the analysis, and potential avenues for addressing these challenges and improving the performance of beat tracking. Section 2.3 concludes by discussing the evolving role of the user in MIR and Machine Learning (ML), and how our work is positioned within this broader research landscape.

Contributions are then presented, as we explore the potential of leveraging user input in the context of beat tracking.

In **Chapter 3 – High-Level User Parameterisation in Beat Tracking**, we investigate the potential of utilising high-level contextual information to enhance beat-tracking analysis, by reparameterising an existing Recurrent Neural Network (RNN) model for highly expressive music. We demonstrate that the ability for a user to choose between default and expressive parameterisation can lead to significant improvements on beat tracking for challenging musical audio, without requiring expensive retraining of the state-of-the-art method. To provide insight into how user high-level input

can be applied to beat tracking, we develop a small listening study in which user decisions regarding the perceived difficulty of tapping are translated into the use of a parameterisation for expressive musical excerpts.

In **Chapter 4 – User-Informed Finetuning for Improved Beat Tracking**, we present our finetuning approach to beat tracking of challenging musical signals. Our technique is based on the adaptation of the present state-of-the-art beat tracker through exposure to a brief user-annotated region. We show that this approach outperforms the state of the art performance and has the capacity to adapt to demanding characteristics in terms of timbre and musical expression of the target signal. Our evaluation is designed to ensure a fair comparison with existing methods, given that our technique produces a model for each music file, and not per dataset, as in conventional beat-tracking approaches. To (partially) mitigate this bias, we exclude the segment of the file used for finetuning from the evaluation, upholding the principle of testing models strictly on unseen data. Furthermore, to illustrate the potential of transfer learning within a semi-automatic annotation workflow, we introduce a user-centric evaluation approach for beat tracking, advancing two novel metrics: the Edit–F-measure (E-measure) and annotation efficiency (Ae). We also present a visualisation tool for the qualitative evaluation of beat tracking systems, that models beat tracking outputs in terms of correction operations (i.e., correct detections, insertions, deletions and shifts).

A comprehensive evaluation of our approach is conducted in two parts:

Chapter 5 – A Comprehensive Examination: Leveraging User-Centric Approaches in Beat Tracking builds upon the techniques from Chapters 3 and 4, offering a comprehensive analysis. In this chapter, we explore the influence of user choice on the performance of our method, systematically testing all 11 possible configurations that a user might select. These configurations combine finetuning techniques, with and without data augmentation, and two post-processing customisations: an expressiveness-adaptation and the use of a tempo range. Alongside this, we expand our previous evaluation methodology by generating results that both exclude and include the fine-tuning segment. To reflect this broader scope, we introduce the Ae metric into our analysis, providing a measure of the overall efficiency of our approach. We commence by examining the performance of our approach across the reference datasets and conclude with an in-depth inspection of specific beat-tracking cases.

Chapter 6 – Adaptive Rhythm Analysis in Challenging Musical Contexts adopts a different perspective, focusing on the potential domains of application for our human-in-the-loop strategy. We first address the field of computational ethnomusicology, wherein we apply our approach to diverse rhythm analysis tasks across challenging

non-Western datasets: onset detection for *Maracatu*, beat tracking for *Bambuco* (and indirectly metre determination) and *Candombe*. Second, we probe into the domain of Creative MIR, where the automated extraction of the beat plays a critical role (e.g., in interactive systems). Within a creative setting, the end-user has high expectations for precise and perceptually accurate analysis regardless of the *difficulty* of the specific track, as their final musical goals depend on it. To simulate this context, we evaluate our approach on Steve Reich's *Piano Phase*, a particularly inventive and demanding piece that effectively pushes the limits of beat tracking. In both application domains, our approach demonstrates adaptability and robustness in tackling complex rhythmic structures, including polyrhythms, polymetres, and polytempi.

The conclusion of this dissertation, found in **Chapter 7 – Conclusion**, discusses the findings of this thesis in a broader context and provides an overview of future research directions.

Additionally, the following appendices are included:

Appendix A – Complementary Results, providing extended results for the experiments presented in Chapters 5 and 6.

Appendix B – Supplementary Experiments, presenting additional details regarding specific elements of our user-driven beat tracking analysis approach not covered in the main body of the thesis. Specifically, we explore the criteria for finetuning region selection, optimization strategies for the finetuning process, and the computational demands of our method in terms of training time.

Appendix C – Music Reference, listing the musical works referenced in our computational analysis, accompanied by streaming links and a dedicated repository for specific tracks not available on mainstream platforms. Listening to these works provides a direct auditory reference for the discussion presented.

2

Background and Related Work

The study of musical rhythm, central to our understanding of the structure and feel of a piece of music, plays a crucial role in the field of Music Information Retrieval (MIR). In this chapter, we begin by presenting the fundamental concepts and terminology associated with musical rhythm, which will form the basis for our understanding of the rest of the thesis.

Building on this foundation, we then explore the current state of the art in beat tracking, the task of automatically detecting the underlying pulse or beat in a piece of music. This section provides an overview of the most important algorithms and techniques for beat tracking, discussing their strengths and limitations, along with the current challenges in evaluation.

We then move to examine the evolution of the role of the user in MIR and Machine Learning (ML), highlighting the shift towards human-centred approaches, that shapes the development, evaluation, and performance of data-driven systems. Furthermore, we focus on the user's role as an annotator, using beat annotation as a case study. An overview of beat annotation is provided, alongside a discussion on the inherent

uncertainty in the process, and methods for assessing annotation efficiency. In these considerations, we underscore the critical importance of user involvement.

Lastly, we position our work within the larger MIR research landscape, outlining the key approaches and methodologies we intend to follow in order to develop effective, user-centred solutions for computational rhythm analysis.

2.1 Musical Rhythm

"To study rhythm is to study all of music", Cooper and Meyer [p.1 1960] asserted in their influential work, emphasizing the importance of rhythm in music. Rhythm can be described as the methodical organisation of *time* in music, encompassing concepts such as movement, regularity, and emphasis [Handel, 1989]. It is a vital element that determines a composition's flow and overall structure [Epstein, 1995], standing out as *the* single most pervasive aspect of music.

Despite its recognition as one of the two (alongside pitch) primary dimensions of music [Meyer, 1973], rhythm has historically received less attention compared to its counterpart. As a result, the evolution of rhythmic theory has been relatively impoverished in comparison to other facets of music [Cooper and Meyer, 1960; Fraisse, 1982]. However, the latter part of the 20th century saw significant advancements in the study of rhythm in music (see Clarke [1999] for a comprehensive review), which can be attributed to various factors.

Technological innovations such as MIDI[3] have allowed for the precise measurement of keyboard attack timings. Furthermore, the broader use of experimental methods in music research has played a significant role in promoting empirical studies of rhythm (e.g. the early works of Repp [1992, 1994], which involved the precise measurement of inter-onset-intervals (IOIs) in musical audio). These advancements have led to a shift in the field towards a more comprehensive study of rhythm and timing within the framework of cognitive science [Honing, 2013].

The development of interpretative models of music temporal patterns has also played a crucial part in advancing the field. One instance is Yeston [1976]'s *The Stratification of Musical Rhythm*, which developed a theoretical framework for analysing musical rhythmic structures by breaking them down into hierarchical layers, the "rhythmic strata". Adding a more profound impact, Lerdahl and Jackendoff presented their groundbreaking work, *A Generative Theory of Tonal Music*[1983]. This study

[3] Musical instruments Digital Interface (MIDI) is a 1983 technical standard that allows computers, musical instruments, and other hardware to communicate.

identified two primary components of rhythmic structure in Western tonal and metric music: *grouping* and *metre*. Grouping refers to the segmentation of music at various levels, from small groups of notes to the overall form of the work, whereas metre involves the regular alternation of strong and weak (accented or unaccented) elements in the music.

Interestingly, these concepts integrate with signal processing concepts, as they can be seen as the extreme ends of a Fourier transform. Time-domain and frequency-domain approaches share a direct analogy with grouping and metre, respectively [Todd, 1994]. Grouping, similar to a time-domain approach, focuses on localised structural units, while metre, akin to the frequency-domain approach, concentrates on the underlying patterns of strong and weak beats. In a similar vein, signal processing researchers have also explored the use of wavelets[4] to analyse musical audio, aiming to make explicit the various layers implicit in a rhythmic signal [Smith and Honing, 2008]. This transform represents the effects of dynamic and temporal accents in establishing hierarchies of rhythmic frequencies, providing a tangible link back to both Yeston's [1976] concept of rhythm strata and the metrical and grouping structure theory of Lerdahl and Jackendoff [1983].

Drawing upon the preceding discussion, it becomes evident that a holistic understanding of rhythm necessitates considering multiple perspectives [Honing, 2001]. In particular, the discrepancies between what is denoted in a musical score, what can be measured from an audio signal, and how a listener perceives music can be considerable [Honing, 2013]. This view is supported by Handel [1989], who underlines rhythm's phenomenalist essence: there exists no definitive "ground truth" within simple acoustic measurements. The only *reality* is determined by human listeners' judgement. Given this, it should be recognized that an approach equating rhythm solely to "musical time", while commonly accepted [Thaut, 2013], might not capture the full complexity inherent in rhythm. The hierarchical nature of music, where individual tones combine to form phrases, and phrases combine to form sections, results in expressive qualities that go beyond the sum of their parts [Cooper and Meyer, 1960]. This concept, well-established in the analysis of harmonic and melodic structure, carries equal weight in the analysis of musical rhythm across various timescales [Roads, 2001] – from small- to medium-scale temporal events, to which we specifically attend, to large-scale temporal occurrences.

Accurate modelling of rhythm at the computational level requires a deep un-

[4] *Wavelets* can be viewed as a generalisation of Gabor transforms, commonly explored in the context of music signal analysis [Dörfler, 2001].

derstanding of the interactions among its various components. Given the inherent complexity of rhythm, a deep comprehension becomes paramount. Indirect definitions provide a useful approach to this goal [Gouyon, 2005; Honing, 2013]. By breaking down rhythm into its parts and studying them from different angles, we can gain a more nuanced perspective on rhythm. Therefore, in the following section, we will dissect rhythm into its components, taking into account three perspectives: human perception, signal processing, and musical notation. Despite these perspectives potentially sharing common terminology and referring to the same concepts, they frequently represent contrasting, and occasionally conflicting, viewpoints – akin to those of the composer, performer, and listener. This understanding will not only underpin the discussion in this section, but will also lay a foundation for the remainder of this thesis.

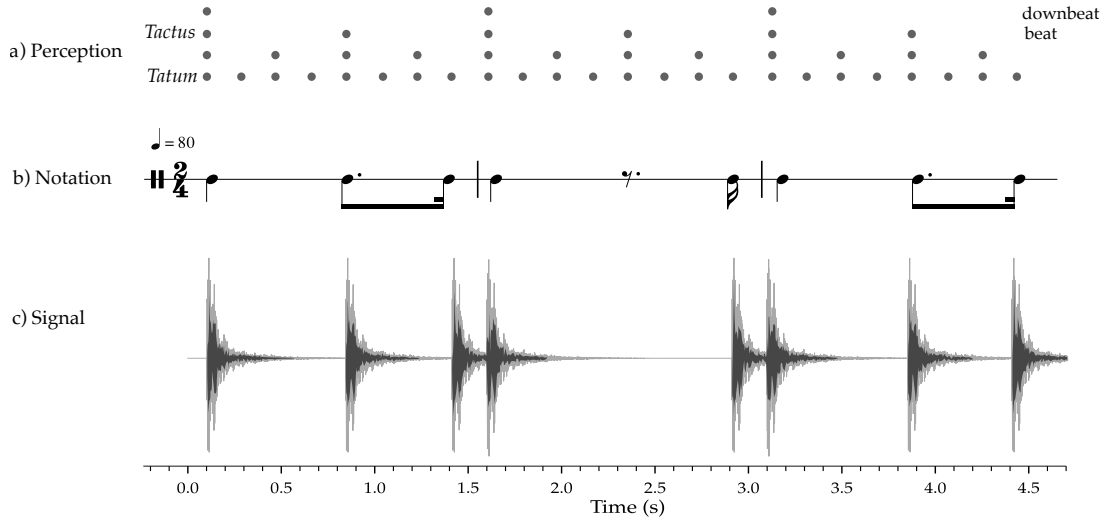


Figure 2.1: Three perspectives of a rhythmic signal: a) human perception, b) musical notation, and c) audio signal.

2.1.1 The Elements of Rhythm

In our examination, we introduce a decomposition inspired by Honing [2013]'s organisation of rhythm. This structure breaks down rhythm into four interrelated components: *rhythmic pattern*, *metrical framework*, *tempo*, and *expressive timing*. Each element significantly contributes to shaping the overall rhythmic structure of a musical piece.

(i) ***Rhythmic Pattern***

To elucidate the concept of rhythm, it is vital to distinguish between its comprehensive definition, encompassing all temporal information in music (the meaning

used thus far), and the conventional understanding of rhythm as per music theory [Cooper and Meyer, 1960]. The latter refers to a perceived category or pattern of durations, which we will term *rhythmic pattern*. To further specify, although technically any sequence of sounds or events with duration could qualify [Honing, 2013], rhythmic pattern typically refers to categories of durations that are perceptually significant within the music [London, 2004]. Rather than being governed by the actual duration of each musical event, these patterns are determined by the relative inter-onset interval (IOI) - the time between the attack-points of consecutive events.

This relative nature of the durational intervals between notes becomes evident in Figure 2.1 b), where the second IOI amounts to 3/4 of the first one. Western music notation aptly embodies these rhythmic patterns through its symbolic representation on a discrete scale, thereby emphasising the proportional nature of the durational intervals between notes.

(ii) ***Metrical Framework: Pulse and Metre***

The second element of rhythm pertains to its interpretation, which relies on a metrical framework composed of a regular pulse, known as the musical *beat*, and a hierarchical organisation of two or more pulse levels, referred to as the *metre* [Honing, 2013].

The *beat*, as the primary musical form [Bilmes, 1993], affords musicians and listeners a stable and dependable framework, critical to the establishment of a piece's rhythmic structure. It is a recurring pulse that maintains a consistent sense of time. While the terms beat and pulse are frequently used interchangeably, a key distinction exists: pulse refers to a sequence, and beat to an element [Gouyon, 2005].

The *metre* consists of a hierarchical arrangement of pulse sensations at different levels, corresponding to various time scales, as depicted in Figure 2.1 a). The most fundamental level is the *tactus*[5], providing a rhythm for us to synchronise our tapping foot with [Fraisse, 1982]. The tactus is often regarded as the "metronomic" unit of the beat, used to establish the overall tempo and pace of a piece. The

[5] Discussions of tactus in music theory date back to at least the late 15th century [London, 2004], when it was linked to resting pulse, breathing rate, or walking period [DeFord, 2015]. In literature, Adam von Fulda first mentioned tactus in his 1490 treatise *De Musica* [Brown, 1980], referring to timekeeping by hand-beating. Since then, the concept has persisted with various authors using the term tactus (e.g. Lerdahl and Jackendoff [1981]) or its synonyms, such as Cooper and Meyer [1960]'s "primary rhythmic level".

tatum[6], the smallest unit of time in a piece of music, is situated below this level and frequently considered the "atomic" unit of the beat. It refers to the shortest durational values in music that are more than incidentally encountered, often coinciding with note onsets [Jehan, 2005]. The other durational values, with few exceptions, are integer multiples of the tatum [Klapuri et al., 2006].

The intertwined processes of beat induction and metre perception enable listeners to understand the rhythmic structure of music and synchronise with its temporal patterns. Through beat induction, listeners extract a regular pulse from a sequence of sounds and use it as a reference to organise their perception of the music. In contrast, metre perception involves not only processing the beat level but also the hierarchical organisation of beats into larger metrical units, such as bars. As noted by Gjerdingen [1989], it can be understood as a "mode of attending" to rhythm.

The metrical framework of rhythm involves an active cognitive phenomenon in which listeners create expectations about musical events based on previously encountered rhythmic patterns, thereby establishing a sense of ground for perceiving rhythmic figures [Honing, 2013; Large and Kolen, 1994]. This metrical scheme is rooted in the interconnectedness of beat, metre, and rhythm in general. It is worth noting that different authors may have varying perspectives on the degree of separation or closeness between these elements, such as Hasty [2020], who does not explicitly separate metre from rhythm. Furthermore, distinctions exist between the function of metre for listeners and performers [London, 2004]; however, our discussion will concentrate on aspects of metre that apply to both, acknowledging that performers also engage in listening.

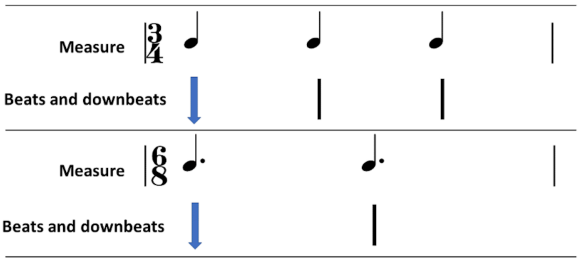


Figure 2.2: Hierarchical relationship between metre, bars, and beats, as defined in Western music theory. In both 3/4 and 6/8, the beats are indicated with vertical lines, and the downbeat with a blue arrow (adapted from Cano et al. [2021]).

[6] A derivation of "atom", as coined by Bilmes [1993, p. 21] in honour of Art Tatum, "whose tatum was faster than all others".

Metre profoundly influences our perception of musical pulse, with accentuation patterns dividing this pulse into *strong* and *weak* beats. The term "downbeat", typically associated with the downward stroke of a conductor's baton, refers to the first and usually the most accentuated beat of a bar. This downbeat has a significant effect on the rhythmic and structural elements of a composition.

In Western music theory, *time signatures* are employed to notate a part of the metrical structure, specifically the arrangement and note value of beats within each bar. Time signatures are typically expressed as a relation (e.g. a fraction), where the upper figure denotes the aggregate number of subdivisions in a bar, while the lower figure represents the rhythmic figure[7] that corresponds to one subdivision.

Metrics categorised as simple correspond to a binary beat subdivision, while those classified as compound align with a ternary subdivision. Simple metrics are often the foundation of popular music, whereas compound metrics are also widely used, though less frequently. Conversely, complex metrics, which cannot be evenly divided into consistent beat groups[8], are frequently found in contemporary or experimental genres. These complex metrics include additive (e.g., $2+3+2/8$), irrational (where the denominator is not a power of two, as in 3/10), and fractional notations (e.g., $2^{1/2}/4$). Despite the range of these signatures, it is worth noting that this system of notation may not always accurately represent certain musical traditions lacking a specific metre and periodicity [Agawu and Agawu, 1995] or where a unique metre cannot be clearly defined [Cano et al., 2021].

(iii) ***Tempo***

Tempo is defined as the impression of the speed or rate of a sounding pattern [Honing, 2013]. It is closely related to the cognitive concept of tactus, which is the speed at which the music's pulse passes at a moderate rate. This can influence our perception of the beat. A pulse is perceivable within a range of approximately 200 ms to 2 s, or 30 to 300 bpm [London, 2004], as demonstrated by a vast body of research[9].

[7] As in the American musical notation convention, note durations are named using terms that represent their proportion as a fraction of a whole note (𝅝).

[8] For example, in the simple 3/4, the beat given by the quarter note is divisible into 2 eighth notes: ♩ = ♪ + ♪. However, in the compound 6/8, the beat given by a dotted quarter note corresponds to the grouping of 3 eighth notes: ♩. = ♪ + ♪ + ♪. In the complex 5/4, the beats can be organised in alternative ways, e.g. $3+2$ or $2+3$, corresponding to different stress patterns.

[9] For example, the work of Parncutt [1994] identifies a pulse perception range of 200 to 1800 ms. A

Furthermore, the range of tempi in which rhythms are accurately perceived, i.e., understood as a perceptual unity, is limited. When music is performed too quickly, successive sounds blend together; conversely, when performed too slowly, rhythmic perception disintegrates[10] into a series of isolated sounds [Fraisse, 1982].

Tempo significantly influences a piece's character and emotional expression. However, it is not deemed a core aspect of perceived temporal organisation [Honing, 2013]. As stated by Cooper and Meyer [p.3 1960], tempo does not serve as a rhythmic "organising force". In musical notation, tempo is typically represented through one of two methods: by providing a metronome marking that denotes the number of beats per time unit (usually minutes, yielding bpm, as depicted in Figure 2.1 b)), or inversely, by specifying the inter-beat interval (IBI). Alternatively, a textual indication such as *"Largo"* or *"Andante"* may be used, providing an intended tempi range (and expressive context) that allows for some flexibility and interpretation by the performers.

Within a movement or piece, a composer may indicate a complete change of tempo, often by using a double bar and introducing a new tempo indication. Similarly, a tempo modification like slowing down may be notated as a *ritardando* or a *ritenuto*, depending on whether it is intended to be gradual or immediate. These compositional techniques can be better understood as examples of *tempo modulation*, an intentional alteration of tempo within a piece of music [Royal, 1995].

In this context, the concept of tempo is related to the notion of *basic tempo* [Repp, 1994], which is a a measure of central tendency of tempo over a complete musical excerpt (i.e., the implied tempo around which the instant tempo fluctuates, though not necessarily symmetrically). As both a psychological fact and a physical one [Cooper and Meyer, 1960], tempo is often underspecified in musical notation, allowing for flexibility and expression in musical performances. Notable examples are a prolonged note or rest of indefinite duration, notated by a *fermata* (𝄐) or a textual indication (*g.p.*), which imply the suspension of musical

notable peak of maximum pulse salience occurs around 600 miliseconds (or 100 beats per minute bpm), representing a periodicity zone of particular importance referred to as the "indifference interval" – a single duration perceived as neither too short nor too long.

[10] A conservative estimate for the lower attention span is around 100 ms between sounds, while the higher range sits at approximately 2.5 s between sounds. This range of time intervals has been referred to as the "psychological present" [Michon, 1978]. Moreover, within these tempo limits, which define perceivable rhythms and allow for synchronisation, individuals exhibit clear tempo preferences, the so-called "preferred tempo" [Mcauley, 2010].

time at the sole discretion of the interpretation, be it the performer or the conductor.

(iv) ***Expressive Timing***:

Expressive timing refers to the subtle deviations from anticipated regularity that enrich a music performance. This concept encompasses variations in duration from categorical values within musical units, such as phrases, and finds relevance in both scored music and improvisation [Palmer, 1997]. As an essential aspect of musical interpretation, expressive timing complements other techniques like dynamics and articulation, collectively contributing to the creation of a rich, emotionally engaging performance.

Music performance can be viewed as a communication system wherein composers encode musical ideas in notation, performers transform the notation into an acoustical signal, and listeners decode the acoustical signal into ideas [Kendall and Carterette, 1990]. Emphasising specific structural content is a fundamental aspect of interpretation [Clarke, 1987], with timing nuances playing a critical role in conveying temporal structure to the listener.

The relationship between the perception of rhythmic patterns and expressive timing lies in the interplay of expectation and surprise in music listening [Honing and Ladinig, 2009]. Our ability to recognise and classify rhythmic patterns allows us to discern and appreciate the nuances of a musician's expressive timing choices. Moreover, microtiming (i.e., the systematic minor deviations in the timing of individual notes), can influence the beat, imparting a performance with a "mechanical" (rigid), "laid-back" (slightly delayed timing), or "rushed" (playing ahead of the beat) quality [Honing, 2013]. Other expressive timing features that may deliberately affect the tempo include the typical deceleration at the end of phrases in Romantic-period classical music and the more extreme *rubato*, which signifies a highly flexible tempo in especially expressive passages [Gatty, 1912]

Expressive timing serves as a crucial component in music performance, enhancing the overall artistic experience for both performers and listeners. This highlights the importance of personal interpretation, enabling musicians to depart from the composer's notated indications and infuse their unique touch into the performance.

Developing computational tools for rhythm analysis fundamentally relies on understanding the elements of rhythm and their interrelationships, a task essential not only

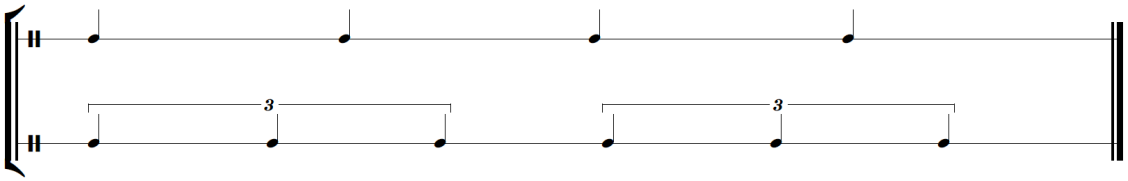
for music theorists and practitioners but also for addressing the challenge of decoding both explicit and implicit musical information. This task becomes especially critical given the diverse representations of music, such as the musical score or an audio signal. For instance, while musical notation clearly indicates the beat location, the beat is often inferred by the listener in the audio, even continuing mentally through silent portions of music [der Nederlanden et al., 2019] (this phenomenon can be observed at the 2.3 s mark in the audio signal presented in Figure 2.1 c)). Similarly, although the time signature notation conveys (part of) the metre, it cannot be directly measured from the audio signal. In this context, computational models help to bridge this gap, translating these divergent representations into a shared understanding of rhythm and timing in music.

Moreover, music perception often entails a certain degree of uncertainty, manifesting as ambiguity or subjectivity. This uncertainty can arise from the inherent properties of the music itself or stem from the individual characteristics of the listener[11]. These challenges become particularly noticeable when musical excerpts feature a high degree of *rhythmic dissonance*, referring to the compositional technique of organising and layering various rhythmic qualities to create tension within a piece of music, as defined by [Krebs, 1987][12].

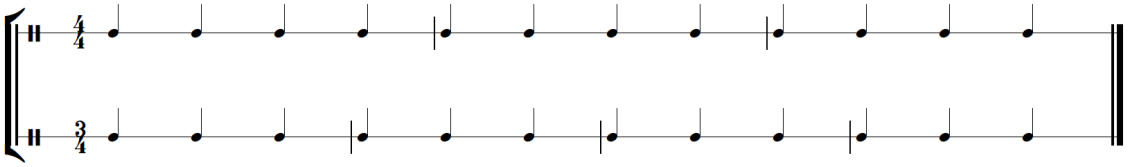
These rhythmic complexities encompass *polyrhythms*, *polymetres*, *polytempi* involve multiple rhythmic patterns occurring simultaneously. Polyrhythms and polymetres display a notable influence from African musical heritage [Agawu and Agawu, 1995], while polytempi, which utilise varying tempo layers, are more prominently tied to contemporary music compositions. Over time, these intricate rhythmic features have transcended cultural boundaries, often finding their way into both non-Western and Western music styles.

[11] Ambiguity in musical perception refers to situations where certain aspects of a musical piece, such as pitch, rhythm, or harmony, are unclear or open to multiple interpretations [Huron, 2006]. Subjectivity, on the other hand, refers to the individual differences in how people perceive and interpret various aspects of music [Clarke, 1999], as a result of personal experiences, cultural backgrounds, musical training, or cognitive processing styles [Repp, 2006].

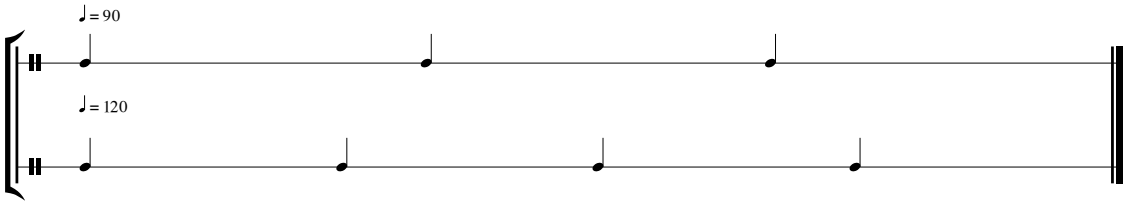
[12] In this source, Krebs delves into rhythmic dissonance in 20th-century music, establishing the following definitions: *grouping dissonance* ("type A") occurs when conflicting grouping structures in different layers have unequal divisions, causing the listener to perceive competing rhythmic groupings, while *displacement dissonance* ("type B") arises from misaligned accents or beats in different layers due to shifts or displacements between them. In Krebs [1999], the author further refines and expands on this framework, providing a thorough analysis of metric dissonance in Robert Schumann's music.



(a) "3-against-2" polyrhythms.



(b) a pattern with a 4/4 sequence played against a 3/4 sequence; after 12 beats the downbeat will align again (i.e. the patterns will come back in-phase).



(c) a stream played at a fast tempo (120 bpm), while another stream is played at a slow tempo (90 bpm).

Figure 2.3: Polyrhythm (a), Polymetre (b) and Polytempo (c).

Figure 2.3 illustrates the three concepts, which are techniques used in music composition to create complexity and tension through the concurrent use of conflicting rhythms, metres, and tempi. While there is lack of consensus both regarding the definitions and the underlying cognitive models[13], in the context of our thesis, we will adopt the following working definitions:

(a) *Polyrhythm*: involves the simultaneous application of two or more independent rhythmic layers with non-integer periodicity relationships [London, 2004]. For instance, a piece combining quadruple and triple rhythms concurrently exemplifies polyrhythm, as depicted in Figure 2.3a. This simultaneous playing of contrasting rhythms imparts an engaging tension to the musical piece.

(b) *Polymetre*: refers to the simultaneous use of differing metrical structures within a composition. The listener may find it challenging to discern the overarching structural layout due to these divergent metrical frameworks. As demonstrated

[13] c.f. [London, 2001; Galvão, 2014], for alternative definitions: the former defines polyrhythm as "the superposition of different rhythms or metres", while the latter sets apart rhythms that recur every measure (measure-preserving) and those that recur at a phrase level (beat-preserving), referring to them as polyrhythm and polymetre, respectively.

in Figure 2.3b and Figure 2.2, polymetre can significantly influence the perception of beat and downbeat locations.

(c) *Polytempo*: this term describes the use of multiple tempo layers within a single musical piece. The presence of differing tempi can enrich the composition, making the listener's perception of time and expectation more intricate and ultimately contributing to a unique auditory experience.

Due to the subjective nature of music, interpretations of rhythmic dissonance vary greatly among listeners [Galvão, 2014]. When we abstract the concepts of polyrhythm, polymetre, and polytempo, we realize that these techniques can potentially yield comparable results. As an illustration, the polyrhythm depicted in Figure 2.3a might be perceived as polytempo, with two simultaneous tempos in a 6:4 ratio (180 bpm versus 120 bpm). Similarly, a polymetre might be interpreted as a polyrhythm, or vice versa, depending on how the listener grounds the rhythmic relationships among the various musical layers. The co-occurrence of these elements within a piece can markedly amplify the experience of rhythmic dissonance [Adams, 2018].

This interpretation of rhythmic dissonance is highly individualistic and relies heavily on a listener's perception of rhythm and phrase structure. Beyond augmenting compositional and performative techniques, the complexities introduced by polyrhythms, polymetres, and polytempi pose significant challenges for computational rhythm analysis. Accordingly, these intricate musical elements will be revisited and analysed in greater depth throughout the course of this thesis.

2.2 Beat Tracking in the Context of Computational Rhythm Analysis

A long-standing area of investigation in MIR is the computational rhythm analysis of musical audio signals. Within this broad research area, which incorporates many diverse facets of musical rhythm including onset detection [Bello et al., 2005; Schlüter and Böck, 2014], tempo estimation [Cemgil et al., 2000; Schreiber and Müller, 2019] and rhythm quantisation [Cemgil and Kappen, 2003], sits the foundational task of musical audio beat tracking. The goal of beat tracking systems is commonly stated as inferring and then tracking a quasi-regular pulse so as to replicate the way a human listener might subconsciously tap their foot in time to a musical stimulus [Hainsworth, 2006; Sethares, 2007; Müller, 2015].

Beat tracking enables the computational use of musical time [Dixon, 2001b]. Given this fundamental tenet, it has found widespread use as an intermediate processing step within larger scale MIR problems by allowing the analysis of harmony [Stark and Plumbley, 2011] and long-term structure [Nieto et al., 2020] in "musical time" thanks to beat-synchronous processing. In addition, the imposition of a beat grid on a musical signal can enable the extraction and understanding of expressive performance attributes such as microtiming [Fuentes et al., 2019a]. Furthermore, within creative applications of MIR technology, the accurate extraction of the beat is of critical importance for synchronisation and thus plays a pivotal role in automatic DJ mixing between different pieces of music [Vande Veire and De Bie, 2018], as well as the layering of music signals for mashup creation [Davies et al., 2014]. In particular for musicological and creative applications, the need for very high accuracy is paramount as the quality of the subsequent analysis and/or creative musical result will depend strongly on the accuracy of the beat estimation.

In this section, we aim to present a comprehensive introduction to beat tracking, a central focus of our thesis. While doing so, we emphasise the stimulating features of beat tracking and the potential challenges it encounters, not only as a pivotal task in computational rhythm analysis but also as one that exemplifies the main difficulties posed in this field. Then, we provide a brief summary of its evolution towards the current state of the art. This evolution is structured around two main "eras": the period before the advent of deep learning and the era following the introduction of deep neural networks. Furthermore, we delve into the evaluation framework for beat tracking, exploring its significance, the metrics and methodologies used, and the

challenges in creating a framework that aptly captures human perception of beats. Finally, we scrutinise the limitations of existing beat tracking systems, identifying areas where further research is needed to enhance accuracy and applicability. By examining these constraints, we aim to provide insights into potential directions for research and development in the field of beat tracking, ultimately striving towards more sophisticated and precise systems that can accommodate a wide range of musical content and user needs.

2.2.1 Key Principles

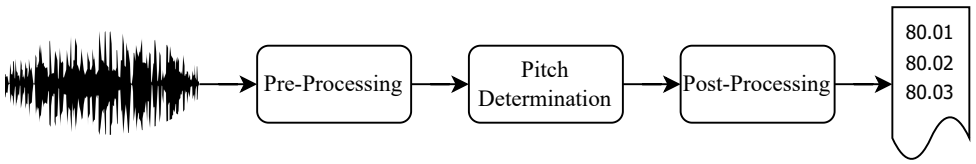
Humans perceive musical forms and structures as associative or hierarchical arrangements [McAdams, 1989]. Cognitively, we form these hierarchical structures optimally, retaining both pitch sequences [Deutsch and Feroe, 1981] and rhythmic layers [Desain, 1992] as hierarchical networks. Emphasising this stratified perspective, music can be considered a complex interplay of periodic patterns occurring across vastly different timescales, wherein pitch and rhythm emerge as essential features of this intricate structure. Within this viewpoint, pitch and rhythm correspond to the same fundamental physical process — impulses in a recurring pattern — differing mainly in the timeframe in which they operate [Adams, 2018].

This perspective aligns with Henry Cowell's theories, which examined the connection between pitch and rhythm as manifestations of the same underlying reality, albeit at dramatically different timescales [Cowell, 1958][14]. Nevertheless, this perspective does not fully account for the cognitive processing of pitch and rhythm, which involve distinct neural mechanisms [Zatorre et al., 2002; Patel, 2008], and thus, while the continuum concept offers valuable insights for audio signal analysis, it does not capture faithfully the complete cognitive aspects of music perception.

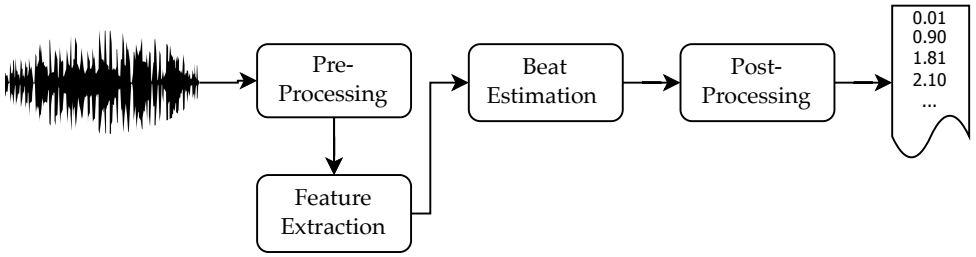
Both pitch determination and beat tracking share a common foundation: periodicity. Pulse is the primary periodic pattern in music; thus, beat tracking is essentially a task of detecting the periodicity of the music signal. Adopting this viewpoint allows us a clearer understanding of the parallels in extracting pitch and beat information from musical signals. One of the classical approaches to detecting the periodicity of a signal is through the autocorrelation function, which quantifies the degree of

[14] This idea can be traced back to Cowell's work, "New Musical Resources". By considering pitch and rhythm as part of the same temporal continuum, Cowell suggested that rhythmic structures could be informed by the relationships between pitches and vice versa. This concept led him to explore new compositional techniques that blurred the boundaries between pitch and rhythm, demonstrating that composers can apply the harmonic series to not only melodic and harmonic ideas but also rhythmic ideas.

similarity between a signal and a lagged version of itself over successive time intervals. The distance between the global maximum and the first peak provides an estimate of pitch/tempo. In the case of a simple signal, such as a pure tone, the autocorrelation function can be used to detect its (fundamental) frequency. More complex signals, however, necessitate additional pre- or post-processing steps. Nevertheless, the autocorrelation function remains one of the most established and successful methods for detecting pitch in both speech [Rabiner, 1977] and music signals [Brown, 1993]. To detect pulse periodicity, we shift from applying the autocorrelation to the signal itself to a representation containing pertinent information about the underlying rhythmic structure. The most common approach utilises the timing of musical onsets as a feature list, and applying the autocorrelation to this feature list extracts the necessary information for pulse estimation.



(a) Pitch determination.



(b) Beat tracking.

Figure 2.4: Block diagram of pitch and beat determination systems based on the autocorrelation function.a) the output is a series of "pitch" values over time (in Hz); b) the output is a series of time values (in seconds).

In both pitch determination and beat tracking building blocks (see Figure 2.4), the pre-processing and post-processing stages can serve common general goals: the pre-processing block prepares the audio signal for subsequent analysis, while the post-processing block refines the output, correcting potential errors, and adapting the results to specific musical contexts or constraints. The primary differences between the two systems lie in their respective feature extraction and central processing blocks. In beat tracking algorithms, the feature extraction stage is responsible for identifying key information related to the underlying rhythmic structure, such as onsets. In contrast, pitch determination algorithms do not include this feature extraction stage, and instead apply the autocorrelation function directly to the signal in the subsequent

processing block. As a result, although the central processing block is based on the same autocorrelation function (ACF), it obtains different information from the signal: in beat tracking, it estimates beat positions, while for pitch determination, it identifies the fundamental frequency of the musical notes present in the signal.

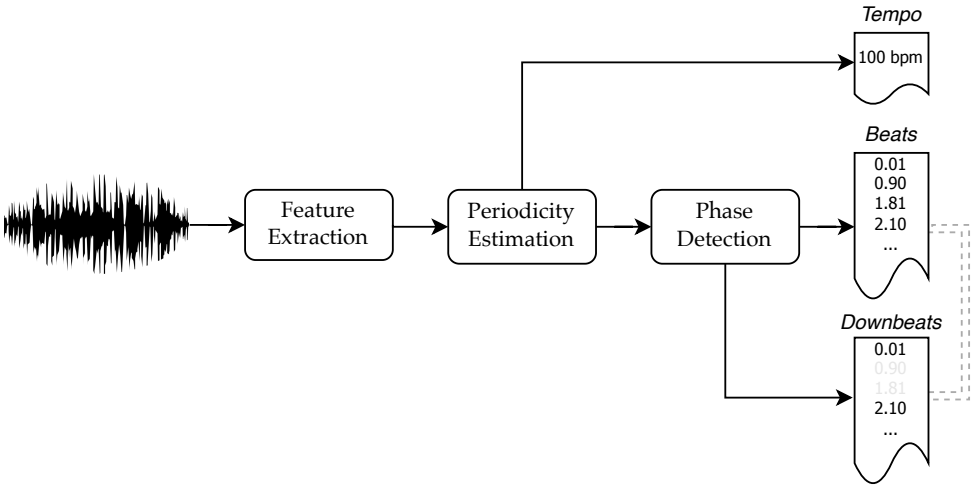


Figure 2.5: Block diagram of tempo, beat and downbeat estimation system.

Contrastingly, a unified method for simultaneous tempo, beat, and downbeat estimation follows a well-established approach [Dixon, 2001b; Laroche, 2001]. This process can be divided into three main parts, as illustrated by Figure 2.5: feature extraction, periodicity estimation, and phase detection. Initially, feature extraction is performed to identify energy changes in the audio signal, capturing the underlying rhythmic information. Subsequently, periodicity estimation is conducted by calculating the autocorrelation of the extracted features and applying peak picking to determine the most prominent tempo candidates. Lastly, phase detection refines the alignment of the estimated tempo with the extracted features, enabling accurate beat and downbeat extraction.

Autocorrelation is undoubtedly one of the oldest and most used techniques for beat detection [Scheirer, 1991; Paulus and Klapuri, 2002], as well as (or in tandem with) other rhythmic features such as expressive timing [Desain and de Vos, 1990], metre determination [Brown, 1993], or rhythmic pattern classification [Dixon et al., 2003]. In "Pulse tracking with a Pitch Tracker", Scheirer [1991] demonstrated the (almost) direct use of a "pitch-extraction" algorithm for the problem of pulse extraction, supported by the operation of the autocorrelation function.

In conclusion, the ACF presents an intuitive and effective approach for understanding beat (also downbeat and tempo) estimation in music signals. By interpreting

music signals as periodicities at different timescales, we gain valuable insights into the processes involved in extracting these fundamental musical features. Stemming from its precursor disciplines, the pursuit of periodicities and patterns, i.e. time-domain and frequency-domain approaches, has driven the advancement of algorithms and systems for the extensive field of MIR.

2.2.2 The Evolution Towards the State of the Art

Building on these foundational ideas, beat tracking has significantly evolved. In this subsection, we will offer a brief overview of how beat tracking methods have progressed from early algorithms to more sophisticated techniques that utilise Deep Neural Networks (DNN).

In the early days of MIR, beat-tracking algorithms predominantly relied on signal processing techniques and heuristic methods aimed at extracting meaningful rhythmic features from audio signals, such as onset lists or frame features [Gouyon et al., 2006]. Following feature extraction, various methods were employed to identify and track beats in the music, including adaptive oscillators [Large and Kolen, 1994], comb filters [Seppanen, 2001], multiagent systems [Dixon, 2001a; Goto and Muraoka, 1994], dynamic programming [Ellis, 2007], or even the classic autocorrelation which spans beat tracking history [Scheirer, 1998; Foote and Uchihashi, 2001; Davies and Plumbley, 2007; Böck and Schedl, 2011]. Subsequently, heuristic methods, typically based on domain knowledge, were employed to interpret the features extracted and determine the beat positions [Allen and Dannenberg, 1990; Goto and Muraoka, 1999]. While these early algorithms demonstrated reasonable success, their performance was often limited by the quality of hand-crafted features and the assumptions (and consequent constraints) made about the structure of the underlying musical signals.

In the early development of learning-based beat tracking, researchers adapted statistical methods to handle musical data, embracing the paradigm of machine learning to learn from the data. Examples of these probabilistic approaches include Hidden Markov Models (HMM)[Klapuri et al., 2006], Monte Carlo methods [Cemgil and Kappen, 2003], and state-space models [Whiteley et al., 2006]. Another problem was the lack of diversity in training and testing examples, which could lead to beat trackers being over-fitted to mainstream styles of music (e.g. pop, rock, or blues) and thus create a glass-ceiling effect [Holzapfel et al., 2012a].

In fact, at that time, researchers argued that beat tracking performance was approaching a glass ceiling with then-current algorithms stagnating at around the 80%

mark when evaluated using the least demanding metrics on common datasets [Zapata et al., 2012]. During this period, Humphrey et al. [2013a] were evaluating the constraints of music signal analysis methods, including those used for beat tracking. As they noted, hand-crafted feature design is suboptimal and unsustainable, the power of shallow architectures is fundamentally limited, and short-time analysis cannot encode musically meaningful structure. They also acknowledged that while most of the advancements in deep learning had been in computer vision, its application to MIR problems presented unique challenges[15]. Despite these challenges, the adoption of deep learning in music informatics was imminent.

Transitioning from the early concerns surrounding deep learning applications in MIR, significant strides have been made in beat tracking research, with deep neural networks becoming a popular approach to tackle the problem. Several DNN-based models have been proposed in recent years, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and their variants. These models have demonstrated superior performance in beat tracking tasks compared to their pre-DNN counterparts, owing to their ability to learn more effective feature representations, model more complex relationships in the music signal, and ultimately achieve better performance.

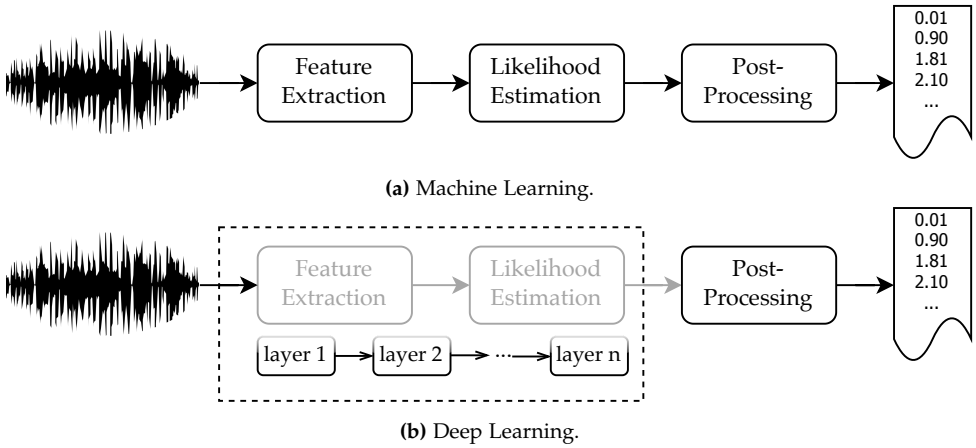
In conclusion, beat tracking has come a long way, evolving from the analysis of discrete data such as parsed scores [Longuet-Higgins and Lee, 1982] and MIDI data [Cemgil et al., 2000], to early efforts in processing continuous audio data [Schloss, 1985], and finally reaching the complex multi-task deep neural networks that compose the current state of the art [Böck and Davies, 2020]. Data-driven approaches have consistently outperformed previous methods across various fields, and beat tracking (and MIR in general) is no exception. Recognising the importance of these advancements, we will delve deeper into data-driven approaches and their impact on beat tracking in the following section.

2.2.3 Data-driven Approaches

The advancement of data-driven methodologies, especially deep learning, has considerably influenced beat tracking and other MIR tasks. In this concise overview, we first examine the distinctions between traditional machine learning and deep learning

[15] a primary concern being that although time-frequency representations are two-dimensional, they are not inherently images.

approaches[16].



(a) Machine Learning.

(b) Deep Learning.

Figure 2.6: General pipeline for contemporary beat tracking systems.

Classical machine learning techniques, shown in Figure 2.6a, often involve designing features and training a classifier to map these features to desired output labels using statistical models. In contrast, deep learning techniques, illustrated in Figure 2.6b, use multiple trainable layers learning directly from the data. The main distinction is that deep networks can learn complex relationships between input and output unattainable with shallow networks [Choi et al., 2017a]. While DNN outputs ideally provide robust estimations of beat positions, they tend to be noisy in practice. Thus, although some end-to-end approaches to beat or downbeat tracking exist [Fuentes et al., 2019b], most methods apply Probabilistic Graphical Models (PGMs) such as the well-known HMM [Rabiner, 1989] to post-process DNN likelihoods, effectively encoding musical consistency constraints during inference [Davies et al., 2021].

Deep learning methods, while initially seeming distinct from classical machine learning approaches, share fundamental similarities. Both paradigms implicitly engage in feature extraction and likelihood estimation for tasks like beat tracking [Davies et al., 2021]. In beat tracking, event-oriented features are commonly employed, assuming that variations in signal characteristics could correspond to beat locations [Degara et al., 2012; Krebs et al., 2013]. Deep learning networks can utilise either a single feature Böck et al. [2014] or multiple musical properties concurrently [Krebs et al., 2016]. Recently, the prevalent strategy has shifted towards employing combinations of logarithmic spectrograms with diverse resolutions as input to deep neural networks [Korzeniowski et al., 2014; Böck et al., 2016b].

[16] for more comprehensive and in-depth reviews, we recommend referring to the works of [McFee, 2018; Purwins et al., 2019; Davies et al., 2021]

In recent years, supervised deep learning approaches have gained prominence, diverging from the traditional formulation by explicitly relying on large amounts of annotated training data. The general deep learning approach, exemplified by Böck and Schedl [2011], formulates beat tracking as a sequential learning problem of binary classification through time, with beat targets represented as impulse trains. The objective of a beat-tracking deep neural network, typically employing recurrent and/or convolutional architectures, is to predict a beat activation function from an input representation (either the audio signal itself or a time-frequency transformation), similar to the target impulse train. To obtain a final output sequence of beats from the beat activation function, the beat activation function is post-processed using a probabilistic model. The *de facto* standard is to use a dynamic Bayesian network (DBN) [Böck et al., 2014] or HMM [Krebs et al., 2015] for inference, better handling spurious peaks or the absence of reliable information.

Deep learning significantly impacts broader MIR objectives such as evaluation, reproducibility, and dissemination. When addressing less conventional problems or application domains, practitioners must choose between manually creating (or curating) datasets and reusing existing resources. This decision has led to a growing awareness within the MIR community of the need to share audio, annotations, and evaluation data [Urbano et al., 2013]. However, due to the high demand of music licensing, the dataset availability is limited and access to audio files is restricted [Bittner et al., 2019]. In addition to ensuring data availability, the propensity of DNNs to overfit necessitates rigorous measures in data preparation, maintaining unseen test data, and facilitating meaningful evaluations of generalisation capabilities [Peeters, 2021]. Compared to the notably vast models in NLP, the limited size of publicly available datasets hinders result reproducibility and impedes the field's progression. This is in contrast to traditional machine learning approaches, which were frequently assessed on small, inaccessible datasets.

Several challenges and open questions remain. Deep learning methods are highly sensitive to data, and their effectiveness depends on the quality of the annotated data and the range of musical material they have encountered. As a result, it is difficult to predict the performance of a beat tracking system when applied to unfamiliar musical material, i.e., outside of the dataset. For example, even state-of-the-art systems that perform well on Western music have been shown to perform poorly on non-Western music [Fuentes et al., 2019a; Cano et al., 2020]. Another challenge lies in the bias towards more straightforward musical material in manual annotations of beat locations. This is due to the laborious nature of creating annotations, thus often favouring music

with a constant tempo, 4/4 metre, and the presence of drums [Holzapfel et al., 2012b; Grosche et al., 2010]. Consequently, more complex musical material, such as those with highly expressive tempo variations, non-percussive content, or changing meters, is underrepresented, and its relative scarcity in annotated datasets may contribute to poorer performance. Moreover, the majority of datasets comprise musical excerpts up to one minute in duration. This means that it is largely unknown how well these systems can track entire musical pieces in a structurally consistent manner.

In summary, the development of data-intensive approaches, has had a profound impact on beat tracking and MIR tasks in general. Conventional machine learning methods, which relied on hand-crafted features and trained classifiers, have been surpassed by deep learning methods that use multiple trainable layers and enable end-to-end learning. While deep learning has led to significant improvements in beat tracking performance, there remain challenges and open questions related to data sensitivity, bias in annotated datasets, and the ability to generalise to unfamiliar or more complex musical material. We now briefly discuss common strategies to address these challenges.

A Note on Low-Data Learning Strategies:

Data scarcity represents a major bottleneck for machine learning in general, but particularly for deep learning. Within the musical audio domain, data curation is often hindered by the laborious and expensive human annotation process, subjectivity and content availability limitations due to copyright issues, thus making the field of MIR an interesting use-case for machine learning strategies to address low-data regimes [Pons et al., 2019]. Following success in the research domains of computer vision and natural language processing, there have been a variety of approaches proposed to address this limitation in the audio domain.

Transfer learning (TL) is an inspiring strategy through which knowledge gained during training in one type of problem is used to train another related task or domain [Pan and Yang, 2010]. By leveraging previously acquired knowledge and avoiding a cold-start (i.e., training "from scratch"), transfer learning enables the development of accurate models in a cost-effective way. Early approaches to transfer learning in MIR were based on the use of pretrained models on large datasets for feature extraction and have been proposed for tasks such as genre classification and auto-tagging [van den Oord et al., 2014], speech/music classification or music emotion prediction [Choi et al., 2017b]. Another methodology involves the use of pretrained weights as an initialisation for the parameters of the downstream model. This technique, known as *finetuning*, involves retraining certain parts of the network by defining which weights

to "unfreeze" while retaining the existing knowledge in the "frozen" components. This parameter transfer learning approach has been used for the adaptive generation of rhythm microtiming [Burloiu, 2020] and for beat tracking, as a way to transfer the knowledge of a network trained on popular music into tracking beats in Greek folk music [Fiocchi et al., 2018] or as part of an interactive musical beat tracking system [Yamamoto, 2021].

Another strategy for low-data regimes is known as *few-shot learning*, which aims at generalizing from only a few examples [Wang et al., 2019], and has been used with success in audio source separation [Manilow and Pardo, 2020]. This strategy is similar to active learning, given that the user acts as an oracle, providing the model with the correct output for a small number of examples. Both paradigms have been studied for music classification tasks [Choi et al., 2019]. Lately, the association between both approaches has become widespread, with transfer learning techniques being widely deployed in few-shot classification, achieving high performance with a simplicity that has made finetuning the *de facto* baseline method for few-shot learning [Dhillon et al., 2020], in what is known as *transductive transfer learning* [Pan and Yang, 2010].

In conclusion, low-data learning strategies, such as transfer learning and few-shot learning, offer promising avenues for addressing the data scarcity challenge in MIR tasks, including beat tracking. By leveraging existing knowledge and finetuning pretrained models, these approaches enable cost-effective development of accurate models that can adapt to new content within the same task or even to entirely new tasks. As the MIR field continues to progress, exploring these low-data learning strategies will become increasingly important in order to develop models that can generalise effectively to unfamiliar or more complex musical material.

2.2.4 Open Challenges

Computational beat tracking depends significantly on the quality of the digital representation of the audio signal. In the case of low-quality signals, the lack of essential information can hinder accurate beat detection. However, our primary focus is on the numerous challenges that computational beat tracking and human beat perception share. These challenges span various musical contexts and affect all aspects of rhythm:

– *Rhythmic patterns*: computational beat tracking encounters difficulties when analysing musical textures lacking clear rhythmic cues, often characterised by soft onsets and minimal percussive content. Such features are typical in genres like ambient electronic pieces or chamber music. On the other hand, intricate

rhythmic patterns with multiple interlocking layers, such as polyrhythms found in West African traditional music or Turkish and Indian art music, also present substantial challenges [Toussaint, 2005; Srinivasamurthy et al., 2014].

– *Metrical framework*: in music with complex or irregular metres, beat tracking algorithms may have trouble accurately identifying the metre, leading to misplaced beats; particularly when the beat is subdivided in three parts (as opposed to the binary subdivision of simple metres), or intricate rhythmic patterns and shifting accents. Such complexities are often found in well-known musical cultures like Afro-Cuban [Chor, 2010] and Andean [Stobart and Cross, 2000], as well as lesser-known traditions such as Scandinavian folk fiddling [Johansson, 2010].

– *Tempo*: beat tracking faces notable challenges related to tempo, including slow tempo, gradual tempo changes (i.e., a stable tempo transitioning gradually to a different stable tempo), and abrupt tempo shifts (i.e., an immediate change in tempo)[Holzapfel et al., 2012b]. Beat-tracking algorithms often assume consistent tempo in the music they analyse, leading to difficulties when adapting to shifting tempos in music with frequent changes [Benadon, 2004]. Such characteristics are present in music spanning various epochs and genres, from baroque to pop and ambient music. Additionally, contemporary practices explore advanced tempo manipulations, as exemplified by the works of György Ligeti, Elliot Carter, and Steve Reich.

– *Expressive timing*: recognised as a fundamental challenge for beat tracking [Bilmes, 1993; Dixon, 2001b; Holzapfel et al., 2012b], expressive timing covers a broad range of rhythmic variations and interpretative choices made by musicians, including microtiming. In this regard, musicians systematically play slightly ahead or behind the beat to create a unique rhythmic feel, contributing to the distinct character of genres such as R'n'B's *groove*[Danielsen, 2010], jazz's *swing* [Friberg and Sundström, 2002], and samba's "suingue"[17] [Gerischer, 2020]. Expressive timing also encompasses other aspects like tempo fluctuations or those related to articulation and phrasing, commonly found in Romantic-period classical music [Palmer, 1997].

Computational rhythm analysis encompasses a variety of perspectives, including cognitive (as performed and perceived), theoretical (as notated in the score), and physical (as measurable in the signal) aspects. These perspectives compound the

[17] the Brazilian variant of *swing*.

complexity of the previously mentioned challenges. In terms of the signal perspective, which is particularly relevant for computational analysis, difficulties in beat tracking are further highlighted by the phenomenalist nature of rhythm, such as exemplified in cases where perceptual beat times do not align with physical event times [Dixon and Goebl, 2002]. Understanding the interplay between these perspectives is crucial for addressing the existing limitations of algorithms.

Traditional beat-tracking algorithms primarily utilised top-down (rule- or knowledge-based) or bottom-up (signal processing) approaches, with minimal [Goto and Muraoka, 1999] or no prior knowledge [Dixon, 2001a] of the music being analysed. While effective for certain signals, these methods encountered limitations when faced with more intricate rhythmic phenomena. To overcome these limitations, researchers sought to develop algorithms incorporating top-down general musical knowledge in the form of statistical information. Klapuri et al. [2006] proposed an HMM representing a three-level metrical grid of tatum, tactus, and measure, while Whiteley et al. [2006] explored Bayesian modelling of tempo, metre, and rhythmic patterns.

In addition to incorporating abstract knowledge about different rhythm facets, researchers have also attempted to integrate domain-specific knowledge about particular musical styles to enhance the effectiveness of beat tracking algorithms [Collins, 2006]. An early effort by Goto [2001] involved extracting bar-length drum patterns and matching them to known rhythmic patterns using a multiagent approach. However, the focus on popular music led to a limited scope, such as straightforward 4/4 meters. Wright et al. [2008] concentrated on Afro-Cuban clave rhythms, utilizing rotation-aware template matching combined with dynamic programming. Jehan [2005] pioneered the use of support vector regression for downbeat detection in Brazilian Maracatu "nação" rhythms and a specific case in funk music. Building upon a similar regression approach, Hockman et al. [2012] extended the application of support vector regression to downbeat tracking for the genres of hardcore, jungle, and drum and bass.

Until this point, incorporating musical knowledge into the system primarily relied on manual crafting and developer intuition. The advent of data-driven approaches, such as Peeters and Papadopoulos [2011]'s HMM-based probabilistic framework that learned a single rhythmic template from data for beat and downbeat tracking, and Krebs et al. [2013]'s model, which built upon the work of Whiteley and learned rhythmic patterns directly from data, made these algorithms more adaptable. With the growing prominence of data-driven solutions, supervised deep learning approaches emerged as the leading paradigm, heavily reliant on annotated training data. The representative method proposed by Böck and Schedl [2011] framed beat tracking as a

binary classification task in a sequential learning context, predicting a beat activation function from input representations such as the audio waveform or time-frequency transformation. Post-processing techniques ranged from basic peak-picking to advanced probabilistic approaches for accurate inference, with Probabilistic Graphical Models (PGMs) like the HMM being widely employed. These models demonstrated adaptability to diverse musical cultures [Holzapfel, 2014; Nunes et al., 2015], particularly to alternative metrical structures [Srinivasamurthy et al., 2017]. In line with this data-driven approach, beat tracking's state of the art advanced significantly, with recent methodologies [Böck and Davies, 2020] achieving top-tier accuracy scores, albeit primarily concentrating on mainstream genres such as rock, pop, dance, and jazz.

Despite these advances, challenges remain, as the performance of deep-learning approaches is increasingly contingent upon access to ample data resources. Performance is closely tied to the quality of data [Peeters, 2021], in terms of both the quality of annotation and diversity of musical style. Adaptive methods, which use pre-existing knowledge models for specific genres, could help to address these issues; however, it is important to note that, despite progress in MIR models, they still lag behind fields like natural language processing in terms of scale, which benefit from extensive datasets and numerous data samples for downstream tasks [Humphrey et al., 2017]. Fiocchi et al. [2018] assessed the transferability of beat tracking knowledge from Western music to Greek music, but this approach has shown lower performance compared to dedicated training on the same dataset [Krebs et al., 2015]. Moreover, the challenges posed by intricate datasets like the *SMC*, combined with the computational demands of the Bidirectional Long Short-Term Memory (BLSTM) RNN architecture, make streamlined model adaptation a concern.

Since the inception of MIR as a field, the majority of its models and technologies have focused on mainstream popular music within the so-called Western tradition [Gómez et al., 2013]. This emphasis has raised important concerns about the applicability of existing algorithms to a broader array of musical traditions [Cornelis et al., 2010]. Data-driven solutions, now increasingly prevalent in MIR, further compound these limitations as they rely heavily on available training data, which predominantly represents mainstream genres with limited rhythmic diversity. Consequently, algorithms often struggle when faced with these challenges [Serra, 2011]. State-of-the-art systems, while successful in most "Eurogenetic" music, have demonstrated poor performance on rhythmic material from non-Western traditions [Fuentes et al., 2019a; Cano et al., 2021], which differ from Western characteristics and conventions [Toussaint, 2019]. Moreover, obtaining annotated data for culturally specific

music traditions proves difficult, as it necessitates specialised knowledge that is hard to secure on a large scale.

Indeed, the scarcity of annotations not only affects culturally specific datasets but also has a significant impact on the majority of MIR datasets. Generating annotations is a resource-intensive process, leading many datasets to consist of musical excerpts limited to one minute or less. Consequently, the ability of these systems to accurately track entire compositions remains largely unexplored. To advance the field, it is crucial to develop suitable evaluation methods and interpretation frameworks that account for the specificities of non-Western musical traditions and address data representation limitations in various MIR datasets.

Rhythm holds a central position in numerous music cultures, often more so than in the Western tradition [Arom, 1989; Tzanetakis, 2014]. Tackling the challenges and biases in computational rhythm analysis is essential for the advancement of MIR and the creation of algorithms capable of analysing the wide-ranging diversity of musical cultures worldwide. This necessity becomes even more urgent in the context of Computational Ethnomusicology (CE) [Tzanetakis et al., 2007], a discipline dedicated to bridging the gap between diverse musical traditions and computational methodologies. In our work, CE serves as a notable application domain to which we shall revisit.

2.2.5 Evaluation

The development and validation of beat tracking algorithms is crucial for advancing MIR. Model improvements on the long term are bound to systematic evaluations, often accomplished through benchmarks or community-focused contests such as the MIREX[18]. The evaluation of beat tracking algorithms serves a dual purpose: it allows developers to assess their models' performance and compare them with existing approaches, and it provides valuable insight into the strengths and weaknesses of these algorithms, guiding improvements [Davies and Böck, 2014]. In this section, we discuss the predominant evaluation approaches, metrics, and considerations for interpreting results in the context of beat tracking research.

[18] An example of such a contest is the Music Information Retrieval Evaluation eXchange (MIREX), an annual community-driven evaluation campaign for MIR, where researchers and practitioners can submit their algorithms for a range of tasks, such as beat tracking. The submissions are then evaluated using standardised metrics, and the results are discussed and published to foster the advancement of the field. More information can be found at https://www.music-ir.org/mirex/wiki/MIREX_HOME

Strategies and Methods

Two primary strategies have been employed in evaluating beat tracking algorithms: *subjective* and *objective*.

Subjective evaluation plays an important role in various artificial intelligence domains, including audio signal processing [Fletcher and Munson, 1933; Terhardt, 1974]. In the realm of music, which is inherently subjective, evaluating elements like beat or tempo demands such subjective assessment techniques [Serra et al., 2013]. This evaluation approach often involves human raters who listen to the outputs produced by algorithms and rate them based on their perceptions. For instance, in beat tracking, beat times are often represented as *clicks*, which are mixed with the original audio and then played to a listener for evaluation. While this method provides valuable qualitative insights, it does come with its set of challenges. Subjective evaluations can be time-consuming, expensive, and difficult to replicate due to the absence of straightforward, unambiguous criteria [Dannenberg, 2005].

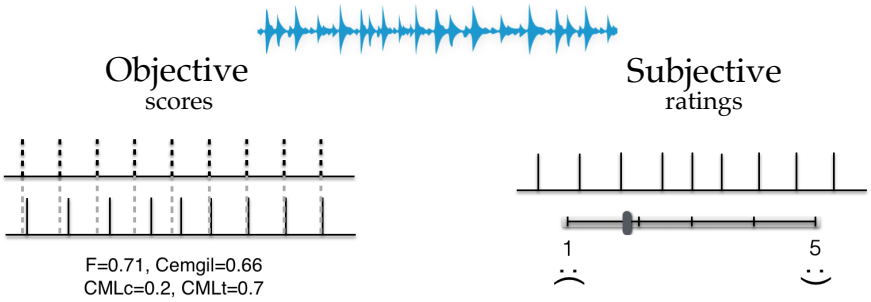


Figure 2.7: Overview of *objective* vs *subjective* strategies in beat tracking evaluation (adapted from Davies et al. [2021]).

Objective evaluation, on the other hand, relies on quantitatively comparing annotated *ground-truth* beat sequences with algorithm estimates. These accuracy scores provide valuable and effective insight into an algorithm's performance, provided that it aligns with the manner in which humans perceive and interpret music [Dixon, 2001b]. A range of have been proposed (for a thorough and comprehensive account, we refer the reader to Davies et al. [2009]), including:

– Goto: Goto and Muraoka [1997]'s approach yields a binary evaluation (1–*correct* or 0–*incorrect*) based on heuristic criteria applied to the annotation interval-normalised timing error[19]. To qualify as *correct*, three conditions must be satisfied

[19] determined by measuring the timing error between ground-truth annotations and beat estimates, with

within a 25% region of the annotations: the maximum error must be less than $\pm 17.5\%$ of the IAI, and both the error's mean and standard deviation must be within $\pm 10\%$ of the IAI.

- Cemgil: developed by Cemgil et al. [2000], this method computes a continuous score in the range $[0, 1]$. It places a Gaussian error function around each ground-truth annotation to penalise estimates with larger distances. The accuracy is determined as the sum of the errors of the closest beat to each annotation, normalised by the greatest of the following quantities: the number of beats or annotations.

- P-Score: introduced by McKinney et al. [2007] for the 2006 MIREX beat tracking evaluation campaign, the P-Score provides a continuous accuracy measure in the range $[0, 1]$. This score is calculated by measuring the normalised sum of the cross-correlation between two impulse trains representing the ground truth and the extracted beats, over a range covering 20% of the median IAI.

- Information Gain: proposed by Davies et al. [2009], this method computes a non-negative score within the range $[0, \log_2 n\_bins]$, where *n_bins* represent the number of timing errors histogram bins, and is measured in bits. It is determined by the Kullback-Leibler divergence (or relative entropy) from a uniform histogram.

Objective metrics for beat tracking evaluation have evolved to better assess the complexity of the task and align with human perception (although some earlier approaches, such as Goto's *binary* evaluation, may not capture this effectively). Throughout the body of research, a group of metrics has emerged as the common ground for evaluating beat tracking performance. These principal objective metrics provide a set of complementary perspectives for comparing different beat tracking algorithms. They include the *F-measure* and the continuity-based metrics: *CMLc*, *CMLt*, *AMLc*, and *AMLt*, which we will now present.

F-Measure The F1-score, or F-measure in the context of beat tracking, is the harmonic mean of precision (P) and recall (R). It offers a balanced assessment of an algorithm's

errors normalised to half the width of the current inter-annotation-interval (IAI) [Davies and Böck, 2014]

performance by combining these two metrics:

$$Fm = 2 \cdot \frac{P \cdot R}{P + R} = 2 \cdot \frac{\frac{t^+}{t^+ + f^+} \cdot \frac{t^+}{t^+ + f^-}}{\left(\frac{t^+}{t^+ + f^+}\right) + \left(\frac{t^+}{t^+ + f^-}\right)} = \frac{2 \cdot t^+}{2 \cdot t^+ + f^+ + f^-} \tag{2.1}$$

Here, t^+ represents the number of *true positives*, f^+ is the count of *false positives*, and f^- stands for the number of *false negatives*.

Building on the foundational concepts of the F1-score, Dixon [2001b] introduced a measure of accuracy (Acc) tailored for the specific evaluation needs of beat tracking:

$$Acc = \frac{t^+}{t^+ + f^+ + f^-} \tag{2.2}$$

By the studies presented in Dixon [2006], the F1-score was firmly established for beat tracking evaluation and has since been referred to as the F-measure. A distinctive element when applying the F-measure to beat tracking is the incorporation of a *tolerance window*. This window delineates a time interval centred around the ground-truth beat annotations. Within this interval, an estimated beat is accepted as a true positive. The specific evaluation context sets the dimensions for this tolerance window. However, a widely accepted benchmark is a window of ± 70 ms around the ground truth, as put forth by Dixon [2006] for beat tracking evaluation[20].

The F-measure offers a holistic view of an algorithm's performance yielding a continuous value in the range $[0, 1]$. However, within the context of beat tracking, it reveals some limitations concerning its alignment with human perception. These include the inability to capture long-term consistency and other perceptually relevant aspects, such as the insensitivity to beat phase.

Continuity-based methods The fundamental principle of this approach is the adoption of *continuity* as a further criterion to the correctness of the beat output [Davies and Plumbley, 2007], a concept introduced by Hainsworth [2004] and Klapuri et al. [2006].

Continuity-based metrics are characterised and set apart by two fundamental properties. First, continuity is enforced by the creation of tolerance windows of $\pm 17.5\%$ of the current inter-annotation-interval (IAI) around each annotation, i.e., these windows are calculated in a relative, *beat*-proportional manner, rather than being fixed in size as in the F-measure. The closest beat to each annotation can only be considered

[20] Notably, for the *onset detection* task, Dixon [2006] recommended a narrower tolerance window of ± 50 ms for F-measure calculations.

correct if a) it falls within this tolerance window and b) the previous beat is also within the tolerance window surrounding the previous annotation, thereby addressing beat phasing issues. A further requirement ensures consistency between the annotations and the beats. Specifically, the IBI must fall within a specific range surrounding the IAI, as determined by the tolerance window. Second, to account for ambiguity in metrical levels, the annotation sequence can be resampled to accommodate "accurate" predictions at perceptually similar beat locations. These metrical variations[21] represent common alternative interpretations of the main metrical level. Consequently, while they might be considered errors from an objective standpoint, their impact on the perceived beat tracking performance is less severe:

(i) *Same* metrical level: 180°out-of-phase (*Offbeat*), as opposed to the (in-phase) ground truth;

(ii) *Even* relation with the annotated metrical level: Twice (*Double*) or half the annotated metrical level, taking every other annotated beat and starting on the first (*Half-Odd*) or on the second beat (*Half-Even*);

(iii) *Odd* relation with the annotated metrical level: Three times (*Triple*) or one-third the annotated metrical level, taking every other annotated beat and starting on the first (*Third-1*), second (*Third-2*) or third beat (*Third-3*);

The two defining features of continuity-based metrics, namely, addressing metrical ambiguity and emphasizing continuity, are embedded in their naming structure as YMLx. In this notation, ML stands for metrical level, Y denotes the metrical level type – either correct (C) or allowed (A) —, and x signifies the calculation of either the ratio of total (t) continuously correct segments to the length of the excerpt or the ratio of the longest continuously correct segment to the excerpt length (c). Therefore, there are four metrics: the stricter CMLc, CMLt, and the more relaxed AMLc and AMLt; each providing a continuous score in the range $[0, 1]$

Objective Evaluation Scores: Characteristics and Limitations The appropriate selection of objective metrics is crucial in obtaining a comprehensive perspective on algorithm performance. Specifically, in the field of beat tracking, it is essential to understand and consider the characteristics of each metric to obtain a balanced understanding of the methods under analysis.

[21] we adopt the terminology used in the `madmom` [Böck et al., 2016a] package for evaluating beat tracking algorithms, which can be consulted at `https://github.com/CPJKU/madmom`.

The F-measure exhibits limitations in reflecting the hierarchical nature of rhythm, as it doesn't fully account for alternative interpretations that might still have musical significance. In a simple metre, predicting twice as many beats as the actual beats (double tempo) results in an F-measure of 2/3 (approximately 0.67), while predicting half as many beats (half tempo) leads to a score of 1/2 (0.50). These scores do not echo the (lack of) severity and balance of such alternative interpretations, emphasizing the F-measure's limitations in capturing the true perceptual nature of music. Moreover, these values depend on the type of tolerance interval, whether absolute or relative, and its size. Modifying the size of the tolerance window can significantly influence these metrics. A wider window might classify poorly localised beats as accurate, whereas a narrower window could categorise perceptually valid beats as errors.

Continuity-based metrics offer additional insight into algorithm performance. The CMLc score indicates that an algorithm consistently selects the annotated metrical level with well-aligned estimates. When the CMLt score is greater than the CMLc score, it suggests the presence of occasional misplaced beats, which may not necessarily be *errors*. The AMLc and AMLt scores accommodate inaccurate metrical level choices. A high CMLc score implies that the algorithm has chosen the same metrical level as the annotated one, reducing the likelihood of selecting an *allowed* but unsuitable metrical level for the piece. Moreover, it demonstrates that beat estimates were consistently well-aligned with ground-truth annotations. In instances where CMLt significantly surpasses CMLc, this could indicate an unusual misplaced beat, which might be an isolated poor annotation that needs correction, rather than an *error*.

In summary, we have examined beat tracking evaluation and highlighted the central role of objective metrics. Specifically, the F-measure and continuity-based methods, which will serve as our primary benchmarks throughout this work. Despite offering crucial insights into algorithm performance, we showed that these metrics also bear limitations, suggesting potential for development within our specific research domain.

2.3 The Role of the User

The evolution of the user's role in both Music Information Retrieval (MIR) and Machine Learning (ML) forms the basis for this section. These research disciplines have shown a shift towards a more human-centred perspective over time.

In this section, we start by addressing the evolving role of the user throughout different stages of MIR development. We focus on the move towards user-centric practices into rise of creative music systems with a strong emphasis on interactivity and user engagement. Subsequently, we examine the multifaceted roles that users have in ML, specifically their involvement in Human-in-the-Loop (HITL) methods such as Active Learning (AL) and Interactive Machine Learning (IML), which bring the user into the (machine) learning process and harness the user for improved model training.

We next focus on the role of the user as an annotator, a role that has become increasingly vital in the supervised-learning paradigm that characterises current MIR solutions. Beat annotation will be showcased as an example. This exploration includes an overview of beat annotation, the inherent uncertainty within the process, and the approaches to assessing the quality and consistency of annotation. We underscore the essential role of user involvement, particularly in situations where data is scarce, for enhancing both system performance and user satisfaction.

To conclude, we position our work within the broader context of MIR research. We outline the main strategies and methodologies that we intend to employ to craft effective, user-centred solutions for computational rhythm analysis.

2.3.1 The User in MIR and ML

The User in MIR The paradigm shifts in the field of MIR have been aptly described as "MIRAges" by Herrera-Boyer [2018]. Throughout these phases, the evolution of this field has consistently moved towards a more human-centred perspective, transitioning from the initial age of engineered features, to incorporating semantic descriptors, advancing to context-aware systems that rely on information provided by the user, and ultimately culminating in creative systems, where, by definition[22], the user is at the core. The age of creative music systems emphasises the user's role and benefits from

[22] Although Fiebrink and Caramiaux [2018] posit that creative systems can be interpreted in two alternative ways, i.e., as systems exhibiting autonomous creativity, which generate novel content without direct human input, or as systems augmenting human creativity, serving as tools for creators who ultimately make decisions, we focus on the latter interpretation, where users play a central role.

the convergence of several research communities, particularly those of MIR, AI, and HCI [Fiebrink and Caramiaux, 2018].

Initially, users primarily served as passive recipients of musical content [Serra et al., 2013], functioning mainly as end-users. However, the emergence of digital music in the 1980s and 1990s transformed the field, leading to a significant increase in available musical data. This shift not only enabled users to search, retrieve, and interact with musical content in novel ways, but also marked the beginning of utilising user-generated data, such as listening habits, preferences, and social interactions, to enhance system performance and personalisation [Lamere, 2008][23]. Consequently, the path towards user-centred design emerged as a vital approach, engaging end-users in the development and assessment of MIR systems [Schedl and Flexer, 2012].

The next stage was marked by the integration of user-generated content, such as annotations, tags, and metadata. This integration empowered users to actively participate in the organisation and discovery of music, fostering a greater sense of community and shared experience [Bertin-Mahieux et al., 2011]. During this phase, various forms of user feedback, such as subjective ratings and implicit feedback based on user behaviour, were utilised to adapt and enhance the performance of MIR systems. Recognising the importance of personalisation in music retrieval systems and the highly subjective nature of musical preferences, researchers began to focus on developing user-centric approaches that took into account individual, interest group, cultural, and global contexts [Schedl and Knees, 2013]. Concurrently, the advent of deep learning profoundly transformed MIR. These techniques required substantial amounts of annotated data to effectively train data-driven models. With the increased demand for ground truth via annotations, users assumed a more crucial role in refining the performance and utility of MIR systems as creators of this indispensable data [Schedl et al., 2013].

In the current stage of MIR, creative systems have significantly evolved to prioritise interactivity, positioning users at the centre of the creative process [Humphrey et al., 2013b]. These systems enable users to actively shape and manipulate musical content, fostering a more immersive and engaging experience. The domain of music creation and performance is naturally interactive, as musicians are accustomed to receiving immediate feedback when interacting with a musical instrument [Amershi et al., 2014]. Therefore, it is not surprising to find interactive musical systems such as

[23] In the context of multimedia retrieval, research focused on advancing techniques to support users in the interactive retrieval process, addressing areas such as semantics, i.e. bridging the gap between low-level features and high-level semantics, leveraging context for improved retrieval, and adaptation to user needs.

the Wekinator[24] [Fiebrink and Cook, 2010]. Research into the development of these systems has revealed that users can adapt their behaviour to achieve specific goals, enabling them not only to make relevant judgements about algorithm performance and interactively improve trained models but also to learn to provide more effective training data [Fiebrink et al., 2011].

The User in Machine Learning Building upon the evolving role of the user within Music Information Retrieval, it is equally crucial to consider the broader context of Machine Learning. The interplay between ML and MIR is substantial, and the trend towards user-centric practices manifests strongly within both domains.

In this landscape, the prominence of Human-Computer Interaction (HCI) is evident, primarily driving the emergence of Human-centred Machine Learning (HCML). This approach is deeply anchored in HCI principles, which have long championed the understanding of users as paramount in the design and development of technologies [Rex Hartson, 1998]. In HCML, the design of ML systems concentrates on enhancing human abilities and aligning the system's objectives with human needs. The aim is not just the development of mathematically sophisticated models but also to incorporate human understanding into the development and application of these models [Gillies et al., 2016].

Transitioning from HCML, we encounter another significant concept, Human-in-the-Loop Machine Learning (HITL). This approach, while also informed by HCI principles, originated in fields like robotics, where human operators were integral to real-time system control [Sheridan, 1992]. The critical differentiation here is the level of user involvement: whereas the former is primarily focused on tailoring the design of systems around user needs and preferences, the latter elevates user engagement to an active role in the learning process [Kamar, 2016].

In Human-in-the-Loop approaches, human users actively contribute to the learning process of the model, offering real-time feedback or correcting its predictions, enabling more effective learning and adaptation to complex scenarios, such as those with data scarcity [Holzinger, 2016]. Depending on the control exerted over the learning process[25], whether by the model or shared by the model and the human (the so-called

[24] Wekinator: an open-source software for building interactive musical systems, such as new instruments or computer listening systems, using machine learning. It enables users to create systems through demonstrations of human actions and computer responses rather than writing code. Available at http://www.wekinator.org/.

[25] Aside from control, humans can also be involved in the learning process in other ways. Firstly, as the focus of the design of the interactions and behaviours that compose the human experience around the AI models [Xu, 2019]: *Usable AI*, focusing on ensuring that AI systems are usable by the people

hybrid mode) we can identify two methodologies: *Active Learning* (AL) and *Interactive Machine Learning*, as depicted in Figure 2.8[26].

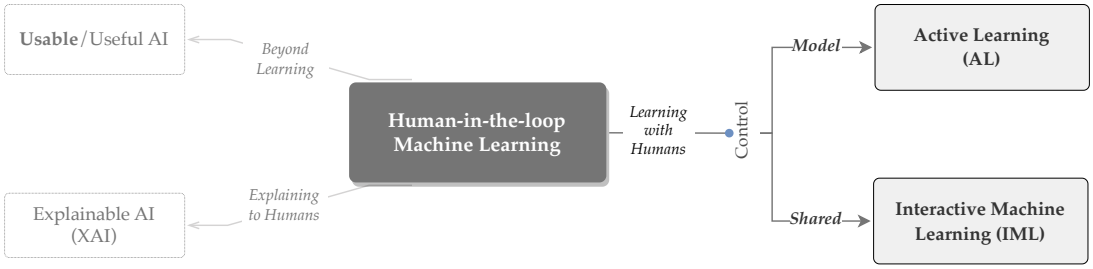


Figure 2.8: Human-in-the-loop Machine Learning.

Active Learning (AL) is a strategy that leverages user input to optimally select the most informative samples for model training [Settles, 2009]. Unlike traditional supervised learning where users provide initial input like annotations or ground-truth labels with their role confined to this initial stage [Fu et al., 2011], AL brings users into the iterative learning process. This interaction enhances model performance by focusing on the most valuable labelled data points [Settles and Craven, 2008]. As a result, the demand for large labelled datasets is reduced, while the performance of machine learning models is improved [Olsson, 2009].

In Interactive Machine Learning (IML), the emphasis is on the constant and dynamic exchange between the user and the model [Fails and Olsen, 2003]. Within the IML framework, the user and the model engage in a steady dialogue, with the user providing input, and the model adjusting its predictions or representations in response. IML can be divided into *Reinforcement Learning*[27] and *Preference Learning* [Mosqueira-Rey et al., 2022]. When faced with a scarcity of training samples, preference learning can utilise an "expert-in-the-loop" approach. This method stresses the role of domain experts in guiding the learning process, offering valuable input and feedback to improve model performance, even with limited training data.

interacting with them, and *Useful AI*, going further by trying to make AI models useful in a broad sense, i.e. to the society in which they are embedded. Also, particularly for critical domains (e.g., healthcare), it is advisable to make the results of AI systems more understandable to humans. Within Explainable AI (XAI) the aim shifts from the accuracy of an algorithm in solving a problem to its ability to justify why a given solution was chosen [Barredo Arrieta et al., 2020].

[26] For the sake of completeness, a third category corresponds to when the control is being exerted by the human is named *Machine Teaching*, referring to the idea of a teacher who teaches an ML model to an ML algorithm Mosqueira-Rey et al. [2022].

[27] This technique which primarily addresses tasks requiring sequential choices where the optimal solution is not readily discernible. In this paradigm, users deliver feedback through a predefined reward structure, allowing models to learn optimal strategies via a process of trial and error [Richard S. Sutton, 2014].

A critical aspect of user participation in machine learning, particularly relevant to our work, involves the evaluation of models. In IML systems, assessing model performance is essential, as users rely on it to determine necessary improvements [Fiebrink et al., 2011]. Involving users in this evaluation process ensures that models align with user expectations and requirements [Amershi et al., 2014]. Moreover, it is crucial to account for the human factors and cognitive aspects of user engagement with machine learning. Gaining insights into user perceptions and interactions with models can guide the development of more effective and user-oriented systems [Dudley and Kristensson, 2018].

Various HITL methods are often employed simultaneously, exemplifying their intersection and integration in numerous applications, particularly in tasks related to sound and music. Early instances of AL have emerged in music retrieval [Mandel et al., 2006], multimedia annotation [Wang and Hua, 2011], and music auto-tagging [Choi et al., 2017b]. Likewise, pioneering examples of IML within MIR can be found in the works of Fiebrink and Cook [2010]; Fiebrink et al. [2011]. A notable example is Kim and Pardo [2017, 2018]'s work on sound event detection through IML, which drew inspiration not only from common music annotation tools but also from an interactive system for electro-acoustic music analysis [Gulluni et al., 2011]. Their research laid the groundwork for further exploration at the confluence of Human-in-the-Loop and Few-Shot Learning, as evidenced by the work of Wang et al. [2020a,b, 2021]. This exemplifies the collaborative nature of MIR research, fostering a bidirectional exchange of ideas, techniques, and methodologies with other related domains. This mutual sharing and adaptation contribute to innovation and progress across MIR and other fields alike.

In conclusion, the evolution of the user's role in both Music Information Retrieval (MIR) and Machine Learning (ML) has seen a shift towards human-centred approaches, with users becoming increasingly integral to the development, evaluation, and performance of data-driven systems. Within the current deep-learning paradigm, users play a crucial role in providing annotated data, which is indispensable for training and evaluating models, particularly in subjective domains. As seen in HITL approaches, by actively participating in the machine learning process, users contribute their preferences, helping to enhance both performance and applicability of machine learning models, which proves to be decisive in low-data scenarios. As the field continues to progress, the integration of human factors and cognitive aspects in data-driven MIR systems will be essential in fostering more effective and user-oriented solutions.

2.3.2 The User as Annotator: The Case of Beat

The process of beat annotation is essential for the development and evaluation of data-driven algorithms in the field of MIR. Beat annotations are obtained by deriving temporal locations, and when handling downbeat annotations or situations where metrical positions are crucial, assigning a metrical label to each beat.

Beat annotations are typically generated through an iterative process [Davies et al., 2021][28], that can be synthesised as follows:

1A. (Manual): *The annotator listens to the music excerpt and taps along in real-time to identify beat locations*[29].

1B. (Semi-automatic): *A beat tracking algorithm is employed to generate initial beat estimates for the music excerpt*[30].

2. *The excerpt is iteratively re-examined, and local temporal imprecisions are corrected through auditive and/or visual inspection*;
3. (Optional) *Labels are assigned to mark downbeat locations or metrical positions within each bar*;
4. *A final review of annotations is carried out, and results are exported*.

The above description provides a generalisation of two approaches to the task of beat annotation: manual and semi-automatic. This framework helps to understand the key steps involved in the annotation process, but it is important to note that the actual process can vary significantly depending on the user's workflow, which includes any supporting software tools. For instance, some annotators may prefer to begin with a clean slate listen before attempting to tap along (an optional step 0), while others may use software like `Sonic Visualiser` to visually inspect the audio waveform or spectrogram to identify beat locations more accurately. Additionally, several factors can impact the duration, complexity, and quality of annotating music excerpts for beat tracking.

Regarding manual annotation (step 1A), the overall duration depends on the quality of the initial real-time taps. However, even for simple music with flawless tapping, it is necessary to complete at least two full listens: one for tapping and another for confirmation (step 1A and step 4, respectively). When dealing with challenging musical

[28] for additional insights and detailed explanations, readers are encouraged to visit the comprehensive resource at: `https://tempobeatdownbeat.github.io/tutorial/`, which includes a thorough examination of the annotation process.

[29] e.g. using annotation software like `Sonic Visualiser` [Cannam et al., 2010].

[30] e.g. using libraries such as `librosa` [Mcfee et al., 2015] or `mir_eval` [Raffel et al., 2014].

excerpts, the number of subsequent edits increases, depending on the annotator's expertise and exposure to both the musical content [Honing and Ladinig, 2009] and the tapping activity [Repp, 2005], which can be influenced by human motor noise and system jitter. Furthermore, achieving very high temporal precision may require a more in-depth examination of the waveform, thus extending the annotation time.

An alternative approach involves using an existing beat tracking algorithm to generate initial beat estimates (step 1B). This method saves time and eliminates tapping-related inaccuracies for musical materials that align with current algorithms' capabilities. However, for complex materials beyond the scope of available beat-trackers, such as those featuring high expressiveness, metrical complexities, or *difficult* properties like unclear onsets in non-percussive signals, an initial automatic pass may offer limited value. Although manual annotation of these complex signals is challenging and time-consuming, the semi-automatic process faces two additional drawbacks. First, the precision of the beat locations will be constrained by the algorithm's frame rate, potentially requiring further manual adjustments for finer temporal resolution. Second, the algorithm's selection of metrical level might influence the annotator's perception, who could have targeted a different metrical level if listening with a blank slate.

Several strategies have been proposed to enhance the efficiency of the beat annotation process, particularly focusing on the correction steps. Valero-Mas and Iñesta [2017] explored user involvement in the onset detection process, demonstrating that interactive strategies can reduce correction workload. However, this reduction was not accomplished by all interaction configurations, and in some cases, the interactive algorithm required more user effort than manual correction. Driedger et al. [2019] proposed an automation approach for beat annotation by snapping manually-tapped beats to the onset detection or beat activation function peaks, improving the subjective quality of the annotations. Their method also included a visualisation tool for identifying *regions of interest*, i.e. areas in the signal that contain potential errors or are intrinsically challenging, thus aiding the annotator through the correction steps. Despite its advancements, the approach's effectiveness depended on the presence of peaks in proximity to correct locations, a challenge that is also encountered with existing beat-tracking algorithms.

These studies underscore the notion that the process of annotation is both demanding and reliant on a certain degree of expertise. In our work, we found that annotating an expressive piece with a duration of 4 minutes and 51 seconds required approximately 15 hours, spread over three days, and involved frequent consultations with musical experts [Pinto et al., 2021]. This equates to an almost 200-fold overhead.

In contrast, Davies et al. [2021] reported that for a relatively simple and concise music excerpt of about 25 seconds, the total time taken to complete the annotation was roughly four minutes, representing a tenfold increase in time. Although the disparity between these examples (10x to 200x) is striking, it merely demonstrates the influence of music complexity on annotation time.

The significance of efficient and accurate annotation processes in MIR cannot be overstated. The field advances through iterative research cycles, rendering the development and assessment of diverse datasets essential. The emergence of deep-learning architectures, which depend heavily on substantial amounts of data, has further underscored the importance of creating and refining datasets for specific tasks such as beat tracking.

Uncertainty in Beat Annotation

Uncertainty in beat annotation manifests in the forms of subjectivity and ambiguity, both of which arise as emergent properties from challenging music signals. We illustrate these aspects with examples drawn from the development of real datasets.

To exemplify the concept of *subjectivity*, we consider the case of the *ACMUS-MIR* dataset [Mora-Ángel et al., 2019]. This dataset comprises Andean Colombian music, including the Bambuco genre, with its known bi-metric nature (3/4–6/8). To account for this specificity, the dataset features independent beat annotations for the two predominant meters, each assuming a unique underlying metre.

Regarding *ambiguity*, we explore the complexities of annotation, as evidenced by the internal data of multiple annotators during the development of the *SMC* dataset [Holzapfel et al., 2012b]. Created a decade ago to test then state-of-the-art beat tracking algorithms with demanding musical audio examples [Davies et al., 2021], this dataset continues to pose considerable challenges for even the most advanced contemporary methods. Figure 2.9 compares the *spontaneous* taps, i.e., the initial annotations made during a clean slate audition without subsequent corrections, from the five primary annotators of this dataset. The varying quality of these annotations underscores the impact of differing levels of exposure and expertise, both of a musical and technical nature.

In the first musical excerpt, subjects A_1, A_2, and A_5 produced beat annotations of similar quality, while A_3 and A_4 chose the double metrical level, leading to the octave error. The second excerpt showed more subjectivity due to its complex musical content. The expressive passage between 15 s and 25 s highlights the inherent challenges of beat

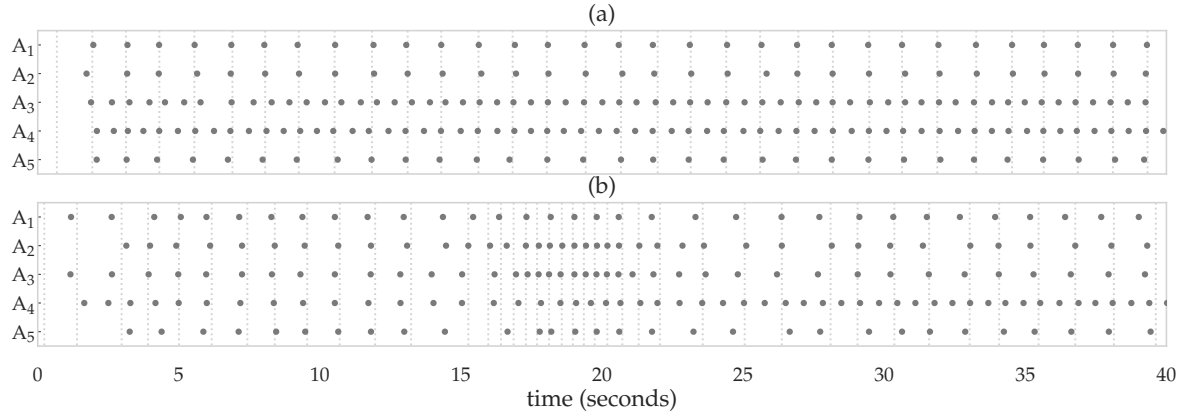


Figure 2.9: Spontaneous taps for five annotators. The ground-truth annotations are shown as dotted vertical lines. (a) Simpler excerpt with regular pulse: SMC_001 - initial 40 s of Johann Pachelbel *Canon in D Major*, by the Württemberg Chamber Orchestra (b) Complex excerpt with an irregular pulse: SMC_006 - first 40 s of Timo Korhonen's guitar-solo interpretation of Heitor Villa-Lobos *Étude №11: Lent*.

annotation: all subjects failed at the beginning of this segment (an abrupt "decrease" in tempo, due to a time suspension), marking a non-existent beat; throughout the rest of this segment, we observe varying degrees of quality, with A_5 being the least accurate and A_2 or A_3 the most accurate. Subject A_4 displays an interesting pattern, first identifying the right metrical level and then shifting to the immediately higher level. In both excerpts, annotating the first two beats proved difficult, as annotators need time to *tune* into the beat and tap it: none of the subjects identified the first beat in either excerpt, which is natural for a first-glance tapping; in the second excerpt, the first (ground-truth) beat occurs at approximately 0.2 s, insufficient for accurate perception and motor execution.

However, if we evaluated these annotations solely through the F-measure, or almost any other objective metric, for that matter, it would be very difficult to observe the same patterns in the quantitative scores. For example, the "best" annotators (A_1, A_2, and A_5) have respective scores of 0.889, 0.698, and 0.438; while the "worst" annotators (A_3 and A_4) have scores of 0.283 and 0.426.

There is substantial variation among subjects' judgements, which remains evident even when accounting for the unprepared nature of the annotations. These observations bring to light the common perception-related challenges in beat annotation, such as the attraction to different metrical levels [Mckinney and Moelants, 2006]. Such findings serve to emphasise the potential ambiguity when evaluating MIR systems that rely on human-annotated ground truth [Flexer, 2014]. This problem is not exclusive to beat tracking, as similar challenges have been highlighted in other tasks, including metre

analysis [Quinton et al., 2015] and musical segmentation [Nieto et al., 2020].

Furthermore, in our high-level analysis, we have assumed that the initial annotation aims to closely resemble the final annotation. However, an experienced annotator may prioritise fewer but more accurate annotations to minimise correction effort, which could explain the observed pattern of skipping every other beat in difficult parts for some subjects.

In this context, the limitations of the F-measure for beat tracking become even more evident, and cannot be resolved by minor adaptations[31]. Moreover, the limitations of continuity-based metrics are also apparent: while they might perform better in certain cases, they still fail to capture the annotator's workflow perspective. The F-measure is calculated using correct, missed, and erroneous beat annotations (i.e., true positives, false negatives, and false positives). However, neither of the objective metrics account for the fact that annotators may adopt different strategies to minimise correction effort, nor do they consider the different operations and relative "costs" involved in correcting annotations.

Bridging User Workflow and Evaluation: The Path to User-Centric Metrics As discussed in the previous section, existing beat-tracking metrics exhibit limitations in two primary areas: algorithm performance assessment and alignment with annotation workflows. Algorithm performance assessment is challenged by the validity of standard classification metrics [Sturm, 2013], emphasising the need for a stronger connection between evaluations and primary musical objectives. Conversely, current metrics fail to capture the nuances of user annotation strategies. This is reflected by the same evaluation score representing distinct correction efforts, and reveal the neglect of the user's role in the evaluation process. Considering these limitations, and to move towards a more comprehensive understanding of the annotation process, it is essential to shift the focus towards user-centric metrics. By placing the user at the core of the evaluation, we can refine the assessment of the annotation workflow and provide a more holistic and effective approach to better understand the interplay between user workflows, musical difficulty, annotator expertise, and algorithmic performance.

To develop user-centric evaluation metrics in beat-tracking, we can draw insights from disciplines like Human-Computer Interaction (HCI), Information Retrieval (IR), and Active Learning (AL), which offer valuable perspectives in terms of user experience

[31] One example of such an adaptation is the evaluation of arpeggiated onsets as established for MIREX 2005 [Dixon, 2006]: in addition to the standard counts of correct detections, false positives, and false negatives, the evaluation considered the number of merged onsets (two onsets detected as a single onset) and double onsets (a single onset recognised as two).

and the search for adequate metrics in different contexts. In the realm of HCI, a variety of insightful metrics exist to evaluate user experience, ranging from task completion time to error rate, learnability, and efficiency, among others [Hornbæk, 2006]. In the case of IR, metrics such as precision, recall, and F1-score[32]are favoured over accuracy for a fairer evaluation of system performance, given the common class imbalance in this domain [Al-Maskari et al., 2007]. Lastly, Active Learning provides metrics like the area under the learning curve (AUC)[Cawley, 2011], which offer an indication of the annotation effort required to achieve a specific performance level [Khoshrou et al., 2015].

As previously discussed (Section 2.3.2), efficiency in annotation is greatly influenced by the interplay between musical complexity, annotator expertise, and the underlying workflow. More complex music demands greater cognitive effort and time investment, while streamlined workflows and increased expertise lead to reduced time and effort for high-quality annotations. Borrowing terminology from other disciplines [Dorf and Bischop, 2011; Rogers et al., 2023], we can consider that the efficiency of annotation is dependent on the user's (adaptable) workflow, given the (fixed) complexity of the music piece being analysed.

In conclusion, we assert the necessity for a more user-centric approach that considers the annotation workflows in its evaluation metrics, thereby providing a better assessment of productivity and efficiency. This untapped potential forms a promising avenue for further contribution, which we will begin to explore in Section 4.3.

2.3.3 Discussion

In this section, we have emphasised the growing importance of user-centred strategies in Music Information Retrieval (MIR) and Machine Learning (ML). Despite considerable advancements, computational rhythm analysis continues to face challenges, as demonstrated in the task of beat tracking, where diverse musical conditions can lead to suboptimal performance even for state-of-the-art methods.

The concurrent evolution of user roles in MIR and ML provides valuable insights into the potential of user-centric approaches for addressing these rhythm analysis challenges. By actively engaging users in the learning process, we can improve model performance and versatility, especially in data-constrained scenarios.

Our thesis asserts that strengthening the user's central role can effectively address these persistent challenges in computational rhythm analysis in diverse musical sce-

[32] The F-measure metric used for beat tracking is an adaptation of this score.

narios. This encompasses both the evaluation, where we have asserted the need for user-centric approaches that consider task subjectivity and annotation workflows, and algorithmic performance, where user involvement can enhance model accuracy and applicability.

Our emphasis on active user participation positions our work within the emerging field of Human-in-the-Loop Machine Learning. By leveraging this approach, we aim to harness cutting-edge methods, tailoring them *in situ* to cater to user needs and the current MIR challenges, with the ultimate goal of enhancing the practical impact of these technologies in challenging musical contexts.

2.4 Summary

This chapter provided an overview of computational musical rhythm analysis in the context of Music Information Retrieval (MIR). We started by delineating the fundamental concepts and terminology associated with musical rhythm. Following this, focusing on beat tracking as the prototypical task, we delved into the current state of the art, discussing critical algorithms, techniques, strengths, limitations, and the inherent challenges in evaluation.

We dedicated a pivotal part of this chapter to examining the evolution of the user's role in MIR and ML. We emphasised the ongoing shift towards human-centred approaches and highlighted the increasing integration of users in the development, evaluation, and performance of data-driven systems. Concentrating on the user's role as an annotator, beat annotation served as our case study. In this context, we addressed the inherent uncertainty involved in the process and deliberated on methods for assessing annotation efficiency.

In conclusion, we positioned our research within the broader MIR landscape, outlining the key approaches and methodologies we propose to employ. Our aim is to develop effective, user-centred solutions for computational rhythm analysis, thereby aligning our work with the field of Human-in-the-Loop Machine Learning.

3

High-Level User Parameterisation in Beat Tracking

In this chapter, we propose an alternative formulation that allows an end-user to drive how the beat tracking is undertaken. Our goal is to enable the user to rapidly arrive at the beat annotation suitable for their purposes with a minimal amount of interaction. Put another way, we envisage an approach to beat tracking where high-level contextual knowledge about a specific musical signal can be given by the user and reliably interpreted by the algorithm, without the need for extensive model training on annotated datasets, as shown in Figure 3.1. In this sense, we set aside the concept of "universal" beat tracking models that aim for equal performance regardless of the musical input signal, in favour of the more realistic goal of identifying different classes of the beat tracking problem, which require different beat tracking strategies. While the end goal of retrieving beat locations may be the same for fast-paced techno music and highly expressive classical guitar recordings, the assumptions about what constitutes the beat, and how this can be extracted from audio signals, are not. Conversely, constraints should not be placed on what musical content can be creatively re-purposed based on the limitations of MIR algorithms.

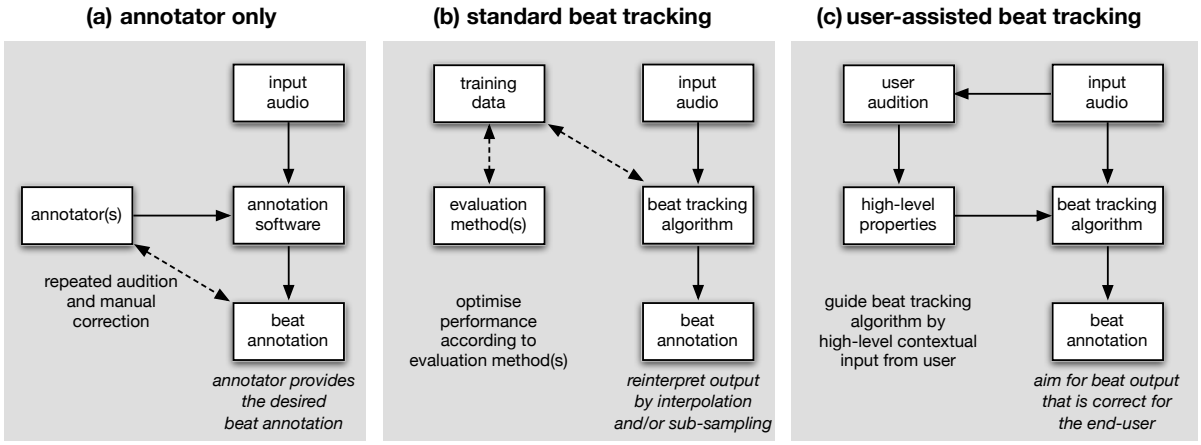


Figure 3.1: Overview of different approaches to obtaining a desired beat annotation. (a) The user annotates the beat positions. (b) A beat tracking algorithm is used — whose performance has been optimised on annotated datasets. (c) Our proposed approach, where user input guides the beat tracking.

The long-term challenges of our approach are as follows: i) determining a low-dimensional parameterisation of the beat tracking space within which diverse, accurate solutions can be found in order to match different beat tracking conditions; ii) exposing these dimensions to end-users in a way that they can be easily understood; iii) providing an interpretable and understandable mapping between the user-input and the resulting beat annotation via the beat tracking algorithm; and finally iv) measuring the level of engagement among end-users who actively participate in the analysis of music signals.

Concerning the dimensions of beat tracking, it is well-understood that music of approximately constant (medium) tempo, with strong percussive content (e.g., pop, rock music) is straightforward to track. Beat tracking difficulty (both for computational approaches and human tappers) can be due to musical reasons and signal-based properties [Grosche et al., 2010; Holzapfel et al., 2012b]. While it is somewhat nonsensical to consider a piece of music with "opposite" properties to the most straightforward case, it has been shown empirically that highly expressive music, without clear percussive content, is not well analysed even by the state of the art in beat tracking [Holzapfel et al., 2012b; Böck et al., 2016b]. Successful tracking of such pieces should, in principle, require input features which can be effective in the absence of percussion and a tracking model which can rapidly adapt to expressive tempo variation. While recent work [Böck et al., 2014] sought to develop multiple beat tracking models, these were separately trained at the level of different databases rather than according to musical beat tracking conditions.

In our approach, we re-examine the functionality of a leading-edge[33] beat-tracking method, i.e., the recurrent neural network approach of Böck et al. [2016b]. In particular, we devise a means to re-parameterise it so that it is adapted for highly expressive music. Based on an analysis of existing annotated datasets, we identify a set of musical stimuli we consider typical of highly challenging conditions, together with a parallel set of "easier" examples. We then conduct a small-scale listening experiment where participants are first asked to rate the perceptual difficulty of tapping the beat, and subsequently to rate the subjective quality of beat annotations given by the expressive parameterisation vs the default version. Our results indicate that listeners are able to distinguish easier from more challenging cases, and furthermore that they preferred the beat tracking output of the expressive-parameterised system to the default parameterisation for the highly expressive musical excerpts. In this sense, we seek to use the assessment of perceptual difficulty of tapping as a means to drive the manner in which the beats can be extracted from audio signals towards the concept of user-informed beat tracking. To complement our analysis, we explore the objective evaluation of the beat tracking model with both parameterisations.

The remainder of this chapter is structured as follows. In Section 3.1 we detail the adaption of the beat tracking followed by the design of a small-scale listening experiment in Section 3.2. This is followed by results and discussion in Section 3.3, and conclusions in Section 4.6.

3.1 Beat Tracking System Adaptation

Our objective is to integrate user input into the music signal analysis process, guiding it with high-level contextual information that is typically easier for human listeners to discern. In straightforward musical cases, a group of leading algorithms for beat tracking, as discussed in section 2.2, has proven to be highly effective. To make a valuable contribution to the field, we concentrate on situations where these algorithms are less effective, especially in cases involving expressive timing. We commence by adopting an RNN-based approach [Böck et al., 2016a], outlining its key functionality before detailing the modifications we introduce.

Originally presented in [Böck and Schedl, 2011], Böck et al. [2016a] utilises deep learning and is readily accessible within the `madmom` library [Böck et al., 2016a]. The crux of the beat tracking model is a recurrent neural network (RNN) trained on a wide

[33] at the time the experiment was conducted, the RNN approach was still the state of the art.

Table 3.1: Overview of default and expressive adapted parameters.

Parameter	Default	Expressive
Minimum Tempo (bpm)	55	35
Maximum Tempo (bpm)	215	135
Transition-λ (unitless)	100	10

variety of annotated beat tracking datasets to predict a beat activation function, which displays peaks at probable beat locations. To generate an output beat sequence, the beat activation function provided by the RNN undergoes post-processing via a dynamic Bayesian network (DBN), approximated by a hidden Markov model (HMM) [Krebs et al., 2015].

While it would be possible to retrain this model from scratch on challenging data, this has been partially addressed in the earlier multimodel approach of Böck et al. [2014]. Instead, we focus on the latter part of the beat tracking pipeline: specifically, how to extract the beat annotation from the beat activation function. To this end, we address three DBN parameters: i) the minimum tempo in beats per minute (bpm); ii) the maximum tempo; and iii) the so-called `transition-`λ parameter which controls the flexibility of the DBN to deviate from a constant tempo[34]. Through iterative experimentation, including both objective evaluation on existing datasets and subjective assessment of the quality of the beat tracking output, we devised a new set of expressiveness-oriented parameters, which are shown, along with the default values in Table 3.1. More specifically, we first undertake a grid search across these three parameters on a subset of musical examples from existing annotated datasets for which the (then) state-of-the-art RNN is deemed to perform poorly, i.e., by having an *information gain* lower than 1.5 bits [Zapata et al., 2012]. An informal subjective assessment was then used to confirm that reliable beat annotations could be obtained from the expressive parameterisation.

As shown in Table 3.1, the main changes for the expressive model are a shift towards a slower range of allowed tempi (following evidence about the greater difficulty of tapping to slower pieces of music [Bååth and Madison, 2012]), together with a lower value for the `transition-`λ. While the global effect of this parameter was studied by Krebs et al. [2015], their goal was to find an optimal value across a wide range of musical examples. Here, our focus is on highly expressive music, therefore we do not

[34] the probability of tempo changes varies exponentially with the negative of the `transition-`λ, thus higher values of this parameter favour constant tempo from one beat to the next one [Krebs et al., 2015].

need to pursue a more general solution. Indeed, the role of the expressive model is to function in precisely the cases where the default approach cannot.

3.2 Methodology

Within this chapter, we posit that high-level user-input can lead to improved beat annotation over using existing state-of-the-art beat-tracking algorithms in a "blind" manner. In order to test this in a rigorous way, we would need to build an interactive beat tracking system including a user interface, and conduct a user study in which users could select their own input material for evaluation. However, doing so would require understanding which high-level properties to expose and how to meaningfully interpret them within the beat tracking system. To the best of our knowledge, no such experiment has yet been conducted, thus in order to gain some initial insight into this problem, we conducted a small-scale online listening experiment, which is split into two parts: **Part A** to assess the perceptual difficulty of tapping the beat, and **Part B** to assess the subjective quality of beat annotations made using the default parameterisation of the RNN beat-tracking system versus our proposed expressive parameterisation.

We use **Part A** as a means to simulate one potential aspect of high-level context which an end-user could provide: in this case, a choice over whether the piece of music is easy or difficult to tap along to (where difficulty is largely driven by the presence of expressive timing). Given this choice, **Part B** is used as the means for the end-user to rate the quality of the beat annotation when the beat tracking system has been parameterised according to their choice. In this sense, if a user rates the piece as "easy", we would provide the default output of the system, and if they rate it as "hard" we provide the annotation from the expressive parameterisation. However, for the purposes of our listening experiment, all experimental conditions are rated by all participants, thus the link between **Part A** and **Part B** is not explicit.

3.2.1 Part A

In the first part of our experiment, we used a set of 8 short music excerpts (each 15 s in duration) which were split equally among two categories: i) "easy" cases with near constant tempo in 4/4 time, with percussive content, and without highly syncopated rhythmic patterns; and ii) "hard" cases typified by the presence of high tempo variation

and minimal use of percussion. The musical excerpts were drawn from existing public and private beat tracking datasets, and all were normalised to −3 dB.

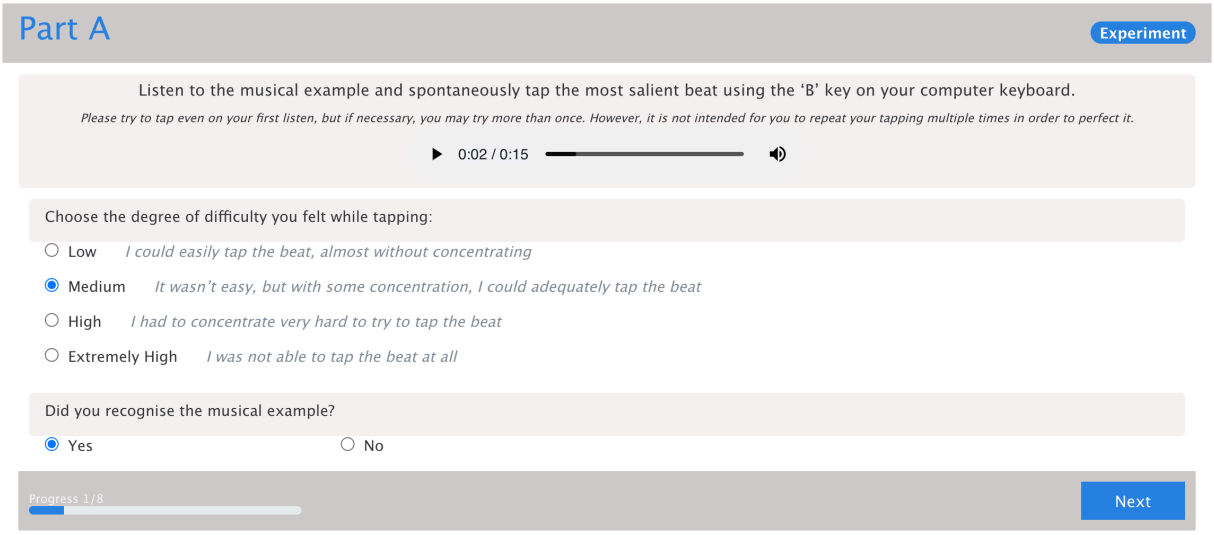


Figure 3.2: Listening Experiment - Graphical Interface of Part A.

We asked the participants to listen to the musical excerpts and to spontaneously tap along using the computer keyboard at what they considered the most salient beat. Due to the challenges of recording precise time stamps without dedicated signal acquisition hardware (e.g., at the very least, a MIDI input device) the tap times of the participants were not recorded, however this was not disclosed. We then asked the participants to rate the difficulty they felt when trying to tap the beat, according to the following four options:

- Low - *I could easily tap the beat, almost without concentrating*
- Medium - *It wasn't easy, but with some concentration, I could adequately tap the beat*
- High - *I had to concentrate very hard to try to tap the beat*
- Extremely high - *I was not able to tap the beat at all.*

Our hypothesis for **Part A** is that participants would consistently rate those drawn from the "easy" set as having Low or Medium difficulty, whereas those from the "hard" should be rated with High or Extremely High difficulty.

3.2.2 Part B

Having completed **Part A**, participants then proceeded to **Part B** in which they were asked to judge the subjective quality of beat annotations (rendered as short 1 kHz

pulses) mixed with the musical excerpts. The same set of musical excerpts from **Part A** were used, but they were annotated in three different ways: i) using the *default* parameterisation of Böck et al. [2016a]; ii) using our proposed *expressive* parameterisation (as in Table 3.1); and iii) a control condition using a completely *deterministic* beat annotation, i.e., beat times at precise 500 ms intervals without any attempt to track the beat of the music. In total, this created a set of $8 \times 3 = 24$ musical excerpts to be rated, for which participants were asked to: *Rate the overall quality of how well the beat sequence corresponds to the beat of the music.*

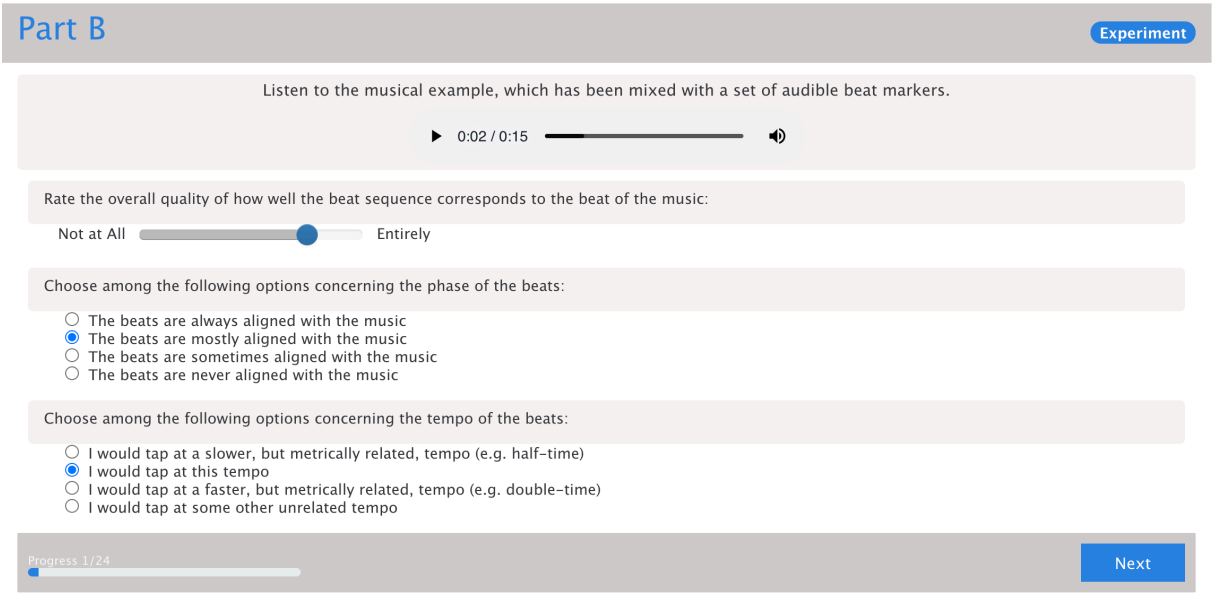


Figure 3.3: Listening Experiment - Graphical Interface of Part B.

For this question, a 5-point Likert-type item was used with (1) on the left-hand side corresponding to "Not at all" and (5) corresponding to "Entirely" on the right-hand side. Our hypothesis for **Part B** was that for the "hard" excerpts, the annotations of the expressively-parameterised beat tracker would be preferred to those of the default approach, and for all musical excerpts that the deterministic condition would be rated the lowest in terms of subjective quality. In this part of the experiment we draw inspiration from evaluation of automatic musical accompaniment driven by real-time beat tracking where our three conditions of: *default*, *expressive*, and *deterministic* can be deemed similar to the use of a beat tracking system, a human tapper, and a quantised beat sequence, by Stowell et al. [2009].

3.2.3 Implementation

The experiment was conducted online within a web browser, with participants recruited from the University of Porto's student body and the wider research network of the Sound and Music Computing Group. All participants gave their informed consent to participate, and the data collected were handled anonymously. They were also asked to provide basic information for statistical purposes. The experiment[35] was designed to ensure a high-quality listening environment and included a compulsory training phase to familiarise participants with the questions. The test took around 30 minutes to complete.

3.3 Results and Discussion

3.3.1 Listening Experiment

A total of 10 listeners (mean age: 31, age range: 23–43) participated in the listening test, 9 of whom self-reported amateur or professional musical proficiency.

For **Part A**, we obtained 40 ratings for each stimulus group "easy" and "hard", according to the frequency distribution shown in Figure 3.4. The most frequent rating for the first group was "low" (82.5%), followed by the "medium" rating (12.5%). For the "hard" group, a symmetrical rating was obtained: the adjacent ratings "medium" and "high" (37.5% each), complemented by the more extreme ratings "low" and "extremely high" (12.5% each). A Mann-Whitney test showed that there was a statistically significant difference between the ratings for both groups, with $p < 0.001$.

These results suggest greater consistency in classifying the "easy" excerpts as having low difficulty, with only two excerpts rated above "medium", than for the "hard" excerpts, which spanned the entire rating scale from low to extremely difficult. The majority of ratings for the "hard" excerpts, however, were either medium or high difficulty. We believe this variability in the difficulty ratings can be attributed to the participants' musical expertise and their familiarity with specific pieces. A distinction was observed between the participants' perceived difficulty in tapping along and the

[35] The experiment was built using `HTML5` and `Node.js`. Participants were required to provide information such as sex, age, their level of expertise as a musician, and experience in music production. They were also informed that they could withdraw from the experiment at any time without penalty, and their partial responses would not be recorded. To prevent order effects, each participant was presented with the musical excerpts in a different random order. Participants were encouraged to take the experiment in a quiet environment using high-quality headphones or loudspeakers and were given the opportunity to set the playback volume to a comfortable level before starting.

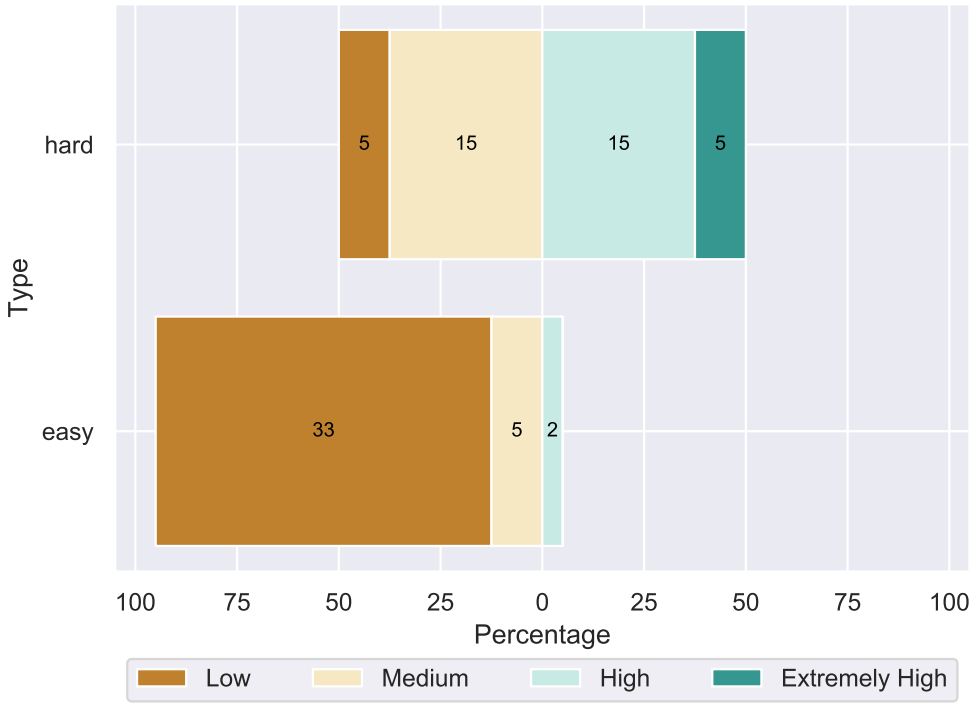


Figure 3.4: Subjective ratings of the difficulty of beat tapping.

presence of expressive timing in the musical excerpts. It appears that experienced listeners had little difficulty tapping along to a piece of expressive music they were familiar with. Therefore, asking expert listeners about the presence of expressive timing might be more effective, while queries about difficulty may be better suited for non-expert listeners who are not conversant with musical terminology.

For **Part B**, the distinction between the ratings of the "easy" and the "hard" excerpts was again evident. A Kruskal-Wallis H test showed a statistically significant difference between the three models (*expressive*, *default* and *deterministic*): $\chi^2(2) = 87.96$, $p < 0.001$ for "easy" excerpts, $\chi^2(2) = 70.71$, $p < 0.001$ for "hard" excerpts. A post-hoc analysis performed with the Dunn test with Bonferroni correction revealed that all differences were statistically significant with $p < 0.001/3$ (except for the pair *default–expressive* under the "easy" stimuli, for which identical ratings were obtained). A descriptive summary of the ratings (*boxplot* with scores overlaid) for each type of stimuli, and under the three beat annotation conditions are shown in Figure 3.5.

The main results from Part B are as follows. For the "easy" excerpts there is no difference in performance for the *default* and *expressive* parameterisations of the beat tracking model, both of which are rated with high scores indicating high quality beat annotations from both systems. We contrast this with the ratings of the *deterministic*

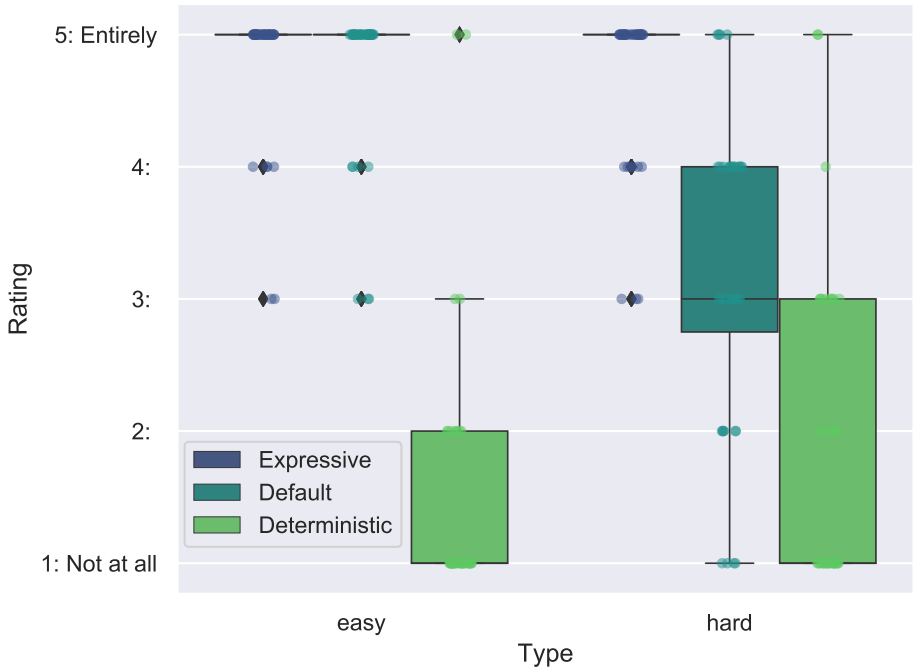


Figure 3.5: Subjective ratings of the quality of the beat annotations.

output (which should bear no meaningful relationship to the music) and which are rated toward the lower end of the scale. From these results, we can infer that the participants were easily able to distinguish between accurate beat annotations and deliberately inaccurate annotations. This result is consistent with the well-known Beat Alignment Test [Iversen and Patel, 2008]. Concerning the ability of the expressively parameterised model to achieve such high ratings, we believe that this was due to very clear information concerning the beat in the beat activation functions from the RNN, and thus there was no alternative "expressive" path for this model to follow.

Conversely, the ratings of the "hard" excerpts show a different picture. Here, the ratings of the expressively-parameterised model are similar to the "easy" excerpts, but the ratings of the *default* model [Böck et al., 2016a] are noticeably lower. This suggests that the participants, in spite of their reported higher perceptual difficulty in tapping the beat, were able to reliably identify the accurate beat predictions of the *expressive* model over those of the *default* model. It is noteworthy that the ratings of the *deterministic* approach are moderately higher for the "hard" excerpts compared to the "easy" excerpts. Given the small number of samples and participants for this experiment, we should not draw strong conclusions about this difference, but for highly expressive pieces, the *deterministic* beats may have inadvertently aligned with the music in brief periods compared to the "easy" excerpts, which may have been

unrelated in a more obvious way to listeners.

3.3.2 Beat Tracking Accuracy

In addition to reporting on the listening experiment whose focus is on subjective ratings of beat tracking, we also examine the difference in objective performance of using the *default* and *expressive* parameterisations of the beat tracking model. Given the focus on challenging excerpts for beat tracking, we initially focus on the *SMC* dataset [Holzapfel et al., 2012b]. It contains 217 excerpts, each of 40 s in duration. Following the evaluation methods described in [Davies and Böck, 2014] we select the following subset: F-measure, CMLc, CMLt, AMLc, and AMLt to assess performance. In Tables 3.2, we show the recorded accuracy on the *SMC* for both the default and expressive parameterisations. Note, for the default model we use the version in the `madmom` library [Böck et al., 2016a] which has been exposed to this material during training (via cross-validation), hence the accuracy scores are slightly higher than those in [Böck et al., 2016b] where cross fold validation was used. In addition to showing the performance of each parameterisation we also show the theoretical upper limit achievable by making a perfect choice (by a hypothetical end-user) among the two parameterisations. Since multiple evaluation scores are reported, and there is no accepted single metric to use within the beat tracking community, we make the optimal choice per excerpt according to each individual evaluation metric.

Table 3.2: Overview of beat tracking performance on the *SMC* dataset [Holzapfel et al., 2012b] comparing the default and expressive parameters together with upper limit on performance.

	F-measure	**CMLc**	**CMLt**	**AMLc**	**AMLt**
Default [Böck et al., 2016a]	0.563	0.350	0.472	0.459	0.629
Expressive	0.540	0.306	0.410	0.427	0.565
Optimal Choice	**0.624**	**0.456**	**0.611**	**0.545**	**0.703**

From Table 3.2, we see that the default parameterisation outperforms the expressive one for all the evaluation methods. This result is not unexpected – as the *SMC* dataset is not entirely composed of highly expressive musical material. We consider the more important result to be the potential for our *expressive* parameterisation to track those excerpts for which the *default* approach fails. To this end, the increase of approximately 10% points across each of the evaluation methods demonstrates how these two different parameterisations can provide greater coverage of the dataset.

While the *SMC* dataset is well-known for containing a high proportion of challenging material, we also believe that it is worthwhile to explore the effectiveness of our method on other musical material. Since the expressive parameterisation should only be effective when applied to music with a slow average tempo and high expression, the gains on datasets composed primarily of pop or rock music will be much lower. In addition, many of the existing beat tracking datasets have been used to train the approach of Böck et al. [2016b] and thus cannot provide insight into the effectiveness of our approach on truly unseen data. To this end, we make use of a more recently annotated dataset which was used in the 2017 IEEE Signal Processing Cup (*SP Cup*) [Jin et al., 2017]. While the dataset is quite small, containing 98 excerpts of 30 s it was compiled in a community-driven fashion where teams participating in the competition selected the audio material and annotated it themselves. In line with the competitive element of the *SP Cup* many teams chose to submit challenging musical excerpts. On this basis, we believe it represents a highly appropriate choice for additional validation of our approach. A summary of the results containing the same three conditions: default, expressive, and the optimal choice between the two, is shown in Table 3.3.

Table 3.3: Overview of beat tracking performance on the *SP Cup* dataset [Jin et al., 2017] comparing the default and expressive parameters together with upper limit on performance.

	F-measure	**CMLc**	**CMLt**	**AMLc**	**AMLt**
Default [Böck et al., 2016a]	0.833	0.660	0.687	0.846	0.877
Expressive	0.783	0.564	0.581	0.805	0.826
Optimal Choice	**0.860**	**0.733**	**0.762**	**0.873**	**0.897**

Contrasting the results in Tables 3.2 and 3.3, we can observe a similar pattern of lower overall performance for the expressive approach compared to the default parameterisation. Depending on the evaluation method, however, once again, the optimal choice between the two provides a notable improvement (of up to 7% points). Given the improvement under both presented datasets we believe this supports the need for different parameterisations to tackle different types of musical content, a concept related to Collins' discussion of "style-specific" beat tracking [Collins, 2006]. In addition, it suggests that training a classifier to choose between expressive and non-expressive pieces would be a promising area for future work.

3.3.3 Individual Example

While results shown in Tables 3.2 and 3.3 focus more on the global effect of these different parameterisations across entire datasets, it is important to consider the practical impact at the level of individual musical excerpts. In this section we consider an annotation workflow perspective, which might rely on the correction of an automatic annotation of the beat of a piece of music, as opposed to completely annotating a piece by hand. In this context, we contend that an informed choice of how to first estimate the beat automatically may have a significant impact in terms of the subsequent work required to obtain an output which is acceptable for the end-user, i.e. by inserting, deleting, and shifting the automatically estimated beats.

To this end, we focus on one specific example within the Hainsworth dataset [Hainsworth and Macleod, 2004]; an excerpt from the composition "Evocaciòn" by Jose Luis Merlin. It is a solo piece for classical guitar which features extensive *rubato* and as such can be considered one of the more challenging pieces within the dataset. In the absence of any other musical instruments, together with the clear guitar plucking technique, this piece is rather a paradox since it is quite straightforward for onset detection, but notoriously difficult for beat tracking. The challenge lies not in the ability to precisely identify where in time the notes are played, but to decode which of these onsets correspond to the beat over a highly variable underlying tempo. To explore this specific musical excerpt in greater detail, we contrast the outputs of the default and expressive parameterisations together with the ground-truth annotation in Figure 3.6 (taken from the supplementary material from [Böck et al., 2019]).

As can be seen from the figure, the output of the expressive parameterisation (in the bottom plot) is much closer to the ground-truth annotations than the default (in the top plot). Across this 30 s section, the expressive output requires just 6 beats in need of correction, while the default output requires no fewer than 18. The number of atomic operations to correct each annotation can be broken down as follows: 13 shifts and 6 deletions for the default output *vs.* 3 shifts and 3 deletions for the expressive output. Taking into account the number of annotations in this excerpt, the amount of editing effort required to converge on the ground-truth annotation is even more illustrative: 21% of the expressive beats output vs 61% of the default beat outputs. Thus, from the perspective of the user (annotator), it is clearly more efficient to correct the expressive output.

In this example, we have explicitly used the ground truth as a means to illustrate the fewer errors made by the expressive parameterisation. However, when such ground-

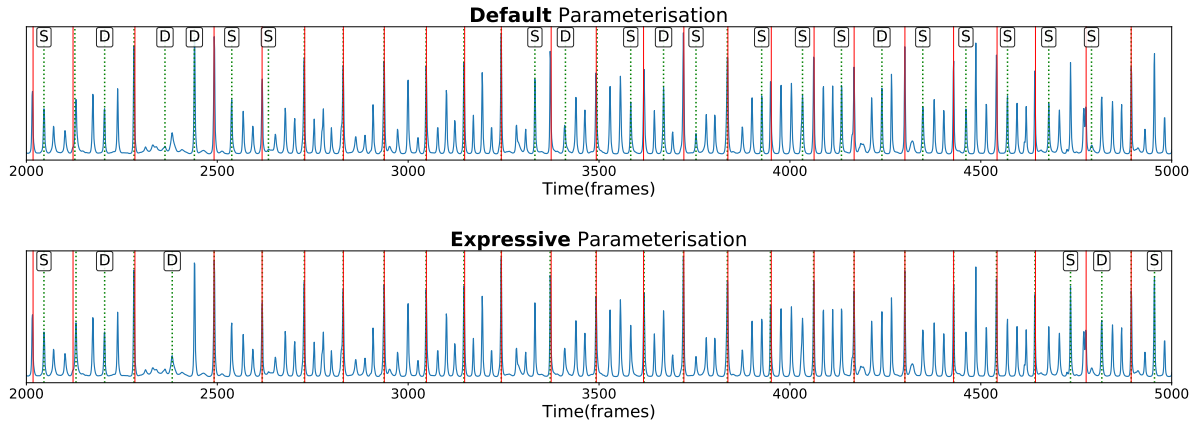


Figure 3.6: Comparison of different beat tracking outputs. The blue solid line indicates the beat activation function given by the Böck et al. approach [Böck et al., 2016b]. The vertical red solid lines show the ground-truth annotations. The vertical green dashed lines show: the default output (top) and the expressive output (bottom). The incorrect beat outputs are labelled with the required operations (**D**elete, **S**hift, **I**nsert) to correct the annotation. The temporal axis represents frames at a rate of 100 frames per second.

truth annotations exist, the need for automatic analysis is negated. Yet, in real-world uses, where there is no ground truth, we would replace this visual comparison with an interactive process whereby the user verifies the output of the algorithm by listening and iterative adjustment. The number of edit operations required to achieve the desired output indicates the extent of interaction between the user and the beat-tracking system. This can provide a direct measure of the impact of user-informed beat tracking in the annotation workflow.

3.4 Summary

In this chapter, we have initiated a discussion on the potential role of user input in guiding MIR analysis. Within the context of beat tracking, we have demonstrated the possibility of reparameterising an existing top-performing approach to yield improved beat annotations for highly expressive music. Furthermore, we have demonstrated that the option to choose between default and expressive parameterisations can significantly enhance the analysis of challenging beat tracking material. It is noteworthy that the benefits of the expressive model were achieved without retraining the RNN architecture, but rather through reparameterisation of the DBN tracking model, which performs inference on the RNN's predictions.

To explore how user input could be utilised for beat tracking, we simulated a scenario where user decisions about the perceptual difficulty of tapping were translated

into the use of a parameterisation for expressive musical excerpts. We speculate that listener expertise and familiarity may contribute to reducing the perceived difficulty of otherwise challenging expressive pieces. Our aim is to delve deeper into the parameters that can be exposed to end-users and to ascertain whether distinct properties exist for expert and non-expert users. While our results are statistically significant, we acknowledge the small-scale nature of the listening experiment. Ultimately, this research sets the stage for a more user-centred approach in MIR, in which the incorporation of listener feedback can enhance the performance of analysis systems.

4

User-Informed Finetuning for Improved Beat Tracking

The aim and motivation for this chapter are to shift away from the notion of targeting and reporting high (mean) accuracy across existing annotated datasets, and instead to move towards the real-world use of beat tracking systems by end-users on specific musical pieces. More specifically, we investigate what to do when even the state of the art is not effective and very high accuracy is required, i.e., when the extraction of the beat is used to drive higher-level musicological analysis or creative musical repurposing.

Faced with this situation, currently available paths of action include: (i) the end-user performing manual corrections to the beat output or even resorting to a complete re-annotation by hand, which may be extremely time-consuming and labour-intensive; (ii) the use of some high-level parameterisation of the algorithm in terms of an expected tempo range and initial phase [Dixon, 2001a; Dalton et al., 2019]; or (iii) adapting

some more abstract parameters that could permit greater flexibility in tracking tempo variation, as addressed in Chapter 3. While our approach has shown encouraging results, there are limitations to consider when dealing with varying signal properties (e.g. timbre). In such cases, user-provided information may have localised utility if the model is unable to make reliable beat-structure predictions. Similarly, when handling highly expressive musical content, static properties like the initial tempo input may become less relevant as the music progresses, thus limiting the usefulness of this high-level information.

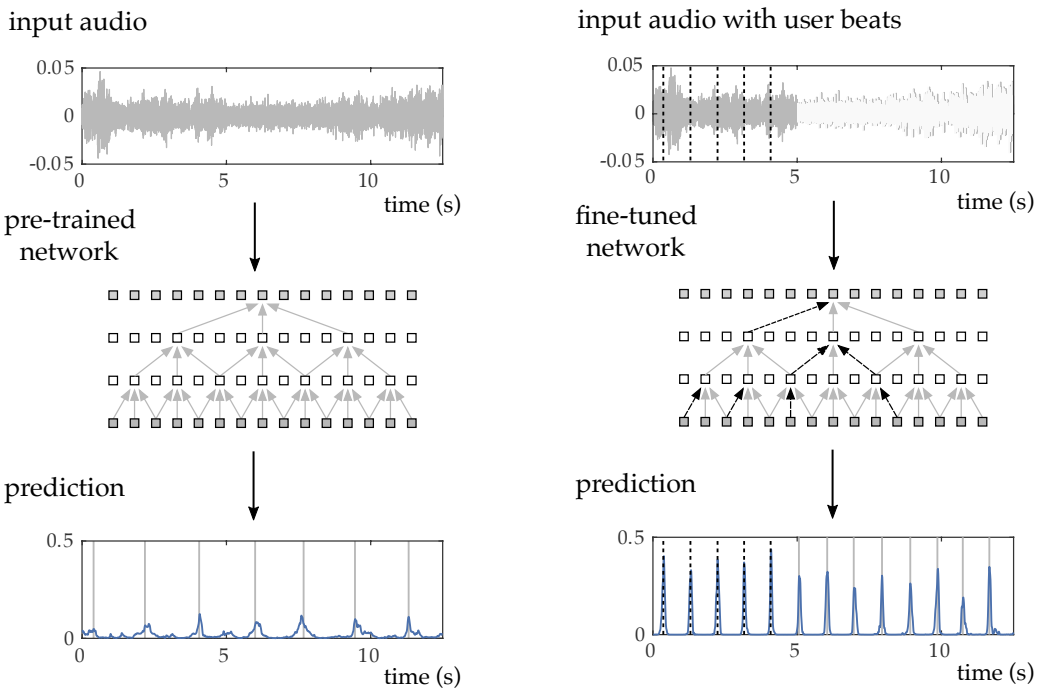


Figure 4.1: Overview of our proposed approach. The left column shows an audio input passed through a deep neural network (for consistency with our approach, this is a temporal convolutional network), which produces a weak beat activation function and erroneous beat output. The right column shows the same audio input, but here, a few beat annotations are provided as the means to finetune the network—with the black arrows implying the modification of some of the weights of the network. This results in a much clearer beat activation function and an accurate beat-tracking output.

In light of these limitations, we propose a method in which a very limited amount of manual annotation by a hypothetical end-user is used to finetune an existing state-of-the-art system [Böck and Davies, 2020] to adapt it to the specific properties of the musical piece being analysed. In essence, we aim to leverage the general musical knowledge of a beat-tracking system exposed to a large amount of training data and then recalibrate the weights of the network so that it can rapidly learn how to track

the remainder of the given piece of music with a high degree of accuracy. A high-level overview of this concept is illustrated in Figure 4.1. However, for this method to be practically applicable, it is crucial that the finetuning process is computationally efficient and does not require specialised hardware; it should be possible to complete the finetuning within seconds on a standard personal computer.

To demonstrate the validity of our approach, we show the improvement over the current state of the art offered by our finetuning approach on existing datasets and specific examples, demonstrating that our approach can learn what the beat is, and also what is not the beat. Additionally, we investigate the trade-off between learning the specific properties of a given piece and forgetting more general information. In summary, the main contributions of this work are: (i) to reformulate the beat-tracking problem to target high accuracy in individual challenging pieces where the current state-of-the-art is ineffective; (ii) to introduce the use of *in situ* finetuning over a small annotated region as a straightforward means to adapt a state-of-the-art beat-tracking system so that it is more effective for this type of content; and (iii) to conduct a detailed beat-tracking evaluation from an annotation-correction perspective, which demonstrates and quantifies the set of steps required to transform an initial estimate of the beat into a highly accurate output.

The remainder of this chapter is structured as follows: In Section 4.1, we provide a high-level overview of the state-of-the-art beat-tracking system used as the basis for our approach and then detail our finetuning approach in Section 4.2. In Section 4.3 we present a novel evaluation method based on the annotation-correction workflow, to address current objective metrics' limitations and support the subsequent evaluation. In Section 4.4 we detail the methodology of our experiments, and in Section 4.5, we present the results of our experiments, focusing on our approach evaluation on a set of benchmark datasets, investigate the impact of finetuning in two specific highly challenging musical pieces, and conclude with an examination of catastrophic forgetting effects within our approach. Finally, in Section 4.6, we discuss the limitations of this approach and briefly comment on its implications.

4.1 Baseline Beat-Tracking Approach

A key motivating factor and contribution of this work is to look beyond what is possible with the current state of the art in beat tracking, and hence to explore finetuning as a means for content-specific adaptation. To this end, we restrict the scope of this work

to an explicit extension of the most recent state-of-the-art approach [Böck and Davies, 2020], and thus use this as a baseline on which to measure improvement.

The baseline approach uses multi-task learning for the simultaneous estimation of beat, downbeat and tempo. The core of the approach is a temporal convolutional network (TCN), which was first used for beat tracking only in [Davies and Böck, 2019], and then expanded to predict both tempo and beat [Böck et al., 2019]. Compared to previous recurrent architectures for beat tracking (e.g., [Böck et al., 2016b]), TCNs have the advantage that they retain the high parallelisation property of convolutional neural networks (CNNs), and therefore can be trained more efficiently over large training data [Davies and Böck, 2019]. With the long-term goal of integrating *in situ* finetuning within a user-based workflow for a given piece of music, we considered the efficiency aspect to be particularly important, and this formed a secondary motivation to extend the TCN-based approach.

Before discussing fine-tuning, we provide a high-level overview of this approach to ensure this chapter is largely self-contained. We now summarise the main aspects of the processing pipeline, network architecture and training procedure. For complete details, refer to Böck and Davies [2020].

Pre-processing: Given a mono audio input signal, sampled at 44.1 kHz, the input representation is a log magnitude spectrogram obtained with a *Hann* window of 46.4 ms (2048 samples) and a hop length of 10 ms. Subsequently, a logarithmic grouping of frequency bins with 12 bands per octave gives a total of 81 frequency bands from 30 Hz up to 17 kHz.

Neural network: The neural network comprises two stages: a set of three convolutional and max pooling layers followed by a TCN block. The goal of the convolutional and max pooling layers was to learn a compact intermediate representation from the musical audio signal, which could then be passed to the TCN as the main sequence learning model. The shapes of the three convolutional and max pooling layers were as follows: (i) 3×3 followed by 1×3 max pooling; (ii) 1×10 followed by 1×3 max pooling; and (iii) 3×3 again with 1×3 max pooling. A dropout rate of 0.15 was used with the exponential linear unit (ELU) as the activation function.

This compact intermediate representation was then fed into a TCN block that operated non-causally (i.e., with dilations spanning both forwards and backwards in time). The TCN block was composed of two sets of geometrically spaced dilated convolutions over eleven layers with one-dimensional filters of size five. The first of the dilations spanned the range of 2^0 up to 2^{10} frames and the second at twice this rate. The feature maps of the two dilated convolutions were concatenated before

spatial dropout (with a rate of 0.15) and the ELU as activation function. Finally, in order to keep the output dimensionality of the TCN layer consistent, these feature maps were combined with a 1×1 convolution. Within the multi-task approach (and unlike the simultaneous estimation in [Böck et al., 2016b]), the beat and downbeat targets were separate, each produced by a sigmoid on a fully connected layer. The tempo classification output was produced by a softmax layer. In total, twenty filters were learned within this network, giving approximately 116 k weights. A graphical overview of the network is given in Figure 4.2.

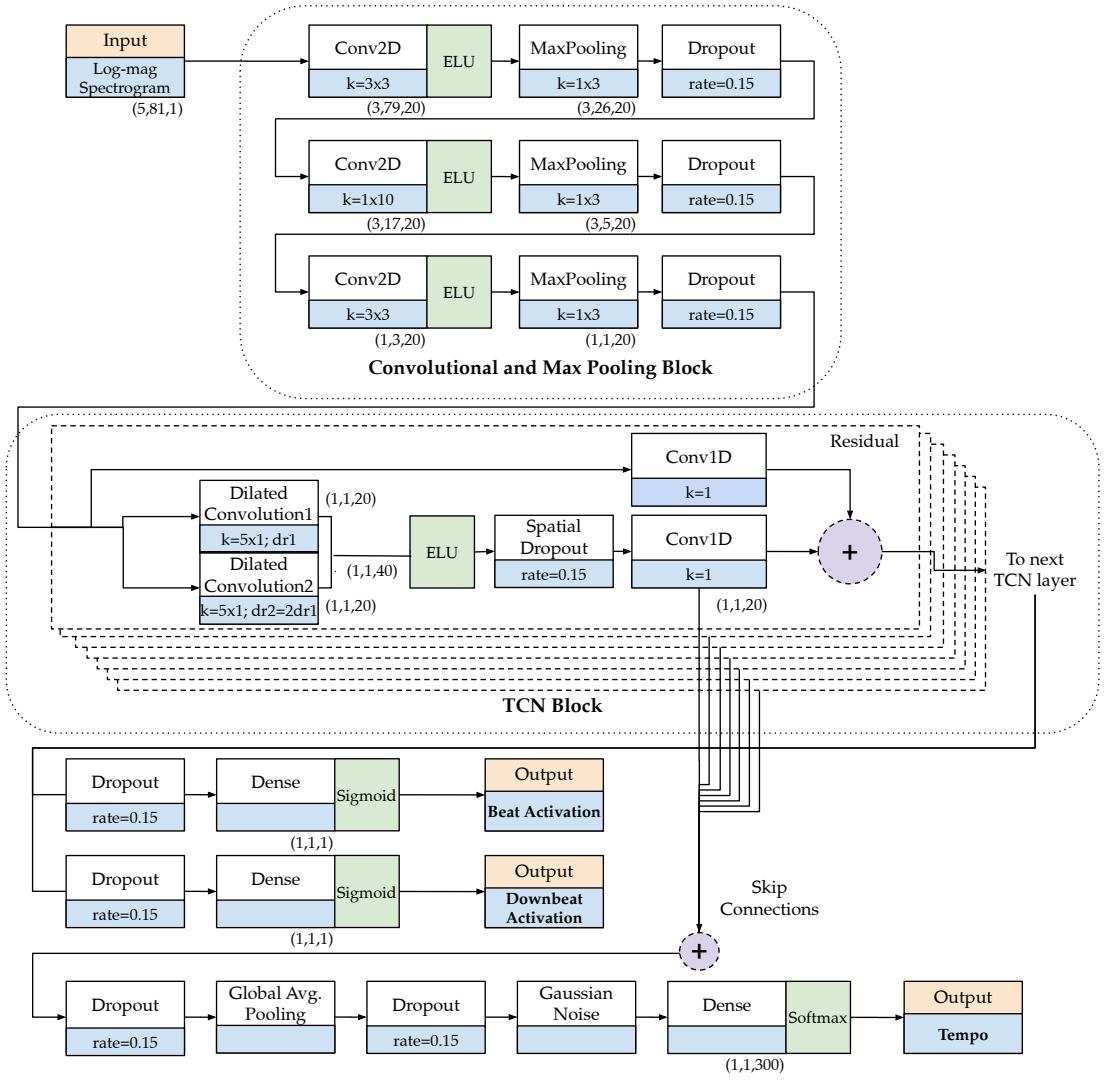


Figure 4.2: Overview diagram of the architecture of the baseline beat-tracking approach.

Regarding the base training, six reference datasets were used, totalling more than 26 hours of musical material: *Ballroom* [Gouyon et al., 2006; Krebs et al., 2013], *Beatles* [Davies et al., 2009], *Hainsworth* [Hainsworth, 2004; Böck et al., 2019], *HJDB* [Hock-

man et al., 2008; Böck et al., 2016b], *Simac* [Gouyon, 2005] and *SMC* [Holzapfel et al., 2012b]. In order to account for gaps in the distribution of the tempi of these datasets, a data augmentation strategy was adopted, by which the training data were enlarged by a factor of 10, by varying the overlap rate of the frames of the Short-Time Fourier Transform (STFT), hence the tempo, and by sampling from a normal distribution with the 5% standard deviation around the annotated tempo and updating the beat, downbeat and tempo targets accordingly. Furthermore, to account for the high imbalance between positive and negative examples (i.e., that frames labelled as beats occurred much less often than non-beat frames), the beat and downbeat targets were widened by ± 2 frames and weighted by 0.5 and 0.25 as they diverged from the central beat frame.

The training was conducted using eight-fold cross validation (6 folds for training, 1 fold for validation, and 1 fold held-back for testing), with excerpts from each dataset uniformly distributed across the folds. A maximum of 200 training epochs per fold were used with a learning rate of 0.002, which was halved after no improvement in the validation loss for 20 epochs, and early stopping was activated with no improvement after 30 epochs. The `RAdam` optimiser followed by *lookahead optimisation* were used with a batch size of one and gradient clipping at a norm of 0.5.

Post-processing: To obtain the final output, the beat activation and downbeat activations were combined and passed as the input to a dynamic Bayesian network approximated via an HMM [Böck et al., 2016b], which simultaneously decoded the beat times and labels corresponding to metrical position. However, given only the beat activation function, it was possible to use the beat-only HMM for inference [Krebs et al., 2015].

4.2 Finetuning

Shifting our focus from the network architecture previously described, we now explore how to adapt it to successfully analyse highly challenging musical pieces. We are particularly interested in musical content where the current state-of-the-art approach falls short, and high accuracy is desired by end-users. In this context, some form of user input could be beneficial to guide beat estimation.

Our strategy broadly involves leveraging the transferability of features in neural networks [Yosinski et al., 2014], effectively utilising the global knowledge about beat tracking from the baseline approach and its training datasets. We then recalibrate it

to fit the musical properties of a specific new piece. By connecting this transferability concept with an end-user who actively participates in the analysis and a prototypical beat annotation workflow, we formulate the network adaptation as a process of finetuning based on a small temporal region of manually annotated beat positions. From the user's perspective, this involves a minimal annotation effort to mark a few beats by hand and then using this information to update the weights of the baseline network, enabling accurate analysis of the complete piece with minimal additional user interaction.

In this chapter, our primary interest is understanding the viability of this approach rather than testing it in real-world conditions. To that end, we simulate the annotation effort of the end-user by using ground-truth annotations over a small temporal region and examining how well the adapted network can track the rest of the piece. Technically, we begin with a pretrained model from the baseline approach described earlier. Then, for a given musical excerpt (unseen by the pretrained model), we isolate a small temporal region (nominally near the start of the excerpt), which we set to be 10 seconds in duration, and retrieve the corresponding ground-truth beat annotations. These three components form the basis of our finetuning approach, as illustrated in Figure 4.1. We focus on: (i) how to parameterise the finetuning; (ii) when to stop the finetuning; and (iii) how to cope with the very limited amount of new information provided by the small temporal region.

Finetuning parameterisation: The first consideration in our finetuning approach is examining which layers of the baseline network to update. While it is common in transfer learning to freeze all but the last layers of the network [Howard and Ruder, 2018], in our context, one important means for adapting the network resides in modelling how the beat is conveyed within the log magnitude spectrogram itself (i.e., unfamiliar musical timbres such as the human voice). To this end, we allow all the layers of the network to be updated by the finetuning process. Since our focus in this chapter is restricted to beat tracking, we mask the losses for the tempo and downbeat tasks. From a practical perspective, this also means that we do not require downbeat or tempo annotations across the 10-second temporal region. Concerning the parameterisation of the finetuning, we follow common practice in transfer learning and reduce the learning rate, setting it to 0.0004 (i.e., one fifth of the rate used in the baseline).

Stopping criteria: The next area is to address when to stop finetuning. In more standard approaches for training deep neural networks, e.g., our baseline approach, cross-fold validation is used with the validation loss driving the adjustment of the

learning rate and the execution of early stopping. In our approach, if we were to use the entire 10 s region for training, then it would be difficult to exercise control over the extent of the network adaptation. Using a small, fixed number of epochs might leave the network essentially unchanged after finetuning, and by contrast, allowing a large number of epochs might cause the network to overfit in an adverse manner. Furthermore, the hypothetically optimal number of epochs is likely to vary based on the musical content being analysed. Faced with this situation, we elect to split the 10-second region into two adjacent, disjoint, 5 s regions, using one for training and the other for validation. In this way, we create a validation loss that we can monitor, but at the expense of reducing the amount of information available for updating the weights. We set the maximum number of epochs to fifty and reduce the learning rate by a factor of two when there is no improvement in the validation loss for at least five epochs, and we stop training when the validation loss plateaus for five epochs.

Learning from very small data: The final area for consideration in our approach relates to strategies to contend with the very limited amount of information in the 5 s temporal region used for training, which may amount to as few as 10 annotated beat targets. Given our interest in challenging musical content (which is typically more difficult to annotate [Holzapfel et al., 2012b]), we should consider the fact that these observable annotations may be poorly localised, and furthermore that the tempo may vary throughout the piece in question. To help contend with poor localisation, we use a broader target widening strategy than the baseline approach, expanding to three adjacent frames on either side of each beat location, with decreasing weights of 0.5, 0.25, and 0.125, from the closest to the farthest frame. On the issue of tempo variability, we reuse the same data augmentation from the baseline approach: altering the frame overlap rate by sampling from a normal distribution with a 5% standard deviation from the local tempo (calculated by means of the median inter-beat interval across the annotated region).

In summary, when considering each of these steps, we believe that our finetuning formulation is quite general and, as such, the same approach could be based upon any other DNN network design. As our departing model (baseline), we chose a state-of-the-art approach [Böck and Davies, 2020], a multi-task architecture for beat, downbeat, and tempo tracking, but used in a single-task (beat tracking) setting. As such, we mask both the tempo and downbeat targets, thus using solely the beat loss in backpropagation to update the network weights.

In a real-world scenario, the end-user chooses the annotated snippet of interest. The choice of this annotated region will bear great relevance to the final performance,

as it defines the signal characteristics to which the network is being finetuned: if the annotated region is well-represented in the rest of the music, or if this region contains very difficult-to-detect beat-tracking events, the adapted network performance will be better than the baseline. In the current work, we adopt a fixed region for the snippet of interest, the first 10 seconds after the first beat annotation position, divided into two equal disjoint sets: the first for validation and the second for training (finetuning). We apply the same data-augmentation as in [Böck and Davies, 2020], by changing the rate at which overlapping frames of the STFT are sampled from a normal distribution with a 5% standard deviation from the annotated tempo. The network is trained for a maximum of 50 epochs with a reduced learning rate of 0.0004, one-fifth of the original learning rate [Böck and Davies, 2020]. We reduce this rate by a factor of 2 when there is no improvement on the validation set for at least 5 epochs, and stop training when the validation loss plateaus for 5 epochs.

To isolate the impact of different techniques on our finetuning approach, we also test a scenario without data augmentation, where the network is trained for a maximum of 200 epochs with the same initial learning rate. Likewise, we reduce this rate by a factor of 2 when there is no improvement on the validation set for 20 epochs, the validation loss reaches a plateau, and training is stopped if no improvement in the validation loss is observed for 30 epochs. The rest of the parameterisation is kept the same as in the original work [Böck and Davies, 2020]. Also, in the present chapter, we directly reuse the default parameters given in [Davies and Böck, 2019]: a tempo range of 55–215 beats per minute, and the `transition-`λ (which controls the model ability to react to tempo changes) at a value of 100.

In conclusion, our approach to finetuning aims to adapt an existing neural network model for beat tracking to better handle challenging musical pieces. By considering aspects such as finetuning parameterisation, stopping criteria, and learning from very small data, we propose a general framework that can be applied to different deep neural network designs. Our experiments demonstrate that incorporating user input, even in a limited fashion, can significantly improve the network's performance on difficult-to-analyse music.

4.3 User Workflow-Based Evaluation

In this section, we propose a new approach for the evaluation of beat tracking, framing the process from the perspective of a user's workflow. The problem is posed in

terms of the effort needed to transform a sequence of beat detections to maximise the well-known F-measure calculation when compared to a sequence of ground-truth annotations. By viewing the evaluation through a transformation lens, we implicitly adopt the widely accepted definition of similarity between two objects (i.e., the beat annotations and the beat detections) in the field of information retrieval [Li et al., 2004], ultimately addressing the question: *How difficult is it to transform one into the other?* With the aim of enhancing the qualitative understanding of beat-tracking results, we have developed an informative visualisation (shown in Figure 4.3) focused on the operations required to correct the algorithmic output, which we now discuss.

In musical audio analysis, the manual alteration of automatically detected time-precise musical events such as onsets [Valero-Mas and Iñesta, 2017] or beats [Driedger et al., 2019] is an onerous process. In the case of musical beat tracking, the beat detections may be challenging due to the underlying difficulty of the musical material, but the correction process can be achieved using two simple editing operations: insertions and deletions — combined with repeated listening to audible clicks mixed with the input. The number of insertions and deletions correspond to counts of *false negatives* and *false positives*, respectively, and form part of the calculation of the F-measure. While this is routinely used in beat tracking (and many other MIR tasks) to measure accuracy, we can also view it in terms of the effort required to transform an initial set of beat detections to a final desired result (e.g., a ground-truth annotation sequence). In this way, a high F-measure would imply low effort in manual correction and vice versa.

In practice, correcting beat detections often relies on a third operation: the *shifting* of poorly localised individual beats. This shifting operation is particularly relevant when correcting tapped beats, which can be subject to human motor noise (i.e., random disturbances of signals in the nervous system that affect motor behaviour [Faisal et al., 2008]), as well as jitter and latency during acquisition. Under the logic of the F-measure calculation, shifting beat detections that fall outside tolerance windows are effectively counted twice: as a false positive *and* a false negative. We argue that for beat tracking evaluation, this creates a modest, but important, disconnect between common practice in annotation correction and a widely used evaluation method. On this basis, we argue that the single operation of shifting should be prioritised over a deletion followed by an insertion.

To account for this operation and to better reflect the annotation workflow in beat tracking evaluation, we introduce a novel metric, the *E-measure*. This metric is a variant of the F-measure (conceived as an *edit–F-measure* and notated as E_m), that departs from Dixon's accuracy [Dixon, 2001b] and reformulates it in terms of the edit operations -

deletions, insertions, and *shifts* - that are typically involved in the annotation process. It is defined as follows:

$$E_m = \frac{t^+}{t^+ + shf + f^+ + f^-} = \frac{det}{det + shf + del + ins} \tag{4.1}$$

where *det* represents the number of correct detections, *del* the count of deletions, *ins* indicates the number of insertions, and *shf* the number of shift operations. This metric outputs a continuous score in the range $[0, 1]$.

Furthermore, it is important to note that the reduction of the inner tolerance window transforms true positives into shifts and thus sends *det* and hence E-measure to zero. In the limit, the modified detections are then identical to the target sequence.

We now specify the main steps in the calculation of the transformation operations:

1. Around each ground-truth annotation, we create an *inner tolerance window* (set to ± 70 ms) and count the number of correct detections (i.e., true positives) – *det*;
2. We mark each matching detection and annotation pair as "accounted for" and remove them from further analysis. All remaining detections then become candidates for shifting or deletion;
3. For each remaining annotation:
   (a) We look for the closest "unaccounted for" detection within an *outer* tolerance window (set to ± 1 s), which we use to reflect a localised working area for manual correction;
   (b) If any such detection exists, we mark it as a shift along with the required temporal correction offset;
4. After the analysis of all "unaccounted for" annotations is complete, we count the number of shifts – *shf*;
5. Any remaining annotations correspond to insertions (i.e., false negatives) – *ins* –, with leftover detections marked for deletion (i.e., false positives) – *del*.

In this method, all operations bear equal weight. However, this is an abstract approximation as, practically, the cost of each operation depends on the user's annotation workflow and the software tool used. For example, certain software may allow the annotation of evenly spaced events using only the initial beat position, tempo (in bpm),

and duration, whereas other tools may require individual annotation for each beat event. Nevertheless, this initial metric proposal, which aligns more closely with the user's workflow, represents a progressive step towards a user-centric evaluation of beat tracking, with further refinements reserved for future exploration.

To allow for metrical ambiguity in beat tracking evaluation, it is common to create a set of variations of the ground truth by interpolation and subsampling operations. We have reversed this practice in our approach, creating variations in the detections instead. Consequently, we pair a global operation applied to all detections (for instance, interpolating all detections by a factor of two) with subsequent local correction operations. The variation yielding the highest E-measure represents the shortest path to an output consistent with the annotations.

The fundamental difference of our approach compared to the standard F-measure is that we view the evaluation from a user workflow perspective, and essentially, *we shift if we can*. By recording each individual operation, we can count them for evaluation purposes, as well as visualising them, as shown in Figure 4.3, which contrasts the use of the original beat detections compared to the double variation of the beats. The example shown is from the composition *Evocación* by Jose Luis Merlin. It is a solo piece for classical guitar, which features extensive *rubato* and is among the more challenging pieces in the *Hainsworth* dataset [Hainsworth, 2004]. By inspection, we can see the original detections are much closer to the ground truth than the offbeat or double variation. They require just 2 shifts and 1 insertion, compared with 12 shifts, 3 insertions and 1 deletion for the offbeat variation (without any valid detection), and 3 shifts and 12 deletions for the double variation, corresponding to very different E-measure scores on the analysed excerpt: 0.8, 0.0 and ≈ 0.44, respectively.

This precise recording of the individual operations enables a more nuanced evaluation, pinpointing which operations are most beneficial and in which order. While shifts are usually more advantageous than isolated insertions or deletions for the F-measure, the temporal location of the operation may be more vital for continuity-based metrics. Combining this transformation perspective with visualisation, we believe our implementation may enhance the qualitative understanding of beat-tracking algorithms.

While the E-measure offers a comprehensive insight into the corrective edit operations inherent to the annotation workflow, it does not encompass the overarching efficiency with respect to the user annotation effort during the finetuning process.

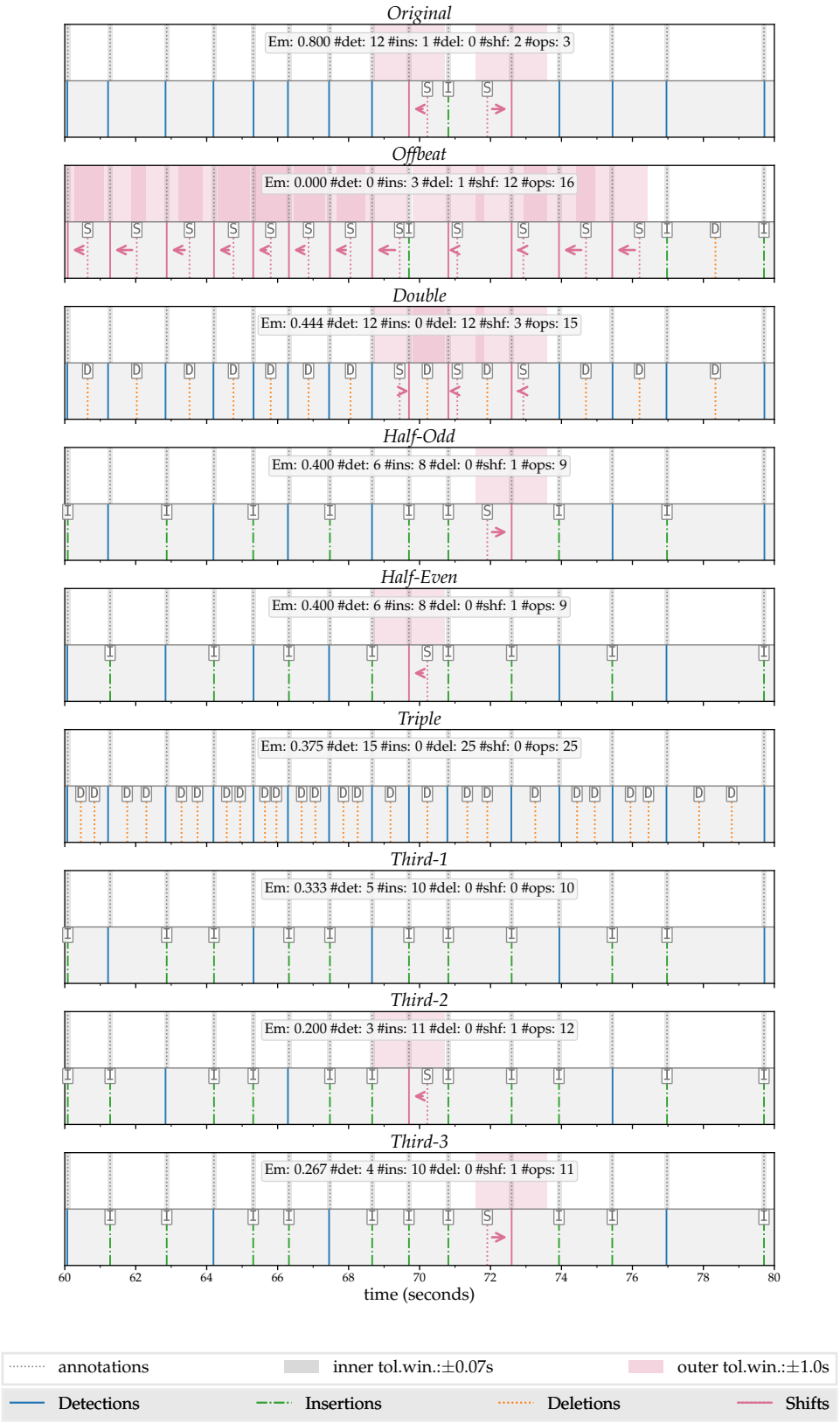


Figure 4.3: Visualisation of the operations required to transform beat detections (to optimise the E-measure when compared to the ground-truth annotations) for the period from 60–80 s, of *Evocaciòn*. (From Top to Bottom) *Original* beat detections; *Offbeat*: 180°out of phase from the original beat locations; *Double*: beats at two times the original tempo; *Half-Odd*, *Half-Even*: half tempo, centred at odd or even beats; *Triple*: beats at three times the original tempo; *Third-1*, *Third-2*, *Third-3*: one-third tempo centred at beat 1, 2, or 3. The inner tolerance window is overlaid on all annotations, whereas the outer tolerance window is only shown for those detections to be shifted.

To address this gap, we propose a novel metric tailored to evaluate the relative performance of a finetuned beat tracking algorithm, in relation to the user-annotation effort. The *Annotation Efficiency* (*Ae*) is thus defined as:

$$Ae = \frac{ops_{bsl} - ops_{ft}}{ft_{anns}} \tag{4.2}$$

In this equation, ops_{bsl} signifies the number of correction operations necessary to optimise the baseline output, ops_{ft} represents the count of corrections to optimise the finetuned algorithm's output, and ft_{anns} stands for the number of ground-truth beat annotations the user provides to finetune the algorithm. The total number of operations corresponds to the sum of the possible edit operations: shifts, insertions and deletions.

The metric is particularly suited to the assessment of user-annotations impact on algorithmic performance, providing an intuitive measure of the finetuning process's efficiency. The Ae metric balances the improvement achieved by the finetuning process and the user effort required in terms of annotations. The two key factors influencing the Ae value are:

- *Algorithmic improvement*: An increase in the difference between ops_{bsl} and ops_{ft} results in an elevated Ae metric. This property of Ae rewards algorithmic improvement as it recognises a reduction in the number of corrections required post finetuning.
- *User effort*: On the contrary, Ae decreases with an increase in the number of user annotations (ft_{anns}). This attribute of Ae penalises user effort, implying the more annotations needed, the lower the Ae value.

In terms of the metric's behaviour, Ae exhibits three distinct characteristics:

(a) *Ideal*: When $ops\_ft < ops\_bsl$, the value of Ae is positive, indicating that the finetuning process has enhanced the algorithm's performance.

(b) *No Improvement*: When ops_{ft} is equal to ops_{bsl}, the value of Ae is zero, signifying no change or improvement from the baseline.

(c) *Worsening*: When $ops\_ft > ops\_bsl$, the value of Ae is negative, implying a decrease in algorithmic performance compared to the baseline.

User-centric Evaluation Scores: Limitations for Cross-dataset Evaluation The *Ae* metric offers a methodical approach to evaluate the efficiency of finetuned beat tracking

algorithms, with an emphasis on incorporating the user-annotation effort. Specifically tailored for per-file evaluations, it finds its optimal application within the semi-automatic annotation workflow. Within this framework, users aim to obtain the beat positions for a specific audio file. To achieve this, they initiate the process by running the state-of-the-art beat tracker. The subsequent actions are then determined based on the assessment of the quality of these initial beat estimates:

- Low ops_{bsl}: If the output quality is satisfactory, with minor or easily fixable errors, it often does not warrant further finetuning. Essentially, if the cost of manually correcting the beat estimates (i.e., the correction operations count) is low, manual correction may be more efficient than introducing finetuning annotations.
- High ops_{bsl}: In cases where substantial editing is required, the user is more inclined to annotate a segment and employ the finetuning approach. Ideally, the sum of the annotations required for finetuning (ft_{anns}) and the corrections post-finetuning (ops_{ft}) should be less than the operations needed for baseline corrections (ops_{bsl}).

However, challenges manifest when applying this metric to different scenarios. In cross-dataset evaluations, the user's judgement in the semi-automatic workflow is replaced with the use of a simulated fixed region for annotations (the finetuning region). Whether this is a relative (such as 25% of the entire file's length) or an absolute region (e.g. the initial 10 s), this predetermined approach to finetuning can introduce complications. Specifically, in situations where ops_{bsl} is low (for example, only one operation is needed for correcting the baseline output), manually addressing the error might be more pragmatic. Yet, in a simulated setting, this pragmatic judgement is absent, and the finetuning process is applied regardless.

Utilising a fixed approach in cross-dataset evaluations incorporates many scenarios where finetuning proves either superfluous or ineffective. This not only skews the average Ae value but also introduces a bias that does not accurately capture the metric's practical application[36]. Such potential distortions necessitate a thoughtful approach when applying Ae to broader dataset evaluations.

In light of this bias, we have chosen a discerning application of the Ae metric within this document. Specifically, we have limited its use to evaluations of distinct files and to those challenging datasets where the baseline performance is consistently low across the dataset. This approach ensures the insights from Ae are indicative of true algorithmic performance.

[36] This effect is depicted in Figure A.1.

Both implementations (concerning the user-metrics and graphical display) are shared with the research community through an open-sourced library[37].

4.4 Methodology

As detailed in Section 4.2, our finetuning process relies on a short annotated region for training and an additional region of equal duration for validation. We reiterate that in this work, where we seek to broadly investigate the validity of finetuning over a large amount of musical material, we simulate the role of the end-user. To this end, we obtain these annotated regions from existing beat tracking datasets rather than direct user input. While the duration and location of these regions within the musical excerpt are somewhat arbitrary compared to a practical use case with an end-user, for this evaluation, we choose them to be 5 s in duration each and adjacent to one another, starting from the first annotated beat position per excerpt. By choosing the first beat annotation as opposed to the beginning of the excerpt, we can avoid any degenerate training that might otherwise arise if no musical content occurs within the first 10 s of an excerpt (e.g., a long nonmusical intro).

For the purposes of evaluation, the impact of this configuration of finetuning across the early part of the excerpt has the advantage that it is straightforward to trim these regions to which the network has been exposed prior to inference with the HMM and then offset the annotations accordingly. In this way, we can contrast the performance of the finetuned version with the baseline model [Böck and Davies, 2020] without any impact of the sharp peaks in the beat activation functions across the training region. Note that due to the removal of the training and validation regions when evaluating, the results we obtain are not directly comparable to those in which the full-length excerpts are used. In summary, our evaluation aims to ascertain the extent to which the network's adaptation over a brief region at the start of each excerpt impacts the remainder of the piece.

[37] Available at `https://github.com/asapsmc/beatflow`. This module provides the calculation of our user-centric metrics (E-measure and *Annotation efficiency* – Ae) from a set of beat annotations, beat estimates, and the underlying user-annotations used during finetuning. Furthermore, we provide a visualisation module that extends the `matplotlib` library [Hunter, 2007] to represent graphically the operations necessary to edit a beat series, in a typical annotation workflow.

4.5 Results

In this section, we evaluate the performance on a set of existing annotated datasets, investigate the impact of finetuning in two specific highly challenging musical pieces, and examine the presence and extent of catastrophic forgetting. Together, these elements facilitate a comprehensive analysis of our proposed approach.

4.5.1 Performance Across Common Datasets

While our long-term interest in this work is centred on an end-user workflow scenario, we deem it crucial to commence by evaluating our approach's effectiveness on existing datasets. This step enables us to assess its applicability across a broad spectrum of musical material.

To this end, we utilise four datasets: two from the cross-fold validation training methodology in the baseline model [Böck and Davies, 2020]: the *SMC* dataset [Holzapfel et al., 2012b] and the *Hainsworth* dataset [Hainsworth, 2004]; and two entirely unseen by the original model: the *GTZAN* dataset [Tzanetakis and Cook, 2002; Marchand and Peeters, 2015], which was held back for testing, and the *TapCorrect* dataset [Driedger et al., 2019], upon which the baseline model has never been evaluated. In terms of the musical makeup of these datasets, *Hainsworth* includes rock/pop, dance, folk, jazz, classical, and choral. *SMC* contains classical, romantic, soundtracks, blues, chanson, and solo guitar. *GTZAN* spans 10 genres, including rock, disco, jazz, reggae, blues, and classical. *TapCorrect* is composed mostly of pop and rock music.

The *TapCorrect* dataset is notable for containing entire musical pieces rather than the more customary use of excerpts from 30–60 s, which could provide insight into the propagation of acquired knowledge from the short training region over much longer durations. A summary of the datasets used is shown in Table 4.1. When performing finetuning on *SMC* and *Hainsworth*, we respect the original splits in the cross-fold validation in [Böck and Davies, 2020] and use the appropriate saved model file, which is held out for testing. As stated above, the *GTZAN* dataset is not included in the splits for cross-validation, meaning we cannot make a deterministic selection of which pretrained model to finetune. In the evaluation in [Böck and Davies, 2020], the final output per excerpt is obtained by predicting a beat activation function with the model from each fold of the cross-validation and then taking their temporal average (so-called "bagging") prior to inference with the HMM. Instead of pursuing this strategy here, which would involve finetuning eight separate times (once per fold) and significantly

increase the computation time, we make a random selection among the trained models and only perform finetuning once. Informal evaluation over repeated runs reveals that the specific choice of model has little impact on the results.

Table 4.1: Overview of the datasets used for the evaluation.

Dataset	# Files	Full Length	Mean File Length
Hainsworth	222	3 h 19 m	53 s
SMC	217	2 h 25 m	40 s
GTZAN[a]	994	8 h 17 m	31 s
TapCorrect	101	7 h 15 m	4 m 18 s

[a] Given that our audio file for *reggae.00086* was corrupt, this file has been excluded from all the analysis. Furthermore, due to a processing error, the following files were unintentionally left unprocessed: *jazz.00003*, *jazz.00010*, *jazz.00014*, *jazz.00018* and *jazz.00020*. Thus, from the original 1000 dataset files, we have only analysed 994 *GTZAN* files.

Table 4.2: Mean F-measure scores across datasets for the baseline and finetuning approaches.

Dataset	Baseline	Finetuned
Hainsworth	0.899	**0.945**
SMC	0.551	**0.589**
GTZAN	0.879	**0.917**
TapCorrect	0.911	**0.941**

To measure performance across these datasets, we used the F-measure with the standard tolerance window of ±70 ms. The results for each dataset are shown in Table 4.2. Inspection reveals that the inclusion of finetuning outperformed the baseline state-of-the-art approach for all datasets—even taking into account the deterministic choice of region for finetuning. However, while some general interpretation can be made by observing accuracy scores at the dataset level, we gain a better understanding of the finetuning impact through a scatter plot of the baseline *vs.* the finetuned F-measure per excerpt and per dataset, as displayed in Figure 4.4.

To identify a positive impact of finetuning in the scatter plots, we search for F-measure scores above the main diagonal, indicating that the F-measure per excerpt with finetuning improved over the baseline. Contrasting the scatter plots in terms of this behaviour, we find that for *Hainsworth* and *TapCorrect*, very few pieces fall below the main diagonal, suggesting that finetuning was almost never worse. At this point, it is worth reiterating that if the performance was already very high for the baseline approach, there was limited scope for improvement with finetuning. Indeed, such

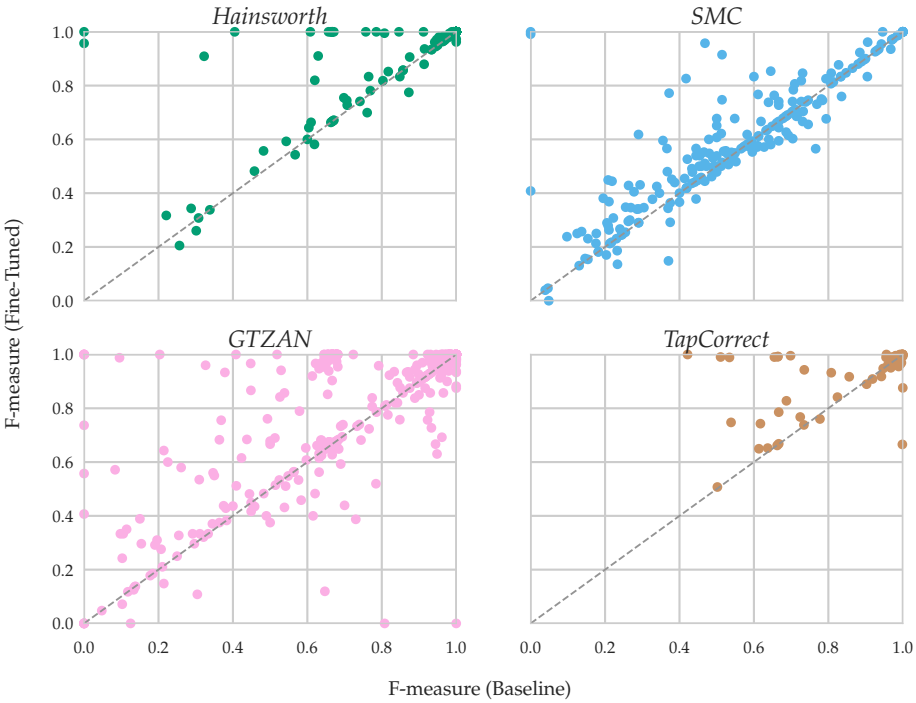


Figure 4.4: Comparison of the F-measure for the baseline and finetuning approaches on in-training datasets *Hainsworth* and *SMC* and out-of-training datasets *GTZAN* and *TapCorrect*.

cases fall outside our primary use-case of interest, which considers the action to take when the state-of-the-art approach fails.

Regarding the nature of the improvements, we observe some explainable patterns. For instance, pieces with F-measure = 0 for the baseline and F-measure = 1 for the finetuning were most likely phase corrections from *offbeat* (i.e., out-of-phase) to *onbeat* (i.e., in-phase) at the annotated metrical level. Similarly, any improvement from F-measure = 0.67 to F-measure = 1 likely resulted from a correction in the choice of metrical level by doubling or halving, i.e., a change to the metrical level corresponding to twice or half the tempo, respectively.

On the other hand, we see that for those pieces that straddle the main diagonal, the impact of the finetuning is negligible. Finally, at the other end of the spectrum, we notice that for *SMC* and *GTZAN*, there are some cases where the finetuning negatively impacted performance. However, there are very few extreme outliers where finetuning was catastrophically worse. Ultimately, the cases of most interest to us are those that sit on or close to the line F-measure = 1 after finetuning, as these represent the clearest benefit.

To gain a more nuanced perspective, we report the counts of all operations necessary to calculate the annotation efficiency, namely the insertions, deletions, and shifts

Table 4.3: Global number of atomic edit operations: correct detections (#det), insertions (#ins), deletions (#del), shifts (#shf) and total edit operations (#ops) for the different test datasets.

Dataset	Model	#det	#ins	#del	#shf	#ops
Hainsworth	Baseline	16,498	923	455	837	2215
	Finetuned	**17,241**	*500*	*246*	*517*	*1263*
SMC	Baseline	4593	810	1337	2457	4604
	Finetuned	**5028**	*670*	*1107*	*2162*	*3939*
GTZAN	Baseline	33,505	3348	1132	2235	6715
	Finetuned	**35,403**	*1911*	*492*	*1774*	*4177*
TapCorrect	Baseline	35,072	3285	1622	910	5817
	Finetuned	**36,659**	*2115*	*1236*	*493*	*3844*

required to transform a set of detections to maximise the F-measure. This information is displayed in Table 4.3. Comparing the baseline and finetuned approaches, we find that across all datasets, fewer total editing operations were required. In fact, per class of operation, the use of finetuning resulted in fewer insertions, deletions, and shifts. In this sense, we deduce that the impact of finetuning was more pronounced than merely correcting the metrical level or phase of the detected beats. Consequently, even considering that from a user perspective, each of these operations might not be equally easy to perform, a reduction across all operation classes highlights the potential for improved efficiency in an annotation-correction workflow.

4.5.2 Impact on Individual Excerpts

In this section, we take a more focused look at the impact of finetuning by examining two specific pieces: a choral version of the song *Blue Moon* from the *Hainsworth* dataset, and a full-length performance of Heitor Villa-Lobos' composition *Choros №1*, as performed by Korean guitarist Kyuhee Park.

Blue Moon

Blue Moon (excerpt number 134 from the *Hainsworth* dataset [Hainsworth, 2004]) is an *a cappella* performance, featuring no drums or other musical instrumentation besides the voices of the performers. Despite this, the performance has a clear metrical structure driven by the lyrics, melody, and orchestration of different musical parts by the singers. Consequently, it represents an interesting case for further exploration, as choral music is known to be extremely challenging for musical audio beat-tracking systems [Holzapfel et al., 2012b].

In Figure 4.5, we plot the log magnitude spectrogram with beat annotations overlaid as white dotted lines. As shown, there is minimal high-frequency information, with most energy concentrated under 4 kHz—consistent with singing. In the middle plot, we observe the beat activation function produced by the baseline approach, along with the ground-truth annotations. Upon inspection, we see that the peaks of the beat activation function are very low, indicative of the low confidence of the baseline model in its output.

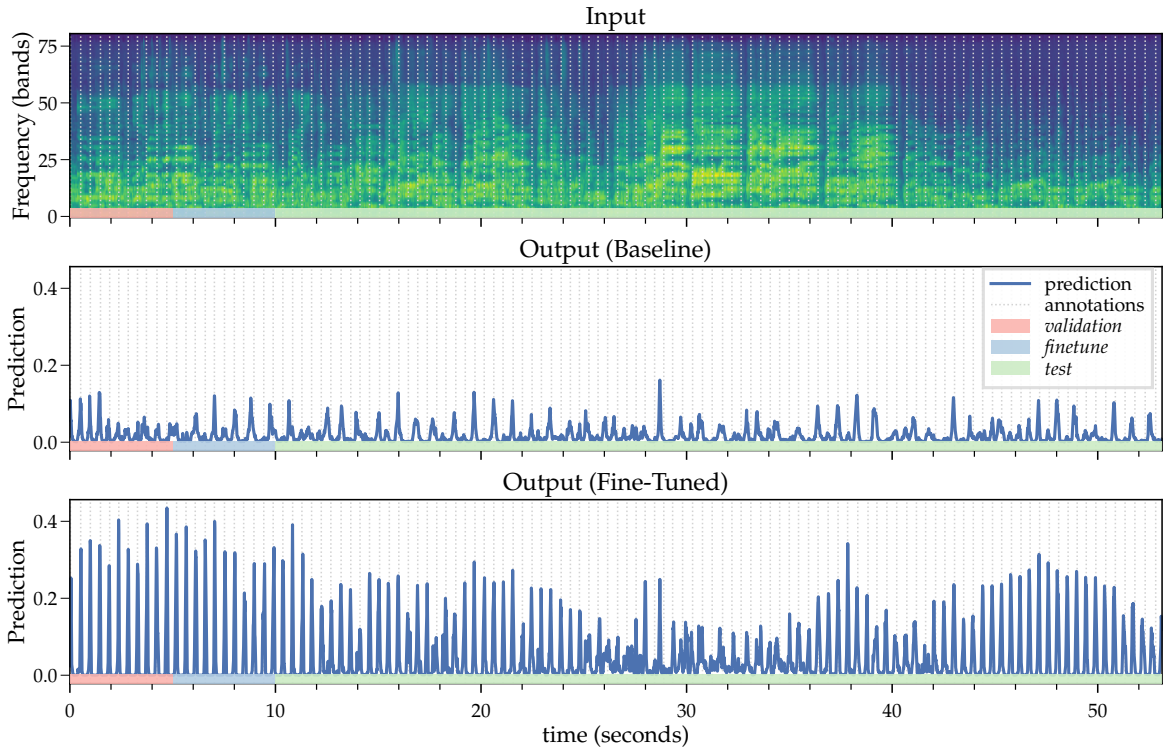


Figure 4.5: Network outputs for the baseline and finetuning approaches on *Blue Moon*. The *validation* region is composed by the 5 s after the first beat annotation (red), the *finetune* region by the following 5 s (blue) and the *test* region starting immediately after and going until the end of the file (green).

Following the same strategy used for the evaluation across the datasets, we use the ground-truth annotations and perform finetuning across the period in the first 10 s of the recording, validating on the first period of 5 s and training on the second period of 5 s. The resulting beat activation function is shown in the lowest plot of the figure. Comparing the two beat activation functions, we observe a profound difference. Once we allow the network to adapt itself to the spectrotimbral properties of the beat structure of this specific piece, we see a series of regular sharp peaks in the beat activation that visually correspond to the overlaid manual annotations.

In terms of quantifying the improvement, we can see in Table 4.4 that when we

finetune, the number of required editing operations drops from 83 to 8, demonstrating the impact that a few annotations can have in transforming the efficacy of the baseline network for challenging content. To visually observe this effect, we can precisely plot which operations are required and at which time instances for both the baseline and finetuned approach, as shown in Figure 4.6. In the upper plot of the figure, we can observe the high number of insertions, which is indicative of the baseline approach estimating a slower metrical level than the annotations. While it is possible to interpolate a set of beat detections to twice the tempo, this is only straightforward in cases where the tempo is largely constant. From the regions around 8 s–11 s and likewise from 25 s–32 s, there are numerous shift operations as well, indicating that the HMM is not able to make reliable beat detections in this region. By contrast, we see far fewer operations in the lower plot with the finetuned beat activation function, all of which are shifts in the form of minor timing corrections. Indeed, a close inspection of the region at the excerpt's end (beyond the 50 s mark), reveals an interesting aspect: the peaks of the beat activation function are strong but misaligned with the annotations. Upon revisiting the manual annotations and the source audio, we can confirm that these specific annotations are drifting out of phase and should be corrected.

Table 4.4: E-measure, correct detections (#det) and insertions (#ins), deletions (#del), shifts (#shf) and total edit operations (#ops) for *Blue Moon*.

	E-measure	**#det**	**#ins**	**#del**	**#shf**	**#ops**
Baseline	0.272	31	56	*0*	27	83
Finetuned	**0.930**	**107**	*0*	1	*7*	*8*

Choros №1

The *Blue Moon* example from the previous section was selected in part due to its challenging musical properties, but also since it could be identified as among the excerpts from the *Hainsworth* dataset whose F-measure score was most improved by finetuning. In this section, we move away from excerpts in existing annotated datasets and instead look towards a simulation of our real-world use case. For this example, we chose a highly expressive solo guitar performance of the Heitor Villa-Lobos composition *Choros №1* as performed by Kyuhee Park[38]. Rather than using a minute-long excerpt, we examined the piece in its full duration of 4 m 51 s. A particular characteristic of this piece and something that is especially prominent in this specific performance

[38] for reference, the specific performance can be found at the following location: https://www.youtube.com/watch?v=Uj_OferFIMk (accessed 25 May 2021)

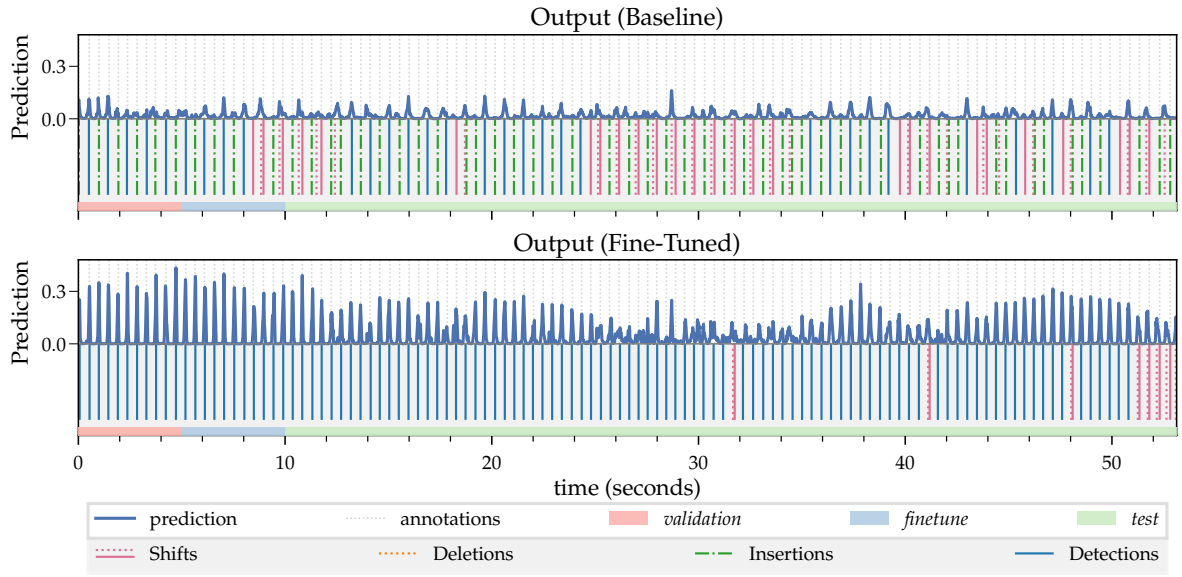


Figure 4.6: Network outputs for the baseline and finetuning approaches on *Blue Moon*. The *validation* region is composed by the 5 s after the first beat annotation (red), the *finetune* region by the following 5 s (blue) and the *test* region starting immediately after and going until the end of the excerpt (green). The dark blue solid line indicates the network prediction. The vertical grey dotted lines show the ground-truth annotations. The vertical light blue solid lines show the correct beat detections. The incorrect beat outputs are notated with the required operation colour (delete—orange, shift—pink, insert—green).

is the extreme use of *rubato* — an especially challenging property for musical audio beat-tracking systems, as discussed in Chapter 2. Indeed, the ground-truth annotation of this piece, conducted entirely by hand in `Sonic Visualiser` [Cannam et al., 2010], was very time-consuming and required frequent reference to the score to resolve ambiguities.

In Figure 4.7, we show the score representation of the beginning of the piece, including the *anacrusis* and the first complete bar. This holds significance as it represents the piece's main *motif*, recurring at several locations throughout its duration. It is composed of three sixteenth notes with *fermata*, indicating a *grand pause*, i.e., that the notes should be prolonged beyond the normal duration, at the discretion of the performer. This notation instructs the performer to an almost *ad libitum* interpretation, which associated with extensive *rubato* across the full piece, creates extreme difficulties for beat analysis. Within the recording, these three sixteenth notes are clearly sounded by plucking, and given the absence of other instruments, they would be straightforward to detect even for a naive energy-based onset detection scheme. However, in the recording, they last over 4 s in duration and are thus highly problematic for beat tracking, because (by reference to the score) all three occur within one notated beat.

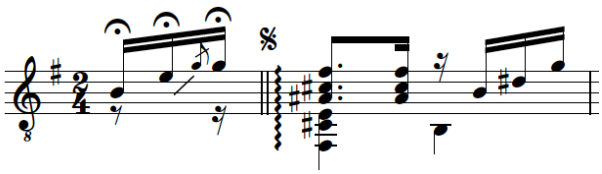


Figure 4.7: Excerpt of the *Choros №1* score (until the end of the first complete bar).

Since the analysis of this piece is not within the domain of annotated datasets, we adapted our finetuning strategy and expanded the region for finetuning to cover the first 15 s of the piece without validation and used the maximum number of epochs. Besides this alteration, we left all other aspects of the finetuning process described in Section 4.2 identical.

In Figure 4.8, the occurrences of a specific musical phrase are clearly marked by a pattern in the log-magnitude spectrogram input of the network, along with the absence of beat annotations. The beat activation function of the baseline network output shows a strong indication of beats at these locations, whereas when performing finetuning, the beat activation is close to zero across all occurrences of the *motif*, despite the existence of clear onsets. In contrast to the *Blue Moon* example in which we observed the network adapt to a specific kind of spectro-timbral pattern to convey the beat, here we find evidence that the finetuning process has allowed the network to learn what is *not* the beat.

The finetuning process has a clear practical impact, as evidenced by fewer required editing operations in Figure 4.9 and Table 4.5. From the zoomed-in plot in Figure 4.9, we can see how well the finetuned network learned to ignore the *motif* once it occurred again just after the 30 s point. Indeed, here we observe a potential downside of the normally advantageous property of the HMM to fill gaps in a plausible way, as we see spurious detections from the finetuned network, which must be deleted. This behaviour, although specific to this piece, suggests that for highly expressive music, including pulse suspensions, a piecewise use of the HMM might be considered. This could prevent these gaps from being filled, either through manual selection of temporal regions for inference or by automatically segmenting and excluding "no beat" regions, as in [Schreiber and Müller, 2018].

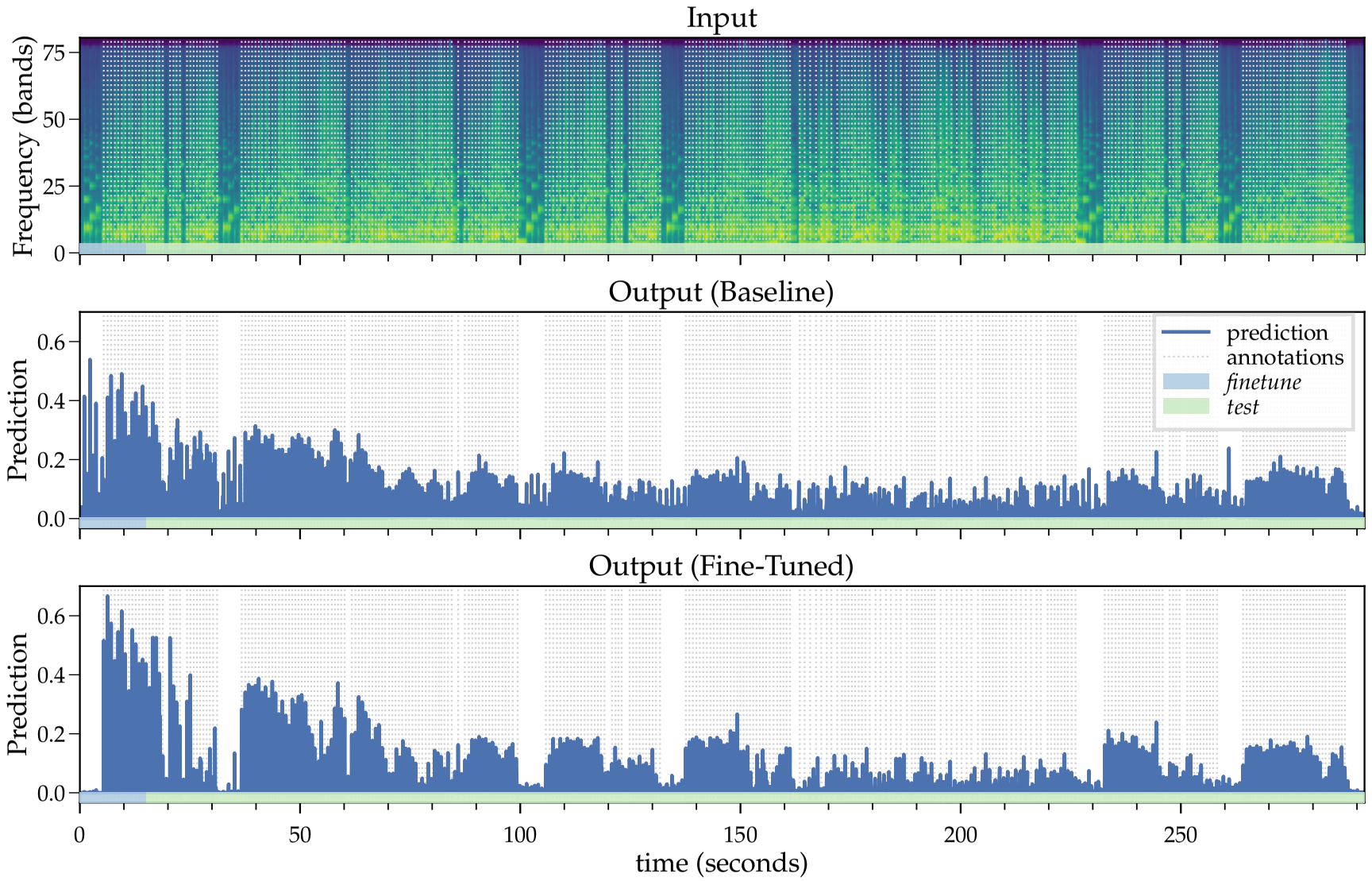


Figure 4.8: Network input and outputs for the baseline and finetuning approaches on *Choros №1*. *Finetune* region 0–15 s (blue) and the *test* region starting at 15 s (green).

Table 4.5: E-measure, correct detections (#det) and insertions (#ins), deletions (#del), shifts (#shf) and total edit operations (#ops) for *Choros №1*.

	E-measure	#det	#ins	#del	#shf	#ops
Baseline	0.555	207	*0*	69	97	166
Finetuned	**0.654**	**236**	*0*	*57*	*68*	*125*

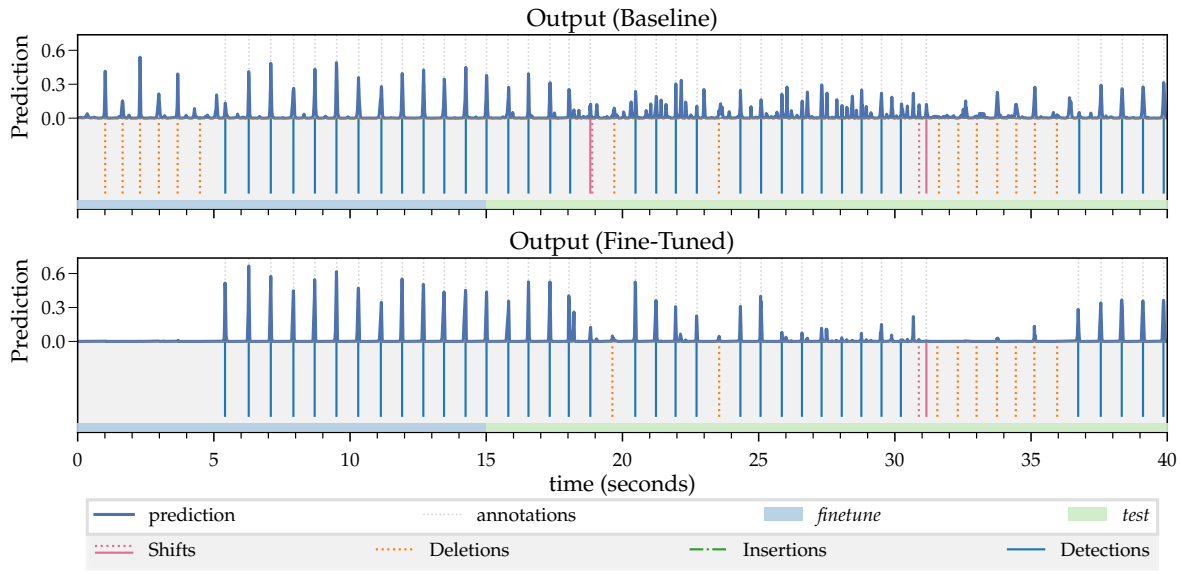


Figure 4.9: Network outputs for the baseline and finetuning approaches on *Choros №1* (zoomed over the initial 40 s). *Finetune* region 0–15 s (blue) and the *test* region starting at 15 s (green). The dark blue solid line indicates the network prediction. The vertical grey dotted lines show the ground-truth annotations. The vertical light blue solid lines show the correct beat detections. The incorrect beat outputs are noted with the required operation colour (delete—orange, shift—pink, insert—green) to correct the annotation.

Catastrophic Forgetting

In the final part of our evaluation, we consider the impact of finetuning from a different perspective. Having established that finetuning is beneficial at the level of individual pieces, we now re-assess the performance of a finetuned network adapted to a given piece on other data. To this end, we investigated the presence and extent of "catastrophic forgetting". Also known as catastrophic interference, it is a well-known problem for backpropagation-based optimisation [McCloskey and Cohen, 1989] and is characterised by the tendency of an artificial neural network to abruptly forget previously learned information upon learning new information. Despite the sequential learning nature of our finetuning adaptation, this is merely episodic, as opposed to the continual acquisition of incrementally available information, which is more commonly

addressed in catastrophic interference [Parisi et al., 2019]. Nevertheless, it is of interest in the context of this work to examine what a finetuned network loses in terms of general knowledge about the beat when adapted to the properties of a specific piece of music.

To explore this behaviour, we return to the *Blue Moon* excerpt from the *Hainsworth* dataset. Across the training epochs of this excerpt, we evaluated the performance of each of the corresponding 24 models over the *GTZAN* and *TapCorrect* datasets. More specifically, for every epoch of the finetuning of *Blue Moon*, we saved the intermediate network and used it to estimate the beat in every excerpt of the *GTZAN* and *TapCorrect* datasets. Using this method, we repeated the evaluation over these datasets 24 times.

Thus far, we have shown that, for this piece, there is a dramatic improvement in the F-measure once the finetuning has completed. However, we have not yet analysed how the F-measure improves over the intermediate training epochs or how the finetuning process, specific to this musical excerpt, impacts performance on other musical content. In the presence of catastrophic forgetting, we should expect some kind of inverse relationship in performance, with the improvement on *Blue Moon* coming at the expense of that on *GTZAN* and *TapCorrect*. In Figure 4.10, we plot this relationship over 24 epochs and indicate that early stopping occurs at epoch 18.

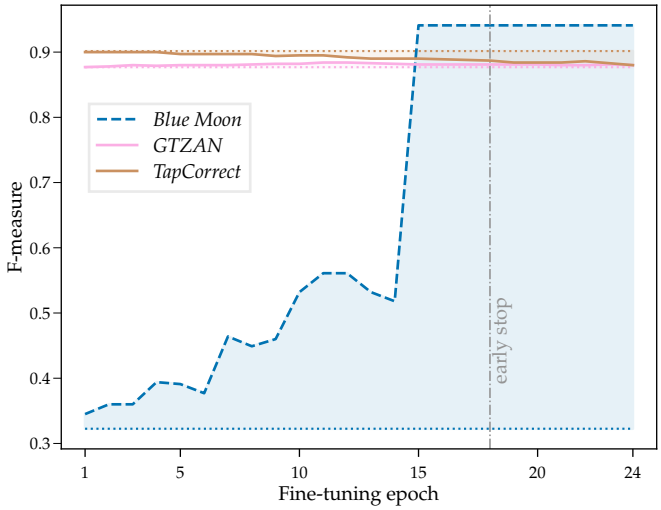


Figure 4.10: Evolution of F-measure during finetuning of the model to *Blue Moon* music, evaluated over the *GTZAN* and *TapCorrect* datasets. Solid lines correspond to the finetuned model and dotted lines to the baseline model.

From the inspection of Figure 4.10, we can observe a rather nonlinear, and indeed non-monotonic, increase in performance for *Blue Moon*. Between epochs 15 and 16, there is a sudden jump in performance, after which the F-measure saturates above

0.90. Looking at the performance across the annotated datasets, we can see that the performance for *GTZAN* is essentially unchanged, and for *TapCorrect*, the F-measure falls by fewer than three percentage points. While our analysis was limited to finetuning on a single excerpt, it would appear that there was a very limited drop in performance due to the adaption of the network to *Blue Moon*. Given that there were approximately 116 k weights in the baseline model, and the network was given a very small temporal observation of 5 s to which it adapted with a reduced learning rate (one-fifth of the baseline training), it may not be surprising that a large proportion of the network weights remained unchanged.

4.6 Discussion

In this chapter, we explored the use of excerpt-specific finetuning of the state-of-the-art system based on exposure to a very small annotated region. We demonstrated that this approach can lead to improved performance across established beat-tracking datasets, and furthermore, we illustrated its potential to adapt to challenging conditions in terms of timbre and musical expression. We believe that the main contribution of this chapter was to demonstrate the potential of finetuning within a user-driven annotation workflow and thus to provide a path towards very accurate analysis on highly challenging musical pieces. Within the wider context of beat tracking, we foresee that this type of approach could be used as a means for rapid, semi-automatic annotation of musical pieces to expand the amount of challenging annotated data for training new approaches.

In spite of the promising results obtained, it is important to recognise several limitations of our work and how they may be addressed in the future. First, our comparison against the state of the art was arguably tilted in favour of the finetuned approach, since per excerpt, we essentially created a new model and compared it to a single general model trained over a large amount of data. That said, our evaluation was carefully designed to exclude the interaction of the trained part of the input signal at inference, and furthermore, we did not claim that our finetuned approach represents a new state of the art. We simply sought to demonstrate that finetuning can be successfully applied across a large amount and variety of musical material. Second, our evaluation was dependent on a rather arbitrary selection of two 5 s regions for training and validation; naturally, as we increase the duration of these regions, we can expect improved performance for the piece under examination, but doing so

would require increased annotation effort on the part of the user, which we sought to minimise as much as possible. Indeed, in the limit, this would resolve to the user annotating the entire piece without any need for an automated solution at all.

The location of these regions was largely dictated by the goal of providing a "fair" comparison with the baseline network. A specific limiting factor of this deterministic assignment of the training region is that if the musical content in the remainder of the piece differs greatly from the information available for finetuning, then we should not expect it to be beneficial. To this extent, we may be underestimating the performance of our approach.

Within a real-world context, we foresee two main differences: (i) the end-user could choose where to annotate and for what proportion of the piece; and (ii) it would likely be advantageous not to exclude the region that has been exposed to the network at the time of inference. Beyond the presence of sharp peaks in the beat activation function, the user-provided beat annotations could also be harnessed for a more content-specific parameterisation of the inference technique, e.g., by setting an appropriate tempo range or some other parameterisation targeted for the presence of expressive timing.

Ultimately, regarding the annotation workflow metrics, we have adopted a preliminary approach where insertions, deletions, and shifts are treated equally for the calculation of the annotation efficiency. We acknowledge, however, that this approach is a simplification and does not reflect the relative costs among operations. Further refinement may be necessary in future studies.

4.7 Summary

This chapter laid out the principles and advantages of user-informed finetuning in optimizing beat tracking performance. We began with an overview of the baseline technique (i.e. the current state of the art) and proceeded to detail our unique methods for finetuning and user workflow-based evaluation.

Then, we evaluated our approach across canonical beat-tracking datasets. Further insights were provided through a detailed examination of the adaptation outcomes for individual music pieces, such as the American standard *Blue Moon* and Heitor Villa-Lobos's *Choros №1*. Finally, we investigated the potential issue of catastrophic forgetting, demonstrating the stability of our approach in this regard.

The chapter wrapped up with a broad discussion on the implications of our results within the larger context of audio beat tracking.

5

A Comprehensive Examination: Leveraging User-Centric Approaches in Beat Tracking

In the previous three chapters, we discussed our work as per our publication timeline. For the purpose of enhancing beat tracking in difficult musical signals, two major contributions were put forth: the high-level parameterisation of beat tracking algorithms by the user (Chapter 3), and the finetuning of these algorithms by means of a user-annotated snippet (Chapter 4). In the current chapter, we aim to integrate these approaches and explore their full potential.

Given the fundamental differences between our human-in-the-loop strategy and "traditional" approaches to beat tracking, the evaluation has been handled carefully. Thus, in order to provide a fair comparison between our technique and the existing state of the art, we have only presented the most conservative estimates of improvement. In this chapter, we will extend this approach to offer a more realistic measure of gains in beat tracking performance. To this end, we will include a greater number of results, using the *de facto* standard metrics for computational beat tracking evaluation, and the

user-centric metrics proposed in Chapter 4.3.

Considering the user-centric nature of our techniques, a case-by-case evaluation approach seems most appropriate. This approach allows for effective customisation and adaptation of our algorithm to each specific context, including audio-specific features and user knowledge and expectations, thereby offering a more nuanced analysis. However, such an approach would require a substantial effort to accomplish comprehensive coverage and ultimately would prove impossible to generalise. So, while our approach is better assessed on a case-by-case basis, large-scale dataset evaluation remains the *de-facto* MIR standard for ensuring reliable results.

To address this misalignment, in this (and also at the following) chapter we adopt a general-to-specific strategy, starting with the evaluation across standard datasets and concluding our evaluation with the most specific and detailed scenarios.

5.1 Scope of Evaluation

A key difference in the evaluation of a finetuning-based approach to beat tracking lies in what is being evaluated (as shown in Figure 5.1). In a traditional evaluation setting, we assess a single model for the test datasets, whereas, in our approach, we evaluate a model for each file of the test datasets.

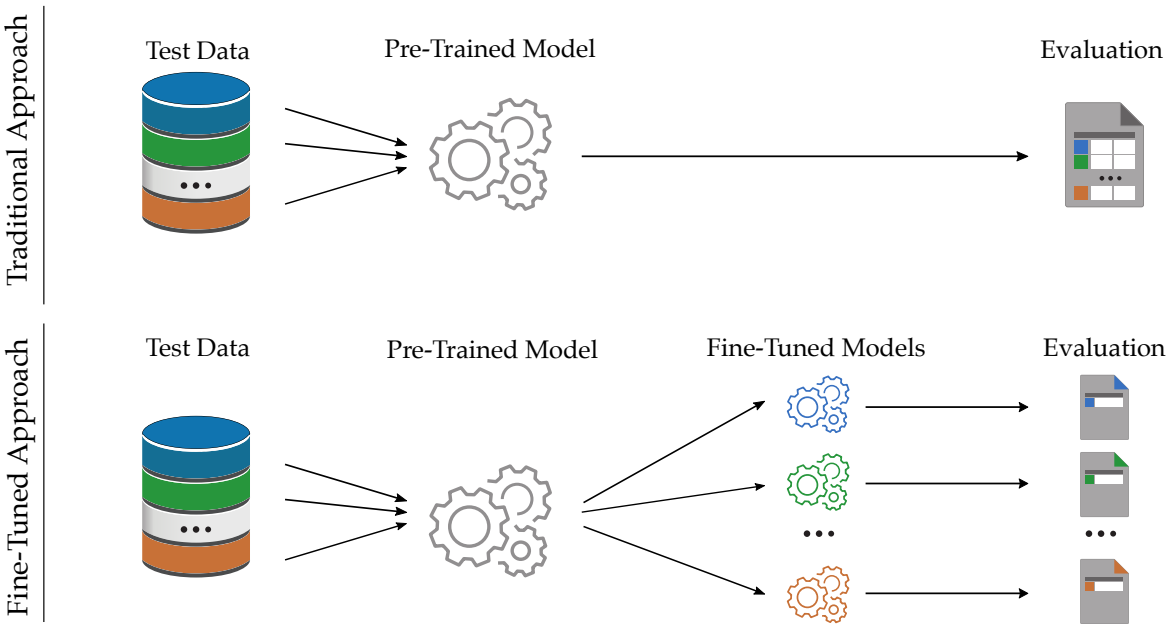


Figure 5.1: Traditional vs Finetuning based evaluation of DL-based beat-tracking.

If not taken into account, this discrepancy may introduce a bias towards our approach and impair fair comparisons with other beat tracking algorithms. For this

reason, the evaluation results reported in the previous chapters were presented in a very conservative manner:

a) we excluded the finetuned part of the input signal for evaluation purposes, thus leaving out some valuable information and potentially resulting in worse performance at inference time;

b) we only reported the beat tracking evaluation of the basic finetuning procedure —finetuning to a user-annotated snippet with the use of data augmentation—, although the existing user annotations could be used beneficially in the beat tracking estimation, namely as high-level contextual knowledge to parameterise the DBN;

c) our evaluation depended on a rather arbitrary selection of two joint 5 s regions for training and validation; of course, we can expect that as we increase the duration of these regions, then we will likely obtain better performance for the piece in question, but doing so would not only compromise a fair comparison, but would also require increased annotation effort on the part of the simulated user (which we sought to minimise as much as possible);

d) the location of the finetuning regions was kept fixed, at the beginning of each sound file; this rule was largely dictated by the goal of providing a "fair" comparison with the baseline network, and additionally as a means of avoiding limitations of the DBN. A specific limiting factor of this deterministic assignment of the training region is that if the musical content in the remainder of the piece differs greatly from the information available for finetuning, then we should not expect it to be beneficial. To this extent, we may have underestimated the performance of our approach.

Conversely, these precautions should not prevent a comprehensive evaluation of our approach, hence the need to broaden the scope of our results. The first couple of points are addressed as follows: a) we report all the results, including (`fullRes`) and excluding (`testRes`) the finetuned part of the input signal for evaluation purposes; and b) besides the basic finetuned model (`ft+da`), we report a representative group of all the available configurations, *i.e.*, the combinations of the user-driven techniques: finetuning (`ft`), data augmentation (`da`) and the DBN customisations (`tg` and `pt`).

Regarding the latter items (c) and d)), we already established in the previous chapter that the most adequate method of selecting the finetuning region is by user judgement.

Yet, despite the central role this option plays in our approach's performance, no other method seems suitable for evaluation across different reference datasets, apart from a general selection. Nevertheless, an adjustment has been made to the length of the finetuning region: instead of adopting a fixed 10 s region for all datasets, we opted to use a per-file segment of 25% of each file length. Our reasoning is based on the fact that all datasets, except *SMC*, consist of variable-length files. Consequently, a relative-length analysis helps equalise the effect of finetuning across all files, regardless of their length.

A final modification was made to reduce the impact of the embedded randomness of the finetuning process in the variability of the results. While this uncertainty stems both from applying dropout in the TCN and using random sampling in the data augmentation procedure, only the latter could be addressed without changing the network architecture. As a result, for the current chapter, the data augmentation procedure has been modified to a deterministic approach, thereby eliminating this source of randomness. Instead of adjusting the frame overlap rate by randomly sampling from a normal distribution (with a 5% standard deviation from the local tempo), we employed deterministic sampling from a linear distribution between $\pm$30% deviation from the local tempo (calculated using the median inter-beat interval across the annotated region). Furthermore, we opted to provide the results averaged over three global runs, except for a limited number of cases that will be specifically noted, where we present results and visualisations pertaining to a single run.

5.2 Methodology

For our comprehensive evaluation, we retained the same datasets used in Chapter 4: the *Hainsworth* dataset [Hainsworth, 2004] and the *SMC* dataset [Holzapfel et al., 2012b] (both known to the baseline model through the cross-fold validation training methodology [Böck and Davies, 2020]), and the *GTZAN* dataset [Tzanetakis and Cook, 2002; Marchand and Peeters, 2015] and the *TapCorrect* dataset [Driedger et al., 2019] (unseen by the baseline model during training). As summarised in Table 5.1, the datasets' total combined duration exceeds 21 hours, while the individual files' lengths range from 12 seconds to approximately 9 minutes, corresponding to a wide variety of musical durations.

Table 5.1: Composition of the Test Datasets.

Dataset	# Files	Type	Full Dataset (hh:mm:ss)	Mean/File (mm:ss)	Min/File (mm:ss)	Max/File (mm:ss)
Hainsworth	222	Variable	03:19:21	00:53	00:12	01:36
SMC	217	Fixed	02:24:40	00:40	—	—
GTZAN	999[a]	Variable	08:20:01	00:30	00:29	00:30
TapCorrect	101	Variable	07:15:07	04:18	02:07	09:05

[a] One of the audio files (*reggae.00086*) was corrupt, thus it has been excluded from all the analysis.

Reported Configurations

In this study, we conduct an extensive analysis of our approach, examining two primary dimensions: *user-driven techniques* and *DBN parameterisation*. User-driven techniques consist of finetuning (`ft`) and data augmentation (`da`), while DBN parameterisation encompasses the use of an *adaptive processor type* (`pt`) and a *tempo guide* (`tg`), whose inner workings are shown in Figure 5.2. In summary, the former operation involves lowering the transition-λ parameter to a value of 75. The second method defines a tempo tolerance window based on the median of the implied tempo of the user annotations for the finetuning region. This effectively *adapts* the DBN processor to handle expressive signals, hence the term adaptive processor type. The latter method defines a tempo tolerance window based on the median of the implied tempo of the user annotations for the finetuning region. As this window is informed by the user's input, we refer to it as the tempo guide, indicating that the tempo is being *guided* by the user's information.

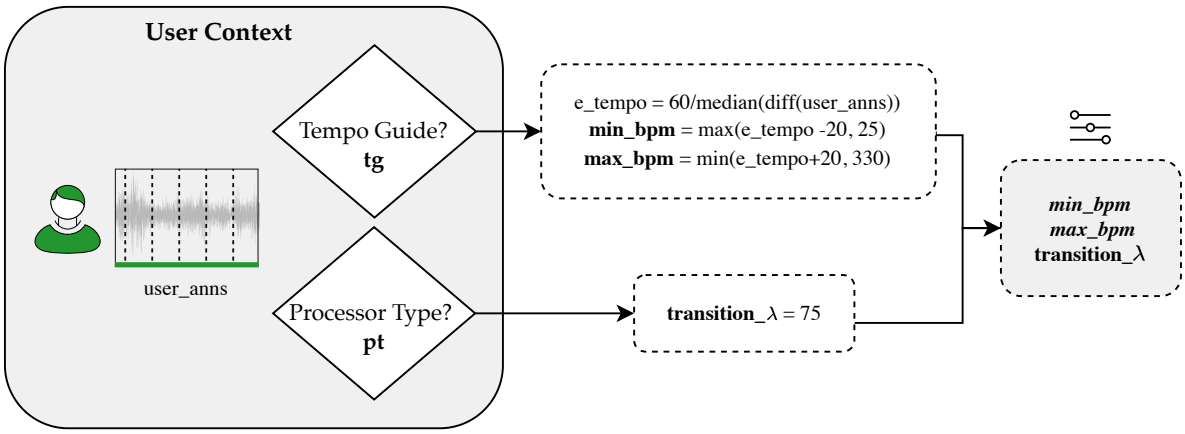


Figure 5.2: DBN parameterisation options.

As shown in Table 5.2, we evaluate eleven valid beat-tracking configurations, though

we primarily focus on the main configurations (i-iv) for the sake of clarity and conciseness. We investigate four primary configurations that encompass two base finetuned models: (i) *without* data augmentation and (ii) *with* data augmentation during the finetuning process. Configuration (ii) `ft+da` represents our primary approach, which has been proposed in previous chapters (referred to as the "finetuned" model).

Table 5.2: Valid beat-tracking system configurations.

	Configurations	*Valid*	*Reported*	*Main*
	(bsl)	✓	✓	✓
(i)	**ft+da**	✓	✓	✓
(ii)	**ft+da+pt**	✓	✓	✓
(iii)	**ft+da+tg**	✓	✓	✓
(iv)	**ft+da+tg+pt**	✓	✓	✓
(v)	ft	✓	✓	
(vi)	pt	✓	✓	
(vii)	tg	✓	✓	
(viii)	ft+pt	✓		
(ix)	ft+tg	✓		
(x)	ft+tg+pt	✓		
(xi)	tg+pt	✓		

Additionally, we investigate two configurations that integrate high-level DBN parameterisation during the inference stage, as depicted in Figure 5.2. The first, (iii) `ft+da+pt`, employs an adaptive processor type tailored for expressive music, as discussed in Chapter 3. This configuration results in a shift towards a slower tempo range and a lower transition-λ parameter value, enhancing the model's responsiveness to tempo changes. The second configuration, (iv) `ft+da+tg`, utilises a tempo guide derived from the user-annotated snippet to parameterise the DBN. To enable a comprehensive evaluation, we also present configurations (v) and (vi) in isolation, applying the `pt` and `tg` techniques solely at inference time on the baseline model.

Furthermore, we include configurations (v) to (vii) to facilitate a fair comparison between the distinct techniques by representing their isolated use. This structure enables us to emphasise the impact of data augmentation on the finetuning process (`ft+da`). We omit the remaining valid configurations, as they either exclude the predominantly beneficial data augmentation from the fine-tuning process (viii to x) or illustrate the combined use of DBN parameterisation techniques during inference on the baseline model (xi).

Finally, we present two categories of results: `fullRes`, encompassing the complete

audio for evaluation, including the fine-tuned and user-annotated segments; and `testRes`, which excludes the annotated portion of the input signal during inference. This approach allows for a more equitable comparison not only with the current state-of-the-art algorithm, which serves as our baseline, but also with other approaches in the field.

5.3 Results

5.3.1 Ablation Study

In this section, we aim to explore how each of the different user-driven techniques presented in the previous chapters contributes to the final performance of the beat tracking system. We include the state-of-the-art [Böck and Davies, 2020] performance as the baseline (`bsl`) to allow for a systematic comparative evaluation.

In Figure 5.3, we display the results across the same group of commonly-referenced datasets, comprising both seen (*Hainsworth* and *SMC*) and unseen data at training time (*GTZAN* and *TapCorrect*).

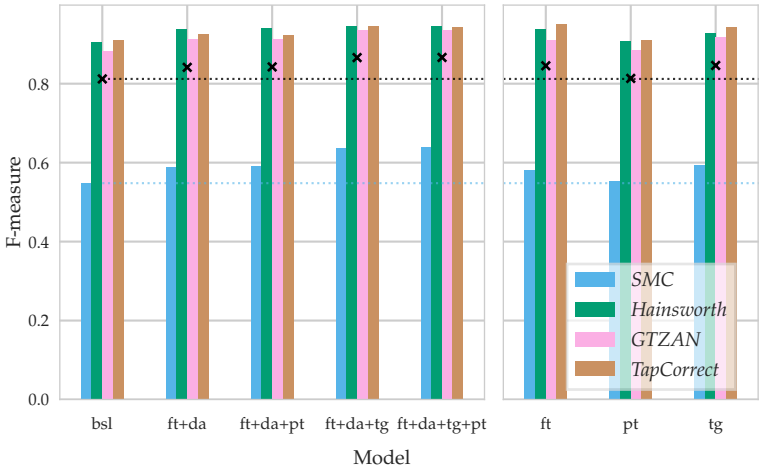


Figure 5.3: Ablation for the *main* and *secondary* set of configurations for the datasets *Hainsworth*, *SMC*, *GTZAN* and *TapCorrect*. The mean of all datasets (in *black*) and *SMC* (in *blue*) baseline F-measure are marked with a dashed line; all datasets mean F-measure are marked with x. (`testRes`)

As expected, given the challenging nature of a large proportion of its musical excerpts, the performance on the *SMC* dataset is quite distinct from the remaining datasets. For that reason, we singled out its average F-measure to allow for a more explicit comparison between the impact of each of the user-driven beat-tracking

configurations on the final beat-tracking performance. It can be observed that the proposed techniques do not contribute equally to the final F-measure across the *SMC* dataset: data augmentation is a clear advantage in the fine-tuning process (`ft+da` vs `ft`), as it not only improves the beat-tracking performance but also accelerates it (given the training regime with fewer epochs). The expressive-adapted DBN configuration (`pt`) reduces the mean performance across this dataset, which hints at the necessity of case-by-case user judgement, also in regard to the effective values used for the `transition-`λ. Finally, it is shown that the most substantial single contribution comes from using the tempo guide parameterisation (`tg`).

Regarding the remaining datasets, similar impacts can be noted, though with reduced magnitude. This behaviour may arise from the broader tempo range distribution of the *SMC* dataset and the corresponding boost on the positive effects of data augmentation and tempo range on the generalisation capabilities of the finetuned network and the quality of beat tracking inference at the DBN level, respectively.

A common trait across all datasets is the positive impact of the finetuning procedure on beat-tracking performance.

Table 5.3 provides a more detailed breakdown of the results. It demonstrates that the main configurations outperform the baseline across almost all datasets, regardless of the evaluation metric. Only two datasets exhibit a different behaviour, albeit for different reasons. In the *Hainsworth* dataset, the AMLc and AMLt scores are so similar across all configurations (except `pt`) that it is impossible to draw any significant conclusions. On the other hand, the results in the *TapCorrect* dataset indicate that data augmentation during the finetuning process does not improve beat tracking accuracy in this case. Given that the only unique characteristic of this dataset is the longer duration of its audio files (approximately 260 seconds, corresponding to a finetuning region greater than 1 minute), we may hypothesise the existence of a duration limit beyond which data augmentation is no longer helpful for finetuning.

Expanding our analysis to the secondary configurations (`ft`, `pt`, and `tg`), we find that the only configuration that does not consistently improve beat tracking accuracy relative to the baseline is the isolated use of the adaptive processor type (`pt`). This outcome is somewhat expected, as only files with expressive characteristics are likely to benefit from this technique by definition.

Table 5.3: Mean of the F-measure, CMLc, CMLt, AMLc and AMLt scores across in-training-set *Hainsworth* and *SMC* datasets and out-of-training-set *GTZAN* and *TapCorrect* datasets for the various configurations. (`testRes`).

Dataset	Model	F-measure	CMLc	CMLt	AMLc	AMLt
Hainsworth	bsl	0.905	0.817	0.853	0.900	0.940
	ft+da	0.939	0.879	0.922	0.913	0.955
	ft+da+pt	0.940	0.880	0.925	0.911	0.955
	ft+da+tg	0.945	0.906	0.943	0.912	0.949
	ft+da+tg+pt	**0.946**	**0.908**	**0.944**	0.914	0.950
	ft	0.940	0.879	0.918	**0.917**	**0.957**
	pt	0.907	0.816	0.857	0.891	0.935
	tg	0.928	0.889	0.928	0.903	0.944
SMC	bsl	0.548	0.376	0.477	0.517	0.659
	ft+da	0.588	0.412	0.532	0.524	0.682
	ft+da+pt	0.593	0.411	0.530	0.524	0.682
	ft+da+tg	0.637	**0.567**	**0.703**	**0.587**	**0.724**
	ft+da+tg+pt	**0.639**	0.563	0.702	0.583	0.724
	ft	0.581	0.403	0.516	0.529	0.682
	pt	0.552	0.380	0.475	0.519	0.657
	tg	0.595	0.537	0.664	0.567	0.701
GTZAN	bsl	0.884	0.792	0.810	0.904	0.928
	ft+da	0.913	0.850	0.871	0.914	0.938
	ft+da+pt	0.913	0.850	0.871	0.911	0.936
	ft+da+tg	**0.937**	**0.907**	**0.938**	**0.917**	**0.948**
	ft+da+tg+pt	0.936	0.906	0.937	0.916	0.948
	ft	0.909	0.841	0.862	0.915	0.939
	pt	0.884	0.793	0.811	0.903	0.927
	tg	0.919	0.885	0.918	0.904	0.939
TapCorrect	bsl	0.911	0.732	0.806	0.850	0.934
	ft+da	0.926	0.754	0.870	0.824	0.941
	ft+da+pt	0.924	0.746	0.865	0.818	0.939
	ft+da+tg	0.945	0.791	**0.933**	0.796	0.938
	ft+da+tg+pt	0.945	0.793	0.933	0.799	0.938
	ft	**0.951**	**0.826**	0.902	**0.873**	**0.951**
	pt	0.910	0.730	0.804	0.842	0.926
	tg	0.943	0.812	0.927	0.827	0.941

Globally, the best results are achieved by combining the base finetuning procedure (`ft+da`) with the tempo range parameterisation of the DBN (`ft+da+tg` or `ft+da+tg+pt`). Upon examining the F-measure scores in the most "straightforward" datasets (*Hainsworth*, *GTZAN*, and *TapCorrect*), the minimum improvement ranges between 1.5 p.p.and 3.4 p.p., while the maximum improvement extends from 4.0 p.p.to

5.3 p.p.. Given that the baseline performance across these datasets is already quite high (F-measure close to 0.9), these results are noteworthy. Again, the most compelling results are obtained with the *SMC* dataset: in this case, the mean improvement shown by the main configurations over the baseline ranges from 4.0 p.p.to 9.1 p.p.(corresponding to 7.3% and 16.6%), positioning our approach at a new state-of-the-art level.

Table 5.4: Mean of the F-measure, CMLc, CMLt, AMLc and AMLt scores across in-training *Hainsworth* and *SMC* datasets and out-of-trainingt *GTZAN* and *TapCorrect* datasets for the various configurations. (`fullRes`).

Dataset	Model	F-measure	CMLc	CMLt	AMLc	AMLt
Hainsworth	bsl	0.904	0.808	0.851	0.888	0.937
	ft+da	0.945	0.878	0.928	0.910	**0.959**
	ft+da+pt	0.947	0.877	0.931	0.906	0.959
	ft+da+tg	0.953	0.907	0.953	0.909	0.954
	ft+da+tg+pt	**0.955**	**0.909**	**0.953**	0.911	0.954
	ft	0.944	0.876	0.923	**0.911**	0.959
	pt	0.905	0.806	0.856	0.877	0.930
	tg	0.930	0.882	0.931	0.891	0.941
SMC	bsl	0.552	0.350	0.465	0.478	0.642
	ft+da	0.607	0.390	0.534	0.491	0.676
	ft+da+pt	0.611	0.391	0.531	0.492	0.675
	ft+da+tg	0.662	**0.550**	**0.718**	**0.559**	**0.727**
	ft+da+tg+pt	**0.665**	0.550	0.717	0.558	0.726
	ft	0.593	0.378	0.512	0.493	0.670
	pt	0.556	0.352	0.463	0.478	0.639
	tg	0.600	0.509	0.662	0.533	0.690
GTZAN	bsl	0.881	0.784	0.809	0.891	0.923
	ft+da	0.915	0.847	0.874	0.905	0.936
	ft+da+pt	0.916	0.847	0.874	0.903	0.934
	ft+da+tg	**0.939**	**0.903**	**0.940**	**0.909**	**0.947**
	ft+da+tg+pt	0.938	0.902	0.939	0.907	0.946
	ft	0.910	0.836	0.864	0.904	0.936
	pt	0.881	0.785	0.809	0.890	0.922
	tg	0.917	0.877	0.918	0.893	0.936
TapCorrect	bsl	0.904	0.718	0.796	0.825	0.916
	ft+da	0.927	0.749	0.869	0.815	0.934
	ft+da+pt	0.925	0.744	0.865	0.810	0.931
	ft+da+tg	0.945	0.787	0.928	0.789	0.931
	ft+da+tg+pt	0.945	0.789	**0.929**	0.791	0.931
	ft	**0.948**	**0.811**	0.896	**0.846**	**0.934**
	pt	0.902	0.716	0.794	0.816	0.909
	tg	0.936	0.792	0.916	0.804	0.930

While these improvements come at a cost of annotating a part of a file (25% in the current evaluation setting), we are using the most conservative (`testRes`) performance scores by leaving this region out of the inference and evaluation process. When we include the user-annotated snippets in the evaluation analysis (`fullRes`), as seen in Table 5.4, the benefits (in terms of beat tracking performance) of annotating just a tiny part of a music piece are unquestionable: the improvement range shifts to the interval between 2.1 p.p.and 6.6 p.p.across the "simpler" datasets, while for the *SMC* dataset the improvement ranges from 5.5 p.p.to 11.3 p.p., numbers which reveal major accuracy improvements in such a difficult dataset. Of note is also the fact that the worst results are consistently obtained for the *TapCorrect* dataset.

For a more user-centric evaluation, we turn our attention to Table 5.5, which presents the counts of operations[39] required to adjust the output set of beat detections in order to achieve the highest F-measure.

Upon comparing the main configurations to the baseline, it is noticeable that there were fewer total editing operations (#ops) across all datasets. In particular, we see maximum gains for the *Hainsworth* and *GTZAN* datasets, with their best configurations (`ft+da+tg+pt` and `ft+da+tg`) presenting cutbacks of roughly two-thirds of the baseline operations. Even for the *SMC* dataset, which by its challenging nature presents the most modest performance, the user-driven configurations enable reductions of 16.0% (`ft+da`) up to 39.0% (`ft+da+tg+pt`) of the baseline correction operations. These results shed light on an additional aspect of our finetuning strategy, which is not just directed toward the annotation of demanding datasets but is also suitable for the annotation of datasets that are more *straightforward*. That is the case of the *Hainsworth* dataset, for which the baseline results are good (E-measure $= 0.886$), but in which with a few user annotations and the right user-driven configuration, the edit–f-measure (E_m)reaches values very close to 1; data that supports the showcase of our method as an all-round tool for user-driven beat tracking.

Given our usage of the `fullRes` results, caution is required when interpreting these findings. This is because the `fullRes` results do not isolate the user annotations (which are right by default), and thus this evaluation is biased towards our approach. Conversely, only half of these annotations are being used for finetuning while the other half are being used for validation. Nonetheless, when interpreting these results cautiously, they still serve a valuable purpose in showcasing the potential of our

[39] Note: To aid in data understanding, columns representing sums (i.e. #det, #ins, #del, and #shf) are retained in the form of integers. However, given that these results are being averaged over three separate trials, there is an underlying rounding error.

approach.

Table 5.5: Mean of the E-measure and Ae scores and sum of the #det, #ins, #del, #shf and #ops scores across in-training-set *Hainsworth* and *SMC* datasets and out-of-training-set *GTZAN* and *TapCorrect* datasets for the various configurations. (`fullRes`)

Dataset	Model	E-measure	#det	#ins	#del	#shf	#ops
Hainsworth	bsl	0.886	20,404	1,157	541	1,079	2,777
	ft+da	0.937	21,337	688	*212*	616	1,514
	ft+da+pt	0.939	21,410	621	*212*	*609*	1,443
	ft+da+tg	0.948	21,916	*84*	232	641	956
	ft+da+tg+pt	**0.950**	**21,935**	85	227	621	*931*
	ft	0.935	21,428	592	336	619	1,547
	pt	0.888	20,445	1,141	510	1,054	2,705
	tg	0.925	21,463	85	237	1,092	1,414
SMC	bsl	0.507	6,287	1,121	1,843	3,293	6,257
	ft+da	0.563	7,017	859	1,571	2,826	5,257
	ft+da+pt	0.567	7,045	890	1,566	*2,758*	5,221
	ft+da+tg	0.639	7,424	*471*	568	2,810	3,849
	ft+da+tg+pt	**0.642**	**7,438**	487	*556*	2,774	*3,818*
	ft	0.549	6,887	919	1,713	2,893	5,525
	pt	0.510	6,322	1,146	1,820	3,233	6,199
	tg	0.579	6,782	493	575	3,426	4,494
GTZAN	bsl	0.859	51,384	4,708	1,436	3,465	9,609
	ft+da	0.901	53,884	3,043	884	2,633	6,552
	ft+da+pt	0.901	53,980	2,955	904	2,622	6,482
	ft+da+tg	**0.935**	**56,557**	*411*	341	*2,588*	*3,338*
	ft+da+tg+pt	0.934	56,537	414	*340*	2,605	3,362
	ft	0.894	53,654	3,154	1,157	2,750	7,062
	pt	0.860	51,448	4,629	1,479	3,480	9,588
	tg	0.912	55,285	472	375	3,800	4,647
TapCorrect	bsl	0.864	36,390	3,764	2,178	876	6,818
	ft+da	0.897	37,532	2,437	1,593	1,060	5,093
	ft+da+pt	0.895	37,461	2,475	*1,591*	1,092	5,161
	ft+da+tg	**0.925**	**39,661**	*141*	2,164	1,229	3,535
	ft+da+tg+pt	0.925	39,622	145	2,141	1,261	3,553
	ft	0.922	39,194	1,327	1,955	*510*	3,791
	pt	0.863	36,318	3,755	2,146	957	6,858
	tg	0.919	39,337	194	1,819	1,499	*3,512*

5.3.2 The *Optimal* Choice

Following a similar approach to the one presented in Section 3.3.2, we aim to demonstrate the theoretical maximum limit on beat tracking accuracy improvement that our

approach can achieve, focusing our analysis on the most challenging dataset: *SMC*.

To do this, we adopt a strategy that closely resembles a real use-case scenario, where a user would select the most appropriate beat tracking configuration for each piece of music they wish to annotate. Instead of evaluating a single beat tracking configuration on the test datasets, we explore the configuration selection on a per-file basis. In this evaluation scenario, we employ a greedy algorithm to find the best configuration based on the F-measure value, our chosen criterion. For each file, we calculate and rank the F-measure for each of the possible configurations, and select the first maximum (as ordered in Table 5.2) as the *optimal* configuration. We use the *reported* set of configurations as the default configuration pool, and the simulated model is denoted as $Optimal_R$.

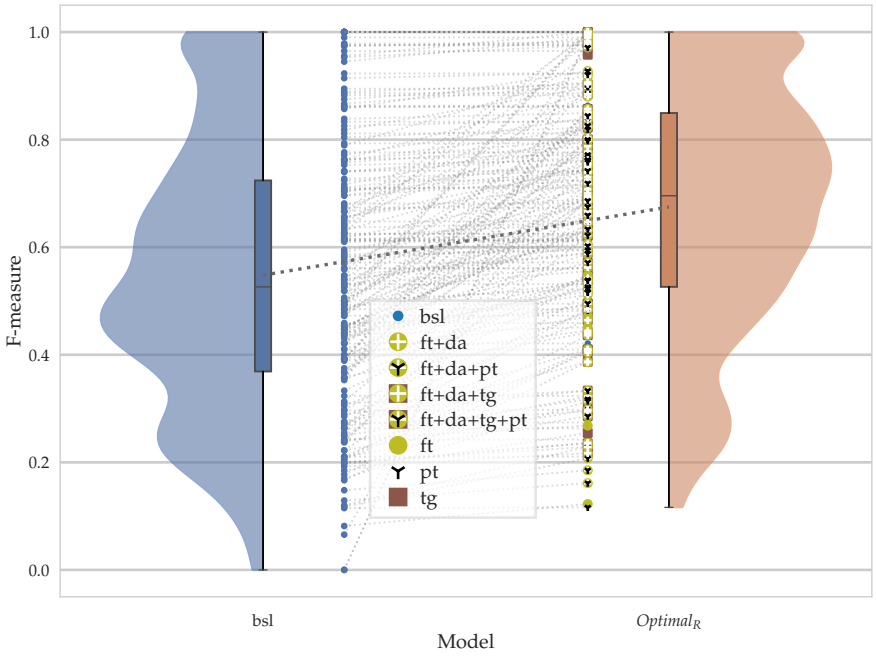


Figure 5.4: Comparison between the *optimal* and the baseline F-measure across the *SMC* dataset (for the set of *reported* configurations and `testRes` type of results).

In Figure 5.4, we demonstrate the impact of optimally choosing user-driven techniques for each file. The improvement over the baseline is quite evident: the optimal set of choices enables a mean increase of 23.1% relative to the baseline performance, while raising the minimum F-measure score from 0 to 0.116. From the perspective of the interquartile range, we observe that the Q1 (25th percentile) of the optimal scores' distribution nearly reaches the median of the baseline scores' distribution, while the median of the optimal scores approaches the Q3 (75th percentile) of the baseline scores.

As illustrated in Table 5.6, there is a clear pattern of improvement (ranging between

Table 5.6: Mean of the F-measure, CMLc, CMLt, AMLc and AMLt scores across the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations).

Model	testRes					fullRes				
	F-measure	**CMLc**	**CMLt**	**AMLc**	**AMLt**	**F-measure**	**CMLc**	**CMLt**	**AMLc**	**AMLt**
bsl	0.548	0.376	0.477	0.517	0.659	0.552	0.350	0.465	0.478	0.642
$Optimal_R$	**0.675**	**0.593**	**0.736**	**0.605**	**0.749**	**0.688**	**0.564**	**0.738**	**0.570**	**0.747**

8.8 p.p.and 21.7 p.p.) over the baseline across all evaluation methods. When we evaluate across the full extent of the files (including the finetuned regions, as shown in the `fullRes` part of Table 5.6), the gains in accuracy are slightly higher (between 9.2 p.p.and 27.3 p.p.). In any case, regardless of the differences between evaluation paradigms (i.e., model per dataset vs model per file), the reported values significantly exceed all previously reported scores for this demanding dataset.

Table 5.7 presents the evaluation results from a user-workflow perspective. Naturally, the best results across all metrics are achieved for the *optimal* model. This global improvement is best summarised by a sharp decrease (a 41% reduction in mean value) in the number of correction operations (#ops) for both types of results (`testRes` and `fullRes`). Of particular note is the remarkable reduction in terms of insertions and deletions (approximately 65%), while the number of shifts decreases by around 20%. Consequently, given the more appropriate weighting of the shifting operation by the E-measure, these scores exhibit greater improvements upon the baseline (14.9 p.p.for the `testRes` and 15.8 p.p.for the `fullRes` type of results, or in percentage terms, 29% and 31%, respectively) than those observed from F-measure. As both deletions and insertions are arguably more costly than small shifting operations, we may conclude that our approach to beat tracking ensures a better annotation correction workflow, not only in terms of quantity (fewer corrections) but also in terms of quality (easier corrections).

Table 5.7: Mean of the E-measure and Ae scores and sum of the #det, #ins, #del, #shf and #ops scores across the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations).

Model	testRes						fullRes						
	E-measure	**#det**	**#ins**	**#del**	**#shf**	**#ops**	**E-measure**	**#det**	**#ins**	**#del**	**#shf**	**#ops**	**Ae**
bsl	0.506	4,593	810	1,337	2,457	4,604	0.507	6,287	1,121	1,843	3,293	6,257	–
$Optimal_R$	**0.655**	**5,572**	*289*	*460*	*1,998*	*2,751*	**0.665**	**7,724**	*395*	*643*	*2,582*	*3,619*	2.582

Furthermore, we look into the global efficiency of the finetuning, as shown by

the Ae score. A value of 2.582 for the annotation efficiency indicates that, through our approach, the $Optimal_R$ output requires approximately 158.2% fewer correction operations than the baseline, per user annotation. Although this is a simulated optimal model, it is still a relevant result, indicating that the user's annotation effort has been very effectively utilised.

Although it might seem challenging for an end-user to consistently pick the optimal configuration, a closer analysis of the results suggests that selecting at least a near-optimal choice is not as demanding.

In Table A.1 (Appendix A), we show the detailed results for each of the configurations for the full list of files from the *SMC* dataset. Upon inspection, we can identify the following principles: a) for most files, several configurations provide the optimal choice; and b) while in many cases the baseline (`bsl`) already offers the best possible results, the user can still pick other configuration(s) (e.g., the one that seems more suitable to the high-level properties of the music) without losing accuracy[40].

More importantly, if we assume that there is an informed user capable of annotating the beat in a given piece of music, the same user should be able to choose between a small subset of high-level basic musical properties (e.g., steady vs expressive, etc.). However, while the ability of users to reparameterise a beat tracking algorithm based on perceived high-level properties has been established in Chapter 3, it is not guaranteed that the best parameterisations for each case correspond to a direct application of the perceived musical properties (i.e., that the best possible result for expressive music is always obtained by using the expressive processor type at the DBN). Nevertheless, the user's perception (with its embedded musical expertise) serves as the best possible proxy for the most effective beat tracking algorithm configurations.

To conclude our analysis of the *optimal* model, we present Figure 5.5, which illustrates the relative importance of each configuration to the *optimal* choice of configurations for all files in the *SMC* dataset. The top row provides the most critical information, indicating the overall weight of each configuration for the per-file *optimal* configuration, calculated through their histogram. The bottom part (associated with the greedy algorithm) mainly serves as supporting data for Figure 5.4. The configurations' contribution to the *optimal* model is filtered for three different sets of configurations: *main*, *reported*, and *valid* (see Table 5.2), displayed from left to right.

We observe a clear pattern across the three initial sets of configurations: the importance of each configuration within the *optimal* set increases from the baseline (`bsl`)

[40] for the *reported* set of configurations and the 217 files of the *SMC* dataset, there are $\approx 9.86 \times 10^{73}$ possible combinations that achieve the optimal F-measure (0.675).

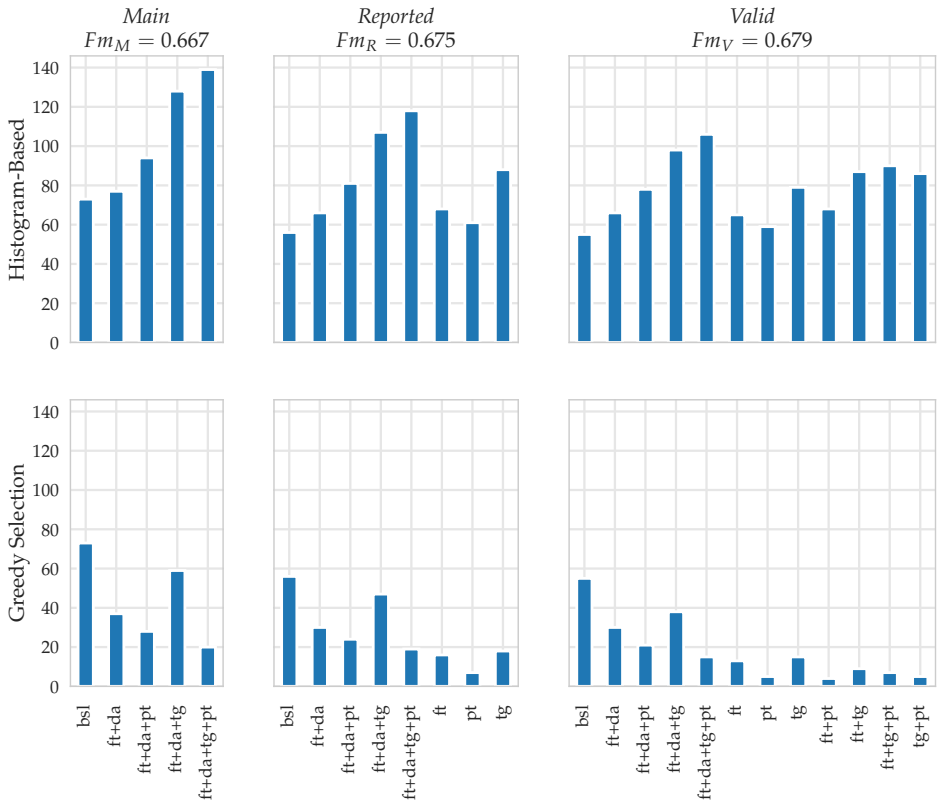


Figure 5.5: Contributions to the *Optimal* F-measure for the different sets of configurations: *Main*, *Reported* and *Valid*. Top row: total number of files for which each configuration presents the optimal F-measure score; Bottom row: number of times each configuration was accounted as the optimal one under the greedy selection algorithm.

to those that integrate more user-driven beat-tracking strategies (`ft+da+tg+pt`). When considering their individual use, the most significant impact on overall performance stems from the tempo guide (`tg`) parameterisation, followed by finetuning (`ft`) and finally, the use of the adaptive processor type (`pt`).

These results are consistent with previous findings, which showed that the use of the adaptive processor type (`pt`) was less effective, while both finetuning and tempo guide demonstrated better improvements. In a broader context, this aligns with the prevalent application of probabilistic graphical models for beat tracking, (Hidden Markov Models (HMMs), which correspond to the practical implementation of the DBN in our baseline approach [Böck and Davies, 2020], are an example of such techniques), which we parameterise through user annotations. Additionally, finetuning enables the adaptation of the preceding architecture, the TCN neural network, allowing the user to guide the overall architecture towards enhanced performance.

Thus, while a case-by-case approach remains the most effective strategy for user-informed beat tracking, the results from this dataset suggest that employing multiple

user-driven techniques leads to a closer approximation to an *optimal* analysis.

5.3.3 Qualitative Analysis of Beat-Tracking Cases

In this section, our goal is to provide a deeper understanding of our approach to beat tracking by examining the beat tracking output of a series of audio files and focusing the evaluation on the user-centred transformation perspective. This case-based analysis enables us to highlight some of the primary advantages and drawbacks of our user-centred strategy.

Given this qualitative approach, we have chosen to present the direct results from a single run (rather than averaging the results over three global runs as in the previous section). We continue to employ the *optimal* model simulation as a means of showcasing our findings, but now we focus our analysis on user-centred metrics, particularly the E-measure, which we use as the criterion for selecting the "best" configuration. Additionally, we extend the analysis to encompass the full extent of the audio files (`fullRes`).

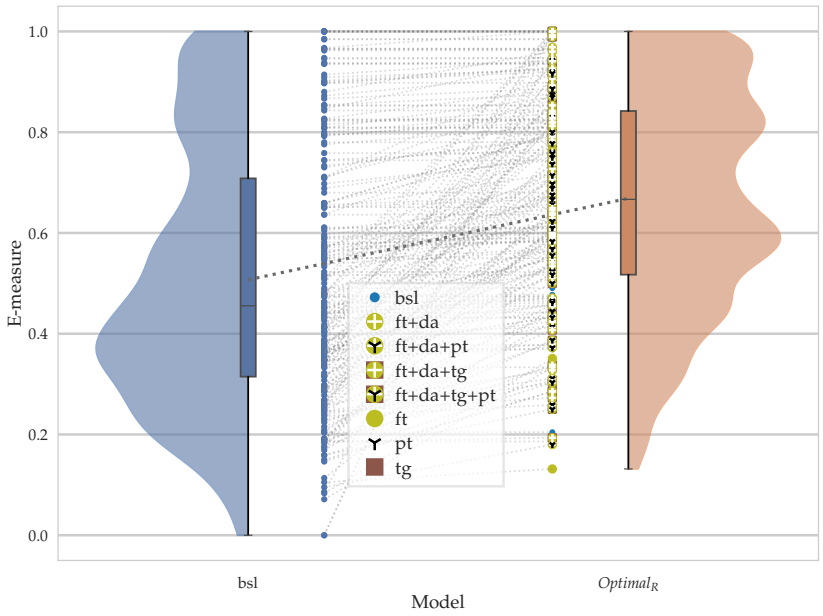


Figure 5.6: *Optimal* choice E-measure accuracy compared to the baseline for each file of the *SMC* dataset (for the set of *reported* configurations and `fullRes` type of results).

As done previously, we begin by examining the impact of an optimal choice of user-driven techniques for every file in the *SMC* dataset. The results shown in Figure 5.6 support those reported earlier (when we used the F-measure as the criterion for selecting the best configuration, as presented in Figure 5.4). The optimal choice of

user-driven techniques per file significantly affects the overall beat tracking accuracy, even when considering the full length of the files (`fullRes`) and using the E-measure as the criterion and the reporting metric.

Figure 5.7 delves deeper into these results by categorizing them based on distinct E-measure increments when comparing the optimal set of configurations and the baseline. An *ad-hoc* inspection aimed at selecting a distribution nearly identical among categories led to the division into six groups.

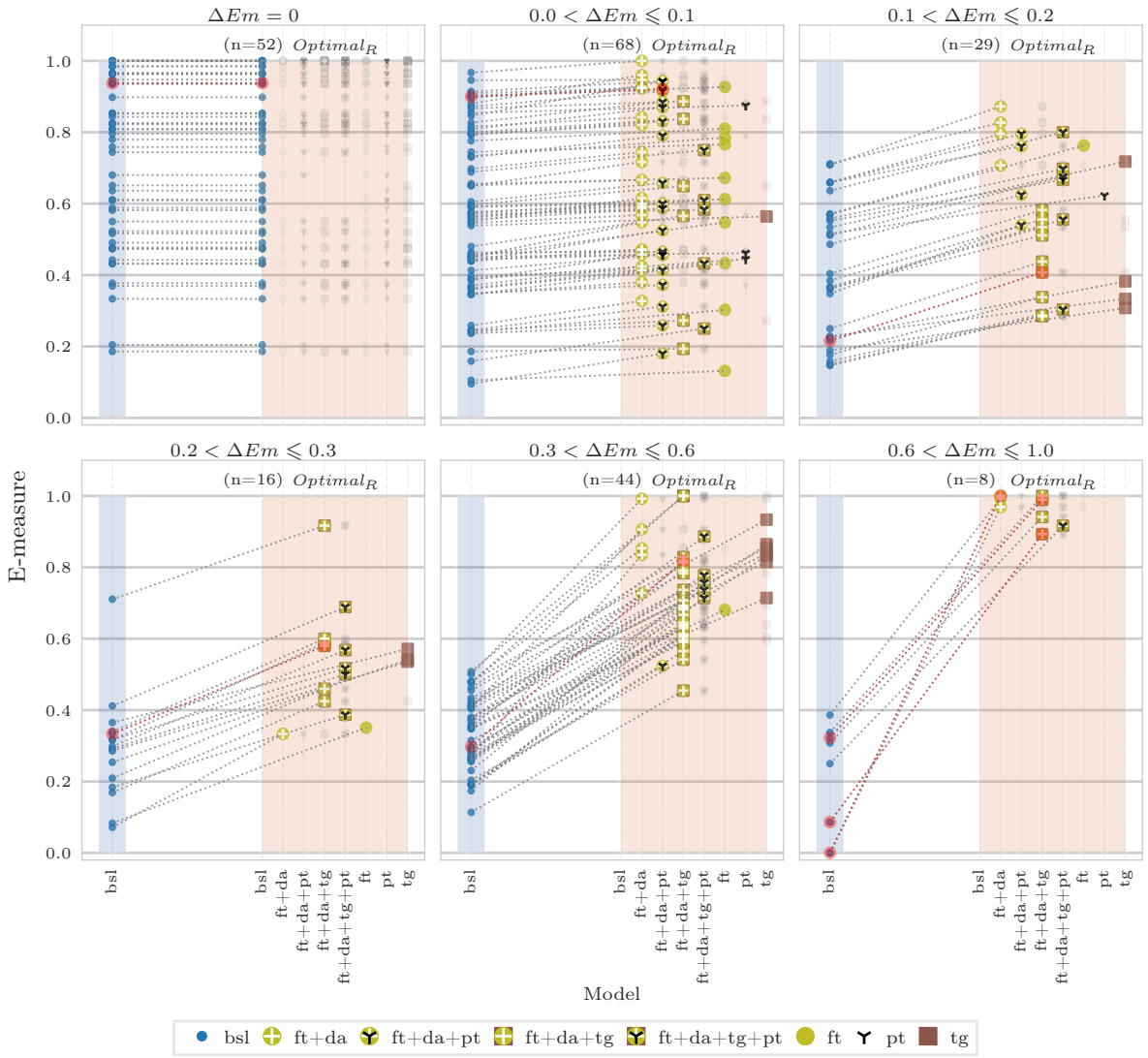


Figure 5.7: Comparison between the baseline and the *optimal* configuration choice for all files of the *SMC* dataset, grouped by ranges of the E-measure increment. The left part (in *blue*) of each subplot represents the baseline E-measure score, while the right part (in *cream*) shows the best possible score for each file, according to the greedy-selection algorithm (in colour) or according to the histogram (in greyscale). The *red* points mark the cases to be detailed in the remainder of this section. (`fullRes`). Notably, the optimal models span all configurations, indicating no clear correlation between E-measure increase and configuration type.

The first observation to note is that there is no apparent relationship between the increase in E-measure and the type of configuration used; in other words, across all six groups, there are *optimal* models representing all configurations. Next, we examine the beat tracking output of some specific files (highlighted in *red*) following this procedure: for each category, we select the first file (sorted by file `id`) and compare the output of the *optimal* configuration with the baseline.

This section includes additional examples that help illustrate our approach to our analysis. We provide detailed information about each music piece, allowing readers to identify and audition the files while reviewing our findings.

(a) $\boldsymbol{\Delta E_m = 0}$: This group accounts for 24% of the dataset files, where the baseline score is already optimal. As observed in Figure 5.7, for a significant portion of these files, there are several configurations that present the optimal beat tracking accuracy.

We examine one of these cases in detail (Figure 5.8), comparing the baseline with a configuration holding the same optimal E-measure score (`ft`). The beat activation function exhibits two distinct parts: one with clear-cut peaks (until 21 s) and another with more ambiguous output (from 21 s until the end). Musically, while in the first part, we can hear a double-bass marking the beat, in the second part, this instrument is absent, and thus there are no beat-synchronous onsets for the beat tracking algorithm to "follow".

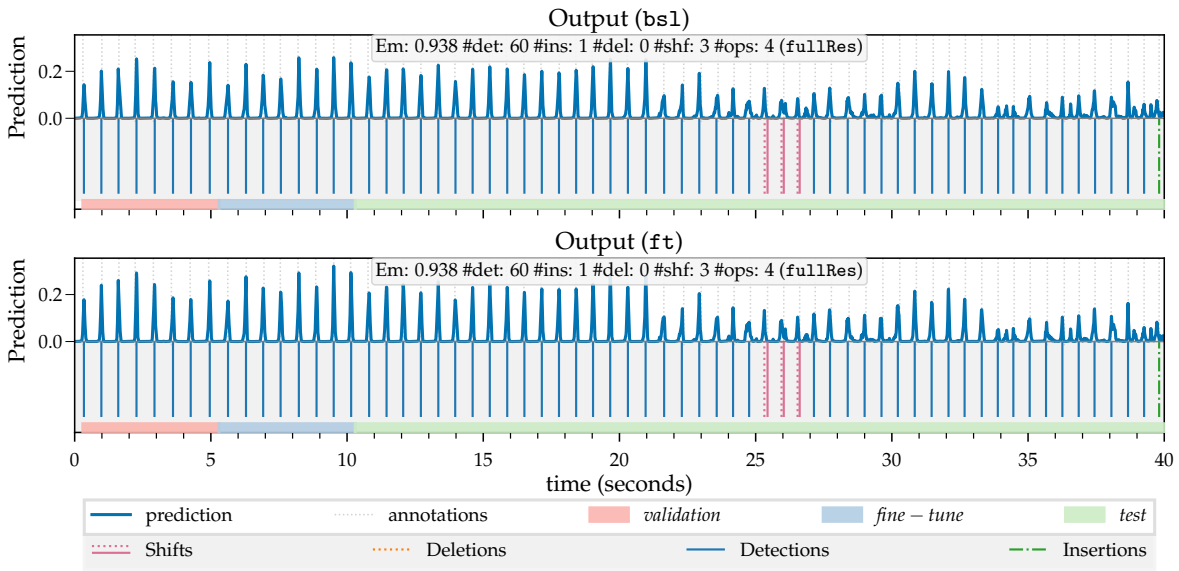


Figure 5.8: SMC_013 — Henryk Wienawski *"Faust" Fantaisie Brillante*, in the Budapest Strings Chamber Orchestra interpretation.

The network's beneficial adaptation is evident from more salient peaks in the finetuned setup, especially in the signal's beginning. However, since the fine-tuning region does not cover the latter problematic section, the learning does not propagate to the rest of the file, providing no significant advantage in terms of beat annotation. The finetuned model exhibits a similarly ambiguous beat activation function, thus producing the same errors as the baseline (a series of shifts after 25s and an insertion near the end of the file).

(b) $\mathbf{0.0 < \Delta E_m \leqslant 0.1}$: In this group, there is a minimal improvement in E-measure accuracy, a pattern that occurs for 31% of the *SMC* dataset.

By examining the annotation-based visualisation of the baseline (in Figure 5.9), we see that all five errors are labelled as shifts. However, these shifts have different causes: while the first is due to the existence of two very close peaks in the network prediction, where the smaller peak, instead of the larger (which is naturally picked by the algorithm), corresponds to the ground truth, the others are due to misalignment between the peak of the beat activation function and the ground-truth annotation.

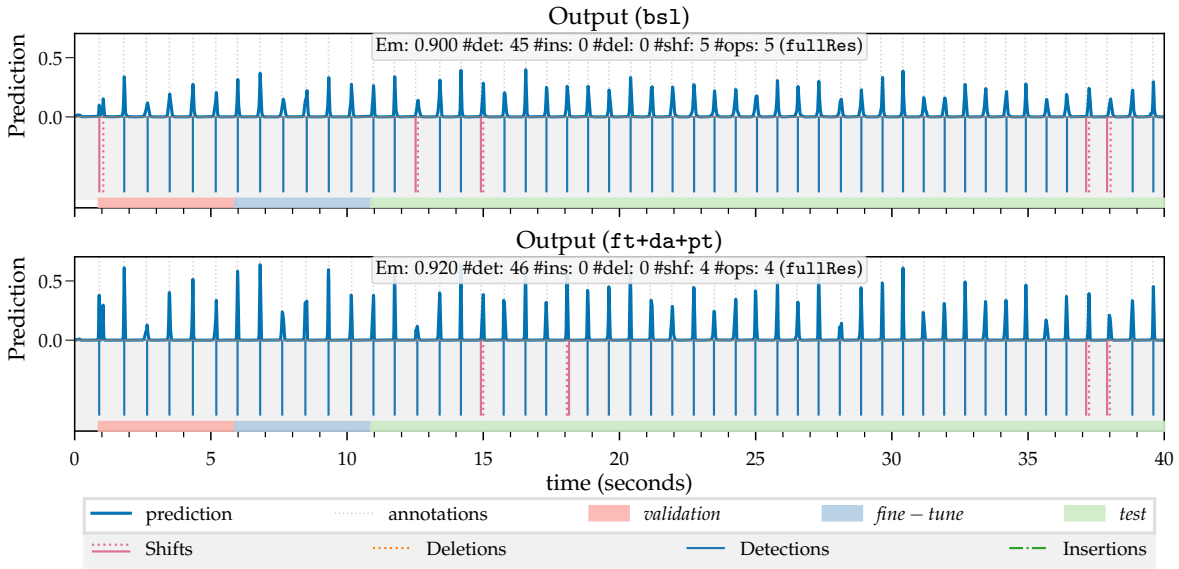


Figure 5.9: SMC_010 — Erik Satie' *Gymnopédie No.3*, II Movement, in Debussy's orchestral form.

The first error is particularly interesting when we examine the underlying musical signal: after auditioning and conducting a detailed inspection of the spectrogram,

we understand that the two instruments (harp and double-bass) are not exact[41] at attacking the initial note, resulting in two distinct attacks that correspond to the onset of the harp and the double-bass. It is also worth noting that through finetuning, we can instruct the beat tracking algorithm which onset it should adhere to. This is the reason that while the finetuned configuration can correctly detect this first beat, the baseline fails to do so. Nevertheless, as the introductory musical motif (up to eleven seconds) does not repeat until the end of the excerpt, the beat detection in this segment does not benefit from finetuning.

(c) $\mathbf{0.1 < \Delta E_m \leqslant 0.2}$: This group contains 13% of the dataset files.

As shown in Figure 5.10, the baseline beat tracker detects roughly double the number of beats compared to the ground truth, a clear signal of a metric level error (i.e., the beats are detected at double the correct tempo); however, the finetuned configuration corrects this issue, as can be seen by the reduction from 24 to 0 deletions. One aspect worth mentioning is the inability of the finetuned configuration to correctly display the first beat, due to the absence of a clear peak in the beat activation function.

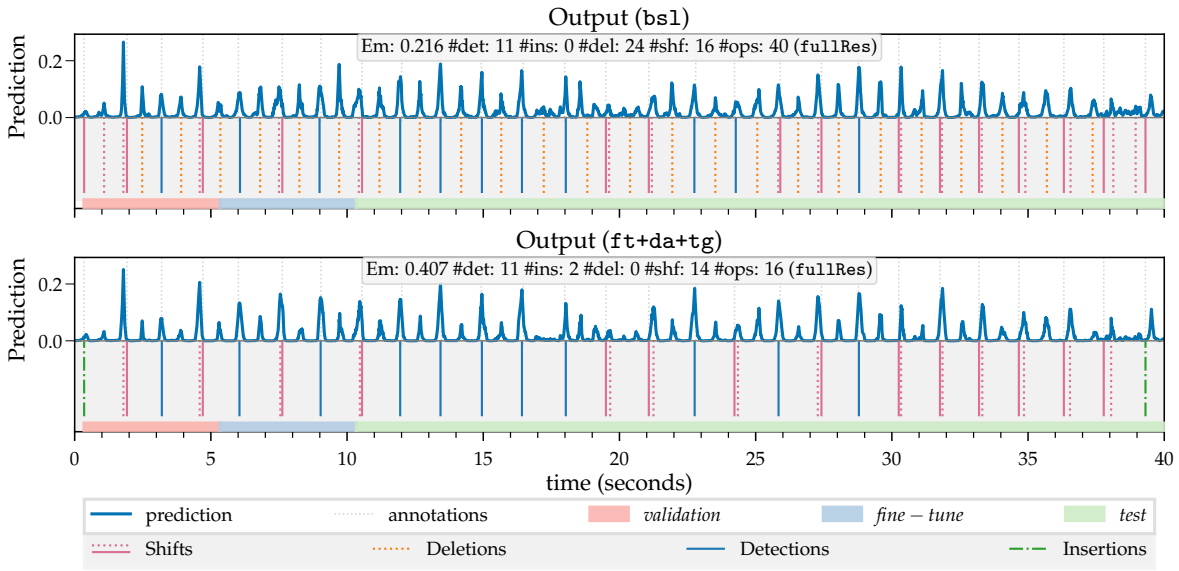


Figure 5.10: SMC_005 — Liszt's *Liebestraum No.3*, interpreted by The Budapest Strings.

Apparently, this behaviour could indicate a possible disadvantage of our prior choice of the start of the finetuned region: exactly at the first beat position,

[41] It is not clear if this effect is due to musical expressiveness or sound recording artifacts, such as the existence of strong delayed signals.

contrary to an earlier selection, which would allow the beginning of the peak shown in the prediction to be encompassed. However, a further test with the beginning of the finetune region exactly at 0 s revealed no improvement in this regard (as shown in Figure 5.11).

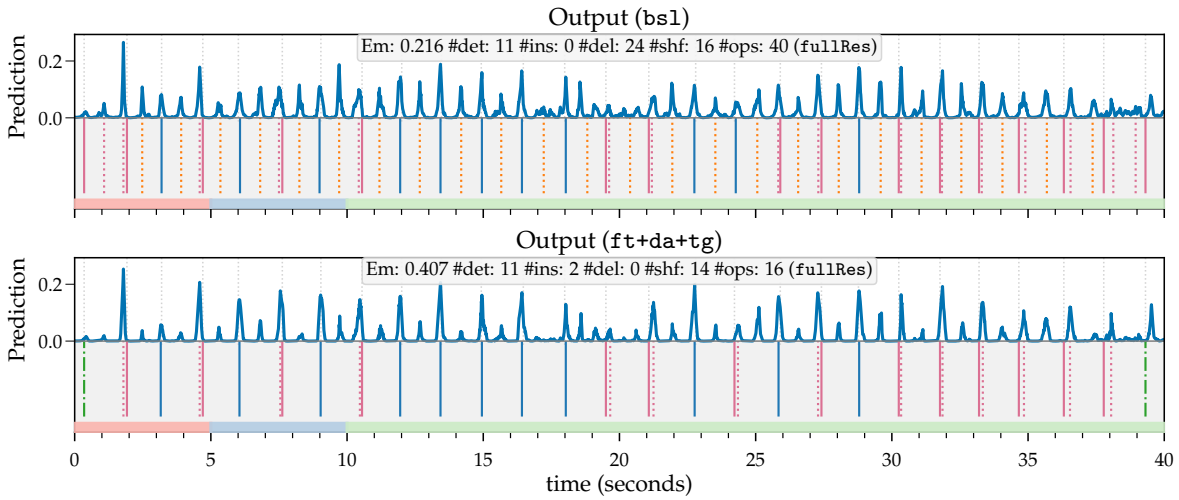


Figure 5.11: Alternative analysis of SMC_005 — finetuned to the region starting at 0 s.

Upon auditioning the musical signal, we realised that the problem lies in the inexistence of any sound at the location of the first ground-truth beat annotation, which is a common characteristic of very expressive music, and illustrates the perceptive nature of human beat induction. Thus, in a preliminary assessment (i.e., without delving into the process of training the network), this demonstrates a limitation of the finetuning procedure in the presence of "no-energy" beats, at least when not preceded by other beats. Finally, the remaining errors of the finetuned configuration are labelled as shifts, and are due to the misalignment between the peaks of the beat activation function and the ground truth; in terms of the root music signal, these may reveal other "no-energy" beats or other types of expressive musical accentuation in the piece of music, or may even be caused by occasionally imprecise ground-truth annotations.

(d) $\mathbf{0.2 < \Delta E_m \leqslant 0.3}$: for this group, that accounts for 7% of the dataset files, there is a larger improvement on the E-measure score.

Figure 5.12 displays a case where the finetuned region is correctly analysed by the finetuned configuration. However, as this musical intro does not recur throughout the excerpt, finetuning benefits vary. For instance, it enhances the 32–40 s region

but offers no improvement for the 25–31 s, likely due to their musical similarity (or lack thereof) to the finetuned snippet.

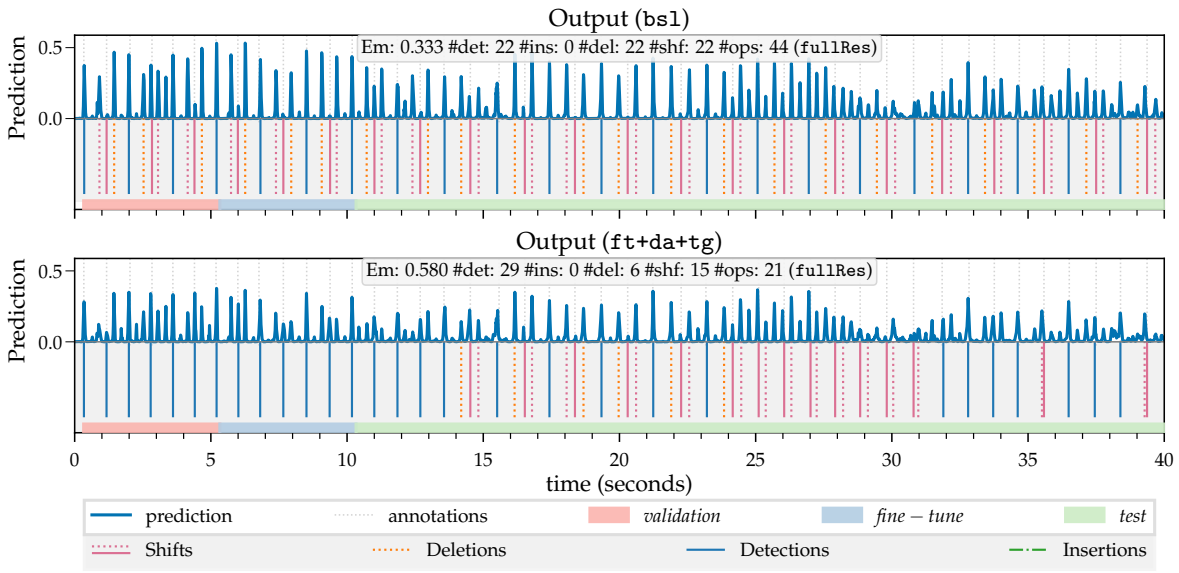


Figure 5.12: SMC_002 — Bizet's orchestral piece *Carmen Fantasy, Op. 25: IV. Allegro Moderato*.

(e) **$0.3 < \Delta E_m \leqslant 0.6$**: approximately 20% of the *SMC* dataset fall into this group.

This example pertains to a guitar-solo piece, which is challenging for beat extraction due to its highly expressive nature. As a result, the baseline configuration exhibits numerous beat tracking errors, resulting in a very low E-measure, i.e. $E_m = 0.298$. A great part of these errors are due to a metrical error, which is revealed by the pattern *shift-detection-...-shift* visible in the top part of Figure 5.13.

The finetuned configuration (`ft+da+tg`) is able to correct a great part of these errors, by enhancing the beat-corresponding peaks in the beat activation function and whitening the remaining noisy signal. Nevertheless, some errors remain, due to "no-energy" beats (e.g., the insertion at 16.5 s) or ground-truth annotations in the vicinity of stronger peaks in the network output prediction (e.g., immediately before the 15 s).

A peculiar type of error is also revealed in this inspection: despite the existence of a strong peak aligned with the ground truth, the algorithm opted for a nearby location with lower predictive energy. This is the case for the first two shifts (near 5 and 12 s), of which we note the first, included in the validation region (if it were in the finetuning region, we would expect it to be correctly detected by the algorithm).

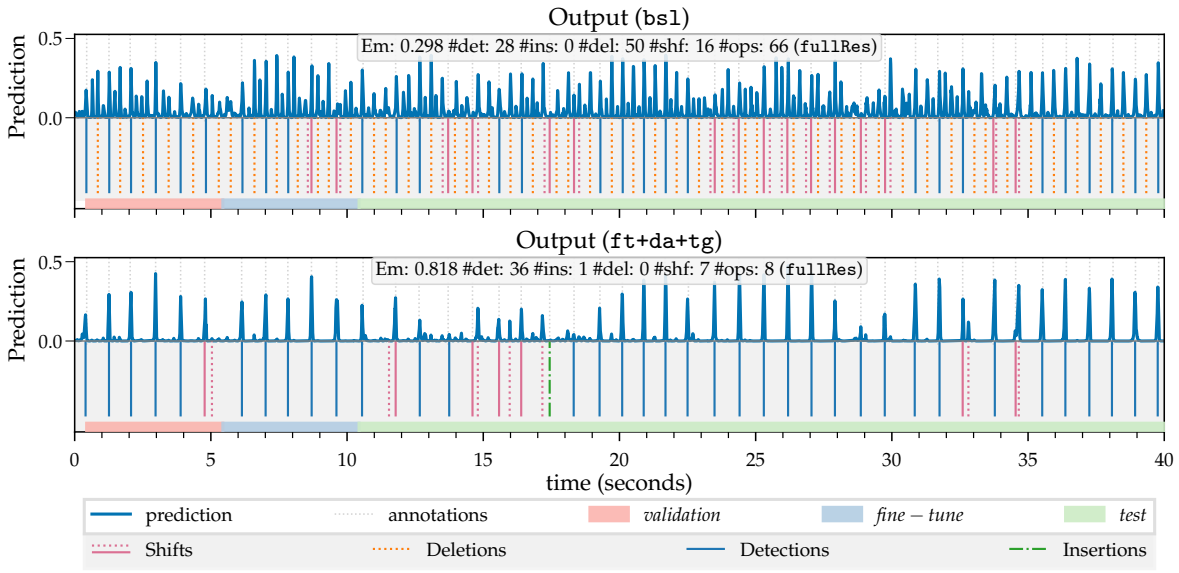


Figure 5.13: SMC_003 — *Étude No.4*, a guitar-solo piece by Leo Brouwer, here interpreted by Timo Korhonen.

(f) $\mathbf{0.6 < \Delta E_m \leqslant 1}$: Although accounting for only 4% of this dataset, there are very illustrative examples in this category, from which we select three.

The first example reveals the perceptual nature of beat induction and (consequently) one of the great challenges of automated beat tracking. This is the case of *VI. Closing*, by Philip Glass. An excerpt of this piece was annotated for the SMC dataset over a 3/4 metre (with the flute playing 3 pairs of eighth-notes as depicted in Figure 5.14a), while in fact, it was written by the author as a 4/4 (with the flute playing triplets as shown in Figure 5.14b). Beyond the discussion of metre perception, this example demonstrates the potential of finetuning-based beat tracking, as it allows us to understand the behaviour of our approach in the presence of concurrent metrical and beat interpretations.

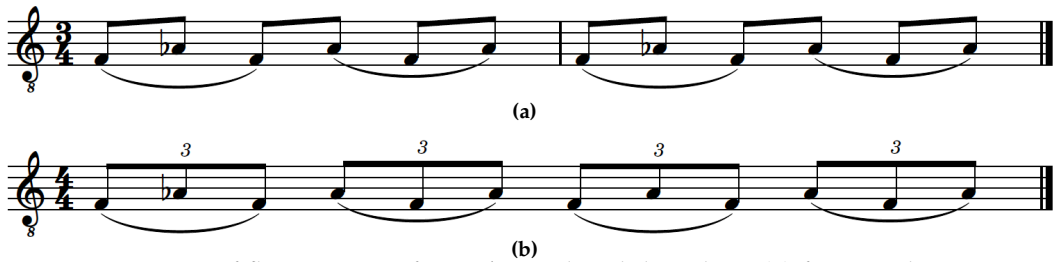


Figure 5.14: Excerpt of flute's voice of *VI. Closing*, by Philip Glass: (a) first two bars in ternary metre, as annotated for the *SMC* dataset; (b) first bar in quaternary metre, as notated by the composer.

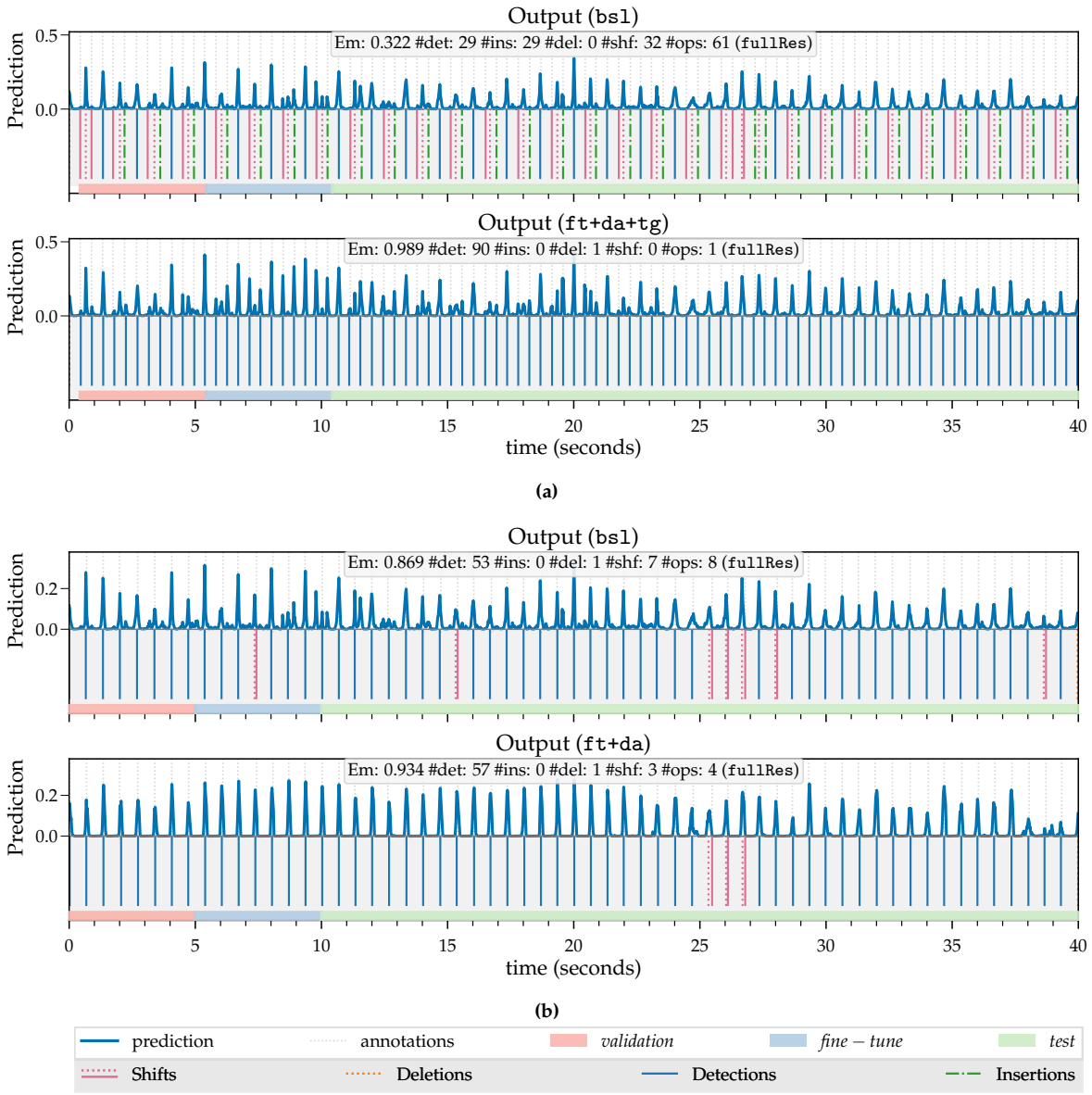


Figure 5.15: SMC_008 — *VI. Closing*, by Philip Glass, interpreted by The Philip Glass ensemble (a) ground truth in ternary metre, as annotated for the *SMC* dataset; (b) ground truth in quaternary metre, as written by its composer.

In Figure 5.15a, we observe a recurrent error pattern for the baseline (a correct detection, a shift, and an insertion for each triad of ground-truth annotations), which may be a sign of the previously discussed ill-suited metre. However, except for an initial deletion (at 0 s), which would most likely be correct if we used this area in the finetuning region, the finetuned output shows an effective adaptation to the rhythm structure intended by the user (the flute playing 3 pairs of eighth-notes on a ternary metre). As can be seen in Figure 5.15b, we demonstrate the successful outcome of using the composer's metrical structure

as the ground truth, ultimately showing that our approach can empower the user to enforce a (musically reasonable) intended metre to the algorithm. In this example, we also relocated the finetuning region to the 0–10 s area, with the intention of "teaching" the algorithm to avoid identifying the initial deletion as a beat, once again effectively.

The intermediate example (shown in Figure 5.16) is a very interesting showcase of the intrinsic difficulties of beat tracking. The instrumentation of the beginning of this piece of music is minimal: a cello and a synthesizer pad, both of which do not exhibit percussive features (in the form of sharp attacks). On an informal analysis[42], some listeners reported an alternative beat interpretation (with a phase shift of 180°). These alternate beat possibilities are displayed clearly by the baseline prediction (although the errors shown are due to the inexistence of peaks in a great part of the beat ground-truth). Yet, in the finetuned predictions, we can see the effect of the network being adapted to the user's preference, by displaying clear prediction peaks solely at the preferred beat phase. This capability is a finetuning hallmark, i.e., the ability to enforce the user's beat interpretation in the presence of alternative beat activation function sets of peaks.

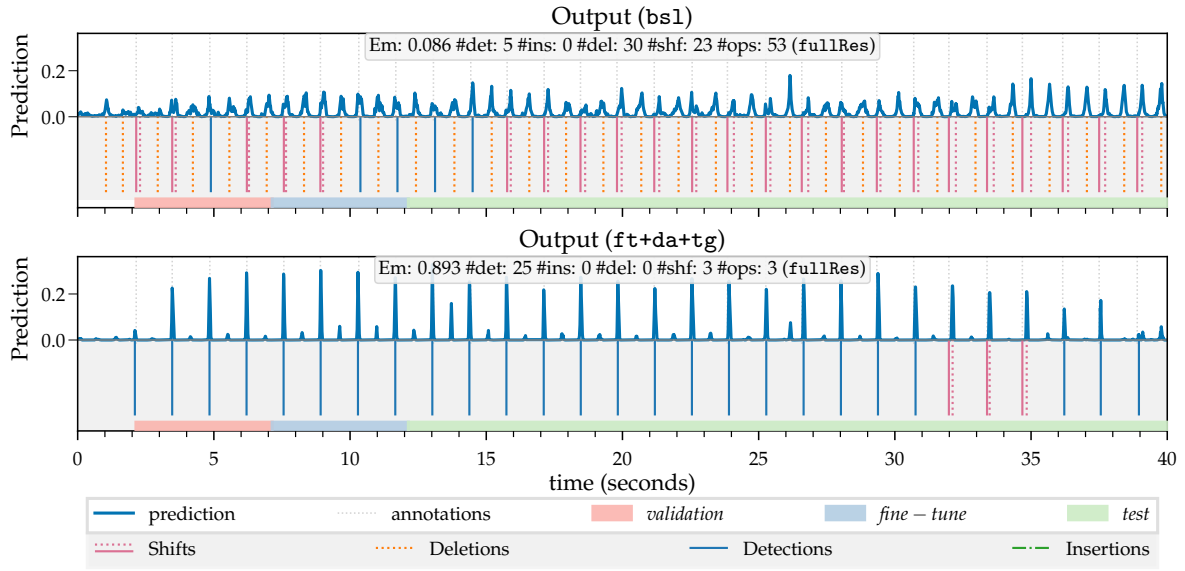


Figure 5.16: SMC_064 — *Ghosts of Things To Come* by Clint Mansell & The Kronos Quartet.

[42] Carried by a group of five professionally trained musicians.

(g) $\Delta E_m = 1$: The final example is depicted in Figure 5.17 and represents an extreme case of beat tracking improvement.

The underlying musical signal is characterised by a perfectly constant pulse and the percussive nature of the signal (thus showing clear peaks at the beat activation function). However, in its initial part, while not all rhythmic elements have been unveiled, and due to the syncopated nature and offbeat accent of the underlying rhythm, the baseline beat tracker adheres to the "wrong" peaks (phased out 180°). The finetuned approach is able to correct this behaviour, also taking advantage of both the percussive and repetitive nature of the underlying musical genre (techno).

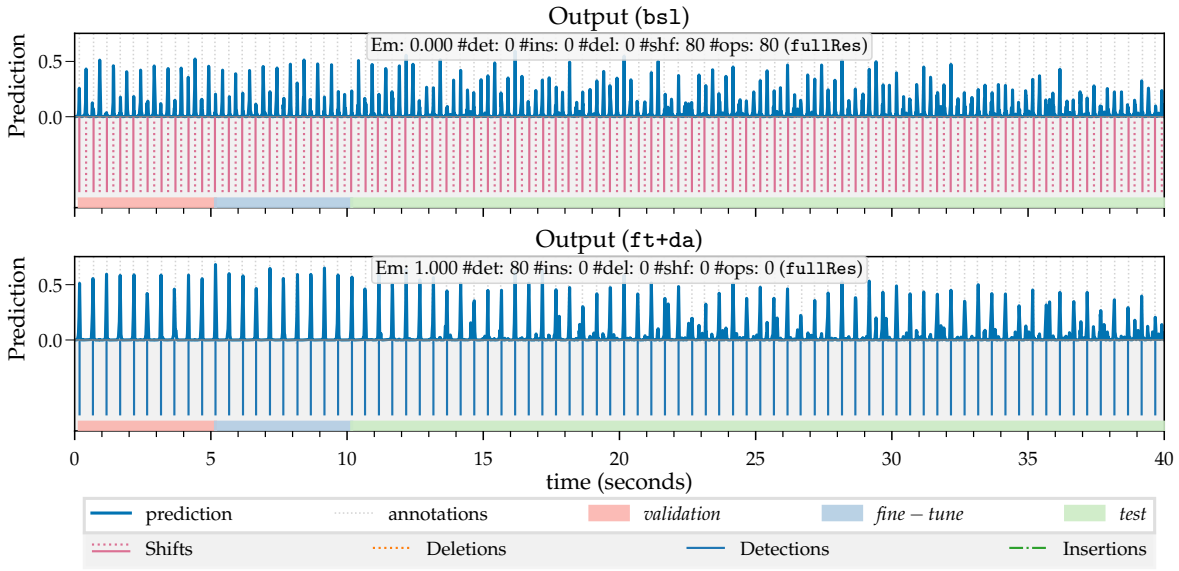


Figure 5.17: SMC_285 — *Montreal* by Autechre.

5.4 Summary

In this chapter, we conducted a comprehensive evaluation of user-centric techniques for beat tracking, adopting a general-to-specific strategy. We compared the different user-driven configurations to the baseline and assessed their performance in terms of beat tracking accuracy and required editing operations.

The results showed that the best performance was achieved by combining the base finetuning procedure with the tempo range parameterisation of the DBN. These configurations led to significant improvements in beat tracking accuracy across all datasets, with the most compelling results obtained for the *SMC* dataset, with mean

improvement ranging from 4.0 p.p.to 9.1 p.p., when contrasted to the state-of-the-art accuracy level. Furthermore, the evaluation showed that finetuning the baseline model using a small part of a music file (12.5% in the current setting) led to substantial benefits in terms of beat tracking performance. The improvements were even more noticeable in terms of the total number of editing operations required, with the best configurations allowing reductions of up to two-thirds of the corrections required by the state-of-the-art beat-tracker output.

In the second section, we explored the theoretical maximum limit of beat tracking accuracy improvement that our approach can achieve, focusing on the most challenging dataset: *SMC*. By employing a greedy algorithm to find the best configuration based on the F-measure value, we demonstrated that the optimal set of choices could enable an average increase of 23.1%, relative to the baseline performance. This optimal configuration selection led to a sharp decrease in the number of correction operations for both types of results, including or excluding the finetuned segment in evaluation, ensuring a better annotation correction workflow in terms of quantity and quality.

In conclusion, we conducted a qualitative analysis focusing on user-centred annotation corrections in various beat-tracking cases. This approach highlighted the primary advantages and disadvantages of our strategy. Predominantly, finetuning and the restriction of the DBN to the user-annotations tempo range emerged as highly effective. Notably, finetuning empowered users to steer the analysis towards improved performance, regardless of the presence of concurrent metrical and beat interpretations or alternative sets of beat activation function peaks. However, it did struggle with handling "no-energy" beats.

Although we have shown promising quantitative results, they should be interpreted cautiously. Some significant accuracy improvements might be attributed to straightforward corrections at the metrical level. A representative example is the final case from our qualitative analysis where the baseline beat tracker adhered to the "wrong" peaks (phased out by 180°). Meanwhile, the finetuned approach managed to rectify this behaviour, resulting in a (potentially misleading) improvement of 100 p.p.. This underlines the need for more appropriate metrics that reflect both the hierarchical nature of the beat tracking task and the user-correction workflow. Our approach to user-centric evaluation only started to address the latter part.

6

Adaptive Rhythm Analysis in Challenging Musical Contexts

The field of Music Information Retrieval (MIR) has yielded a range of tools that have become indispensable resources for musicologists, music theorists, and practitioners. However, in certain contexts, such as the domain of Computational Ethnomusicology, the applicability of MIR approaches is limited due to the bias of MIR research towards Western mainstream music [Tzanetakis, 2014]. This is particularly evident in the domain of rhythm analysis, given the foundational role rhythm plays across many cultures. These traditions present challenges to algorithms (and humans) unequipped to handle such rhythmic characteristics. Furthermore, the effectiveness of deep learning models in these contexts is often constrained by the scarcity of annotated data. Similarly, in the domain of Creative-MIR, end-users' high expectations in terms of algorithm accuracy and perceptual relevance present additional challenges. A key distinction in applying beat tracking for creative application scenarios is the presence of a specific end-user who intends to directly use the music analysis, thereby prioritizing the accurate extraction of beats for a specific piece over high mean accuracy scores across existing databases.

In this chapter, we seek to show that our approach is a general method that allows adaptability to various styles, genres, and user preferences. This versatility is essential for addressing the limitations of MIR research in mainstream music. It enables tackling multiple tasks within computational ethnomusicology including *beat tracking*, *onset detection*, and *metre analysis*, particularly in challenging scenarios characterised by rhythmic dissonance effects, such as *polyrhythms*, *polymetres* and *polytempi*.

The chapter is structured as follows: Section,6.1 focuses on rhythmic analysis in non-Western music. It explores onset detection in Brazilian *Maracatu* and beat tracking in Uruguayan *Candombe* and Colombian *Bambuco*, the polymetric characteristics of which allow us to assess the applicability of our approach to metre determination. In Section 6.2, we present an extremely challenging case of beat tracking, showcasing Steve Reich's *Piano Phase*, a minimalist composition that exemplifies *polytempo* through dynamic phase shifting between two voices. Finally, Section 6.3 provides a summary of the chapter, emphasizing the potential of adaptive rhythm analysis in tackling complex musical contexts and enriching our understanding of various musical styles.

6.1 Rhythmic Analysis in Non-Western Music

In this section, we explore computational rhythmic analysis in non-Western music by examining three unique case studies. The first study, Onset Detection in Brazilian *Maracatu*, centres on applying our finetuning approach to detect onsets amidst the intricate rhythms of this musical tradition. Through the exploration of both Inductive and Transductive transfer learning settings, the study seeks to offer further understanding of the transferability of the features learned.

The following two studies, Beat Tracking in Colombian *Bambuco* and Beat Tracking in Uruguayan *Candombe*, focus on Latin-American music genres known for their rhythmic complexity, which present a significant obstacle for computational analysis. These investigations evaluate how effective is our user-driven strategy for beat tracking within the respective contexts of Colombian *Bambuco* and Uruguayan *Candombe*. Whilst similar in structure, these case studies explore the distinct challenges associated with each genre and highlight the adaptability of our approach to diverse rhythmic complexities.

6.1.1 Onset Detection in Brazilian *Maracatu*

Maracatu de baque solto, also known as *Maracatu* "rural", is a vibrant carnival performance originating in Pernambuco, Northeast Brazil, that combines music, poetry, and dance [Bessoni e Silva, 2021]. The musical structure of *Maracatu* "rural"[43] revolves around a few rhythmic patterns, produced by an ensemble called the "terno", comprising percussive instruments such as the *mineiro*, *gonguê*, *cuica*, *tarol*, and *tambor*. The prevalent sub-genres, "marcha" and "samba", are characterised by fast-paced tempi (ranging from 165 bpm to around 180 bpm, respectively) and intricate rhythms.

In recent years, the *Maracatu* musical tradition has been the focus of a comprehensive ethnographic study examining the relationship between music, dance, health, and emotions [Baraldi, 2022]. Of particular relevance to our domain is a specific investigation centred on understanding rhythmic interactions among musicians during live performances. Within this context, our finetuning approach served as a foundation for onset microtiming analysis [Davies et al., 2020; Fonseca et al., 2021]. By adapting an existing deep-learning model trained on onset detection, the neural network was tailored to the unique characteristics of each "terno" instrument, resulting in more accurate estimations of onset locations. This formed a crucial component of a semi-automatic onset annotation pipeline, marking the first real application of our finetuning approach.

In this section, we deviate from the common research path, transitioning from application back to research, with the aim of further examining the onset detection task in *Maracatu* music. Our primary objective is to investigate and compare different transfer learning settings. The first setting is *Inductive* transfer learning (as utilised in [Davies et al., 2020]), where the base model is trained on the same target task. The second setting is *Transductive* transfer learning, where the base model is trained on a distinct task, specifically beat-tracking, and then finetuned for a different target task, in this case, onset detection. Moreover, taking advantage of the unambiguous nature of the task at hand (when compared to beat tracking), we aim to explore various retraining strategies by selectively finetuning specific layers of the neural network to the user annotations, and attempt to gain insights into the transferability of learned features.

[43] Although we will refer to it simply as *Maracatu*, it is important to note that *Maracatu de baque solto* (or *Maracatu* "rural") is distinct from *Maracatu de baque virado* (or *Maracatu* "nação"). Despite sharing some characteristics, such as their African origins and the fact that *Maracatu de baque virado* is also known as particularly challenging for downbeat tracking [Jehan, 2005], they are two distinct types of *Maracatu*, with substantially different instrumentation, practice and narrative [Santos et al., 2009].

Methodology

In Chapter 2, we established that onset detection and beat tracking are closely related tasks, with both focusing on the identification of temporal events in an audio signal. Prior to the advent of deep neural networks, numerous beat tracking algorithms employed onset detection as a pre-processing step [Ellis, 2007; Davies and Plumbley, 2007]. More recently, data-driven solutions for both tasks have showcased significant overlap in their architecture [Böck and Schedl, 2011; Schlüter and Böck, 2014].

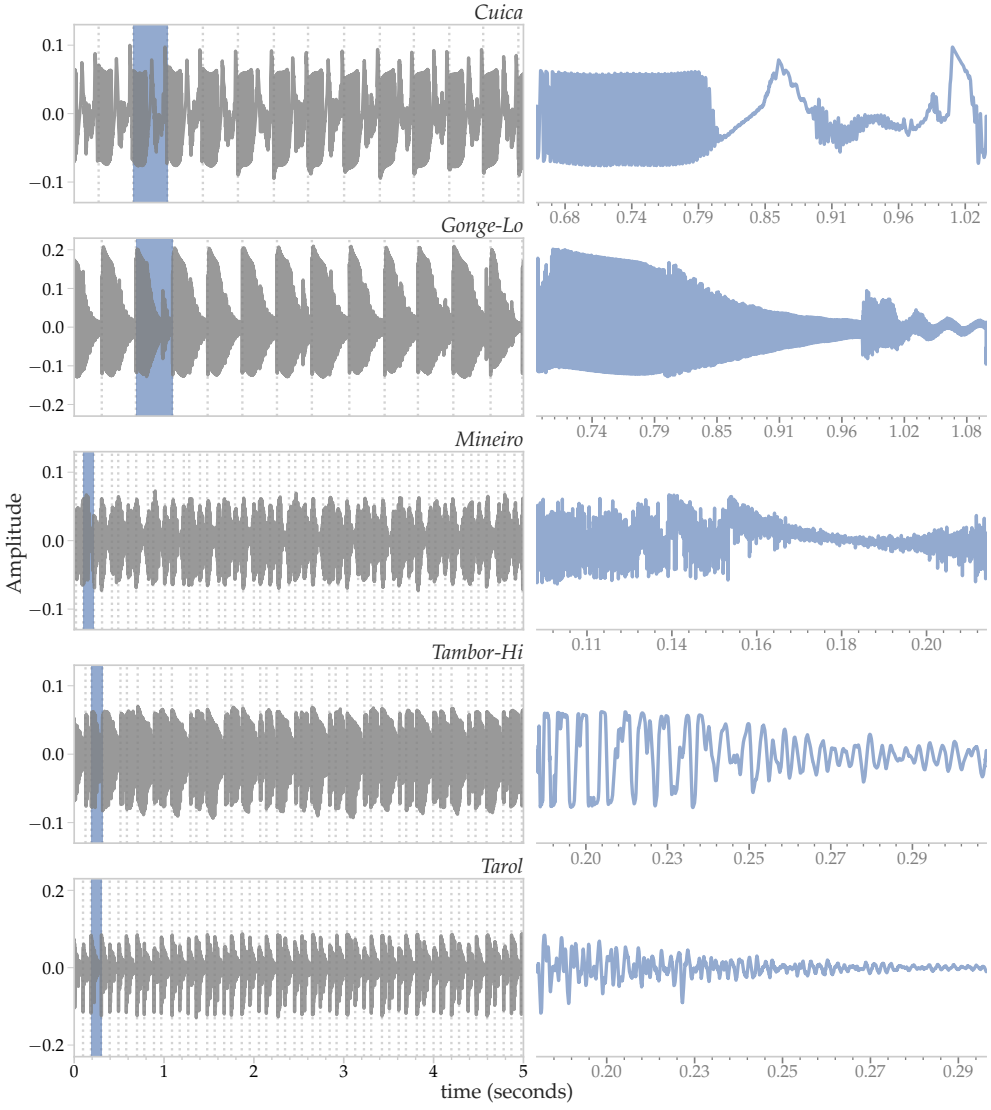


Figure 6.1: Onset-annotated waveforms 5 s snippets for the *Maracatu* subdatasets: *Cuica*, *Gongue-Lo*, *Tarol*, *Mineiro* and *Tambor-Hi*. Left: Finetuning snippet; Right: Zoomed in waveform, from the second onset to the sample before third onset (in *blue*).

Nevertheless, it is worth noting that onset detection can, at most, be considered a moderately subjective task [Daudet et al., 2004]. Onset detection for monophonic music

signals is generally viewed as a well-defined signal processing task (i.e., determining the starting points of all musically relevant events in an audio signal). Yet, it can become more challenging in other contexts, such as with polyphonic music (e.g., arpeggiated chords) or bowed string instruments (e.g., overlapping notes on different strings).

In contrast, beat tracking necessitates an understanding of the dynamic behaviour of rhythmic hierarchy over time, involving a greater degree of subjectivity. As a result, beat tracking can be considered multidimensional, requiring analysis of various aspects of the music audio signal to extract underlying rhythmic information. While beat tracking necessitates recalibrating an existing model to accommodate a variety of musical properties in an unknown piece, onset detection, particularly in the context of monophonic music signals, presents a more straightforward objective: adapting an existing network to a specific instrument, tailoring it to the individual signal characteristics (as illustrated in Figure 6.1).

Considering the context of onset detection, our methodology becomes more streamlined. We employ a simpler peak-picking process to extract the sequence of detected onsets from the onset activation function. In consequence, we eliminate all configurations related to DBN post-processing (`tg`, `pt`, and combinations). Additionally, we exclude the use of data augmentation (`da`), as its purpose of increasing the network's exposure to a wider range of tempi (and consequently beat and downbeat information) is detrimental for onset detection.

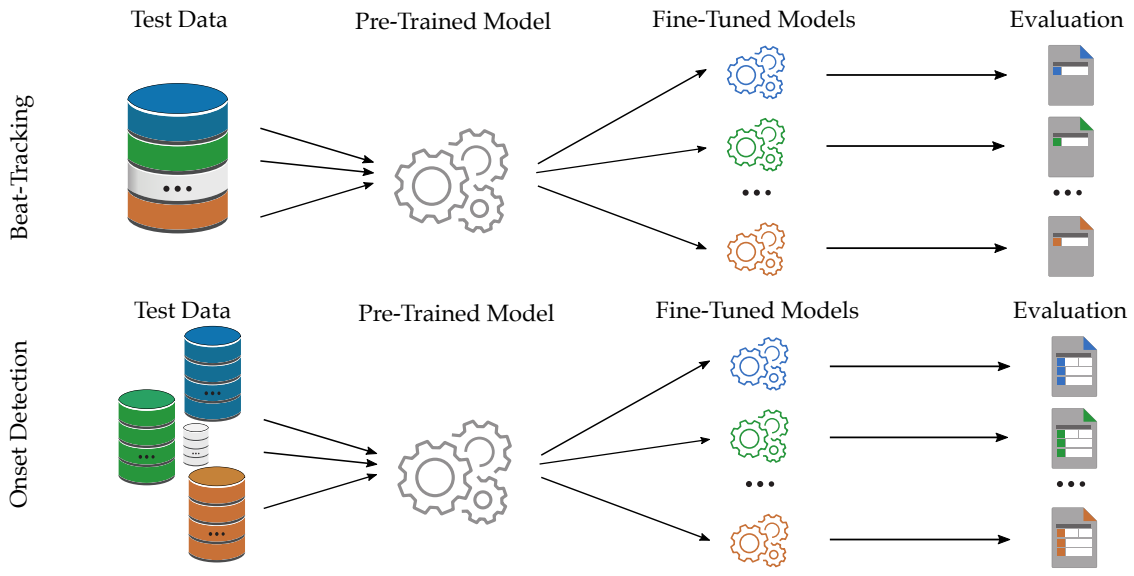


Figure 6.2: Beat tracking *vs* onset detection evaluation.

Moreover, a shift in the evaluation paradigm is evident as we transition from beat-tracking to onset detection (see Figure 6.2). As the focus changes from adapting to the musical properties of each audio file to tailoring the model to a specific instrument, finetuning is performed on a per-subdataset (i.e., per-instrument) basis.

The *Maracatu* dataset is a multi-instrument collection created by utilising contact microphones attached to each instrument of the "terno" during a studio performance [Davies et al., 2020]. In Table 6.1, we display its composition in terms of the instruments and the onset annotations for each instrument of the "terno".

The distinct functions of each "terno" instrument in the rhythmic texture of *Maracatu* become apparent through the varying number of onset annotations. Time-keeping instruments, such as *cuica* and *gonge-lo*, provide greater stability and periodicity but are less frequently present in the audio signal. In contrast, *tarol*, *mineiro*, and *tambor-hi* serve as "voicing" instruments, displaying higher pervasiveness and expressiveness.

Table 6.1: Composition of the *Maracatu* dataset.

Sub-Dataset	**# Anns**	**# Files**	**Type**	**Full Dataset** (mm:ss)	**Mean/File** (mm:ss)	**Min/File** (mm:ss)	**Max/File** (mm:ss)
Instrument		33*	Variable	30:10	00:56	00:24	01:46
Tarola	18,585						
Cuica	4,594						
Gongue-Lo	4,723						
Mineiro	17,918						
Tambor-Hi	13,375						

** The original dataset [Davies et al., 2020] contains 34 files per instrument. Due to the existence of a corrupted audio file (*Mineiro_34*), we have excluded the corresponding (i.e. *Instrument_34*) file for all sub-datasets. Furthermore, the first file of each sub-dataset (*Instrument_01*) was used for finetuning, and as such, it was also excluded from final analysis.

Another significant aspect to consider is the distinct waveform shape of the *mineiro*, which has complicated its annotation and ultimately led to its exclusion from Davies et al. [2020] work. As a result, the annotations for this instrument are less accurate than those for the others. Furthermore, as observed during our experimentation, some of the underlying audio includes non-informative data, such as extended periods with virtually no onsets, which could potentially mislead the finetuning process. Consequently, while we include the *mineiro* in our analysis for a general understanding, we refrain from conducting detailed examination or interpretation due to the imprecision of these annotations.

In the following two experimental scenarios, we explore:

1. **Inductive** Transfer Learning: in this experiment, we aim to test the finetuning of an onset detection model for the same task. We utilise a modified version of the TCN model from Davies and Böck [2019] (with an additional 11th dilation rate level), train it from scratch for onset detection on a known dataset [Böck et al., 2012], and then finetune it (for the same target task) on a dataset with a different data distribution. This model is referred to as `TCNv1`.

2. **Transductive** Transfer Learning: in this experiment, we alter the target domain. We employ the TCN model from Böck and Davies [2020] (as depicted in Figure 4.2), which is used throughout the rest of this thesis. By masking its tempo and downbeat loss (effectively converting it into a single-task beat network), we finetune it for the onset detection task. This model is referred to as `TCNv2`.

In practical terms, though, the accessibility to the corresponding base models dictated the use of slightly different network architectures for both scenarios. Although the precise network architecture is not a crucial aspect of our experimental scenario, we summarise the main characteristics of each network in Table 6.2: they are very similar, with the differences lying mainly in the convolutional block, yet resulting in very distinct network sizes: `TCNv1` having 21,890 parameters, while `TCNv2` has 116,302 parameters.

In both scenarios, the finetuning training parameterisation was identical to our main approach for the beat-tracking task (as presented in Section 4.2): we followed common practice in transfer learning and reduced the learning rate to one fifth of the rate used in the source domain training. We set the number of epochs to 50 and reduced the learning rate by a factor of 2 when there was no improvement in the loss for at least 5 epochs. The optimisation techniques were consistent with those used in pre-training[44].

To obtain the baseline onset estimates, we applied a standard peak-picking algorithm [Böck et al., 2012] to both TCN models' output activation functions: the beat activation function for the transductive scenario and the onset activation function for the inductive scenario. Similarly, for each configuration, we obtained the sequence of detected events by applying the same peak-picking algorithm to the corresponding

[44] Due to a technical issue with the `Tensorflow` implementation for the *macOS* M1 processor, the same optimisation techniques used in the original training were employed, instead of adopting more advanced optimisers available during the finetuning phase.

Table 6.2: Overview of network parameterisation and training optimisation.

	TCNv1	TCNv2
Signal Conditioning		
Audio sample rate	44.1 kHz	
Window type	*Hann*	
Window and FFT size	2048 samples	
Hop size	10 ms	
Filterbank freq. Range	30...17,000 Hz	
Sub-bands per octave	12	
Total number of bands	81	
Convolutional Block		
# filters	16, 16, 16	20, 20, 20
Filter size	3x3, 3x3, 1x8	3x3, 1x12, 3x3
Max. Pooling size	1x3, 1x3, –	1x3, 1x3, 1x3
Dropout rate	0.1	0.1
Activation function	ELU	ELU
TCN		
# stacks	1	
Dilations	11	
Number of filters	16	
Filter size	5	
Spatial dropout rate	0.1	
Activation function	ELU	
Base Training		
Optimiser	Adam	Rect.Adam + Lookahead
Learning rate	0.001	0.002
Batch size	1	1
Output activation function	sigmoid	sigmoid
Loss function	binary cross-entropy	binary cross-entropy

finetuned model's onset activation function. For evaluation purposes, we used the madmom default tolerance window of 25 ms [Böck et al., 2012].

Layer–wise Finetuning

As previously introduced, we leverage the straightforward nature of onset detection (in contrast to beat tracking) to conduct more extensive experimentation on the transferability of features in music-related tasks.

It is well-established that as the layers of a neural network deepen, the learned features become more abstract [Karpathy et al., 2015; Zhou et al., 2019]. Based on this rationale, optimising a network for a specific instrument might be achieved by recalibrating only the layers closest to the musical surface and the indispensable final

output layer, as suggested in [Davies et al., 2020]. To further investigate this concept, we primarily focus on the shallower convolutional layers and also examine the freezing of various groups of network layers, extending from these convolutional layers to the deeper TCN dilation levels (as depicted in Table 6.2). It is important to consider that these layers, which represent increasingly wider receptive fields, could potentially provide critical information for localised onset detection within specific time ranges, and this aspect will also be investigated during our study.

To represent the range of frozen layers, we use the following notation: $\texttt{ft}_{\texttt{A-B}}$, with A and B denoting the initial and final frozen layers. Since the final output layer always remains unfrozen (i.e., its weights are consistently updated through backpropagation), the deepest possible frozen layer is the one preceding the output layer and is thus excluded from the notation. Consequently, we have two primary cases: `ft` signifies a configuration where all network layers are unfrozen, and $\texttt{ft}_{\texttt{A}}$ represents a configuration in which all network layers are unfrozen except for the region between A and the last layer before the output layer, inclusive. For instance, $\texttt{ft}_{\texttt{Conv2}}$ corresponds to freezing all the layers in the network from the second convolutional layer (`Conv2`) up to, but not including, the final output layer.

In summary, we examine the following layer groupings: $\texttt{ft}_{\texttt{Conv1...3}}$, $\texttt{ft}_{\texttt{Tcn1...1024}}$, as well as the base finetuning configuration `ft` and baselines (`bsl` and `bsl*` for the inductive and transductive scenarios). Although we investigate 15 finetuning layouts, only the most relevant ones (`ft` and $\texttt{ft}_{\texttt{Conv1}}$ through $\texttt{ft}_{\texttt{Conv3}}$) are depicted in the main body of this document. The performance of the remaining configurations ($\texttt{ft}_{\texttt{Tcn1}}$ through $\texttt{ft}_{\texttt{Tcn1024}}$) can be found in Appendix A.3.1.

Inductive Transfer Learning Results

Figure 6.3 depicts the F-measure obtained by each finetuning configuration in comparison to the baseline. The results can be grouped by the rhythmic role played by the instruments: time-keeping *vs.* voicing.

For the time-keeping instruments (*cuica* and *gongue-lo*), the baseline displays an average performance, with roughly 0.5 for both instruments. However, performance improvement to values in the 0.8–1.0 region is observed with the finetuned configurations. In contrast, the expressive instruments (*tarol*, *mineiro*, and *tambor-hi*) exhibit higher initial F-measure values, approximately in the 0.9–1.0 range, which reduces the margin for relative improvement. The disparate baseline accuracy can be justified by the more conventional nature of the *tambor-hi* and the *tarol*, and their close similarity

to the baseline training material, as opposed to the *cuica* and *gongue-lo*. On the other hand, we cannot draw any conclusions about *mineiro*, which seems to contradict this observation with its clearly unusual waveform shape, due to the lower precision of its annotations. Regarding the final finetuned performance, we observe that adaptation benefits all instruments, as the best finetuned configuration consistently outperforms the baseline. However, this improvement is more pronounced for time-keeping instruments due to their lower initial accuracy. Intuitively, one might argue that it is easier to detect more distributed onsets than those condensed in time, even if we are far from the network's temporal resolution (10 ms). This is the case for the *cuica* and *gongue-lo*, which have sparser signals when compared to the *tarol*, *mineiro*, and *tambor-hi*.

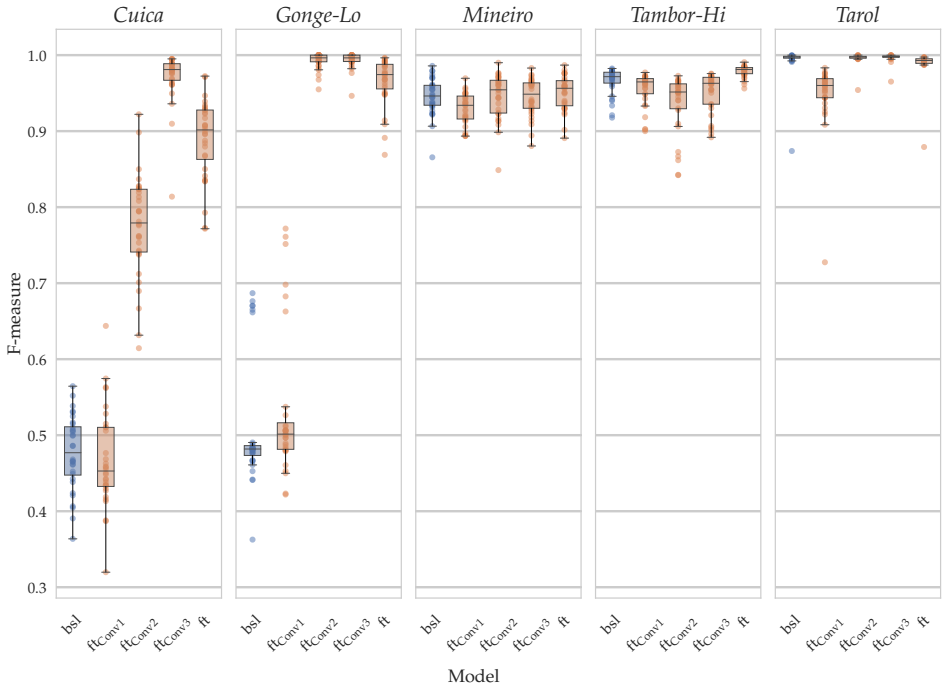


Figure 6.3: Distribution of F-measure scores by model configuration for the *Maracatu* datasets (*Inductive* Transfer Learning).

The improvements achieved through finetuning vary depending on the specific retraining configuration. The best-performing configurations are $\texttt{ft}_{\texttt{Conv3}}$ (optimal for *cuica*, *gongue-lo*, and *tarol*) and the `ft` configuration (optimal for *mineiro* and *tambor-hi*). This observation highlights that the optimal set of frozen layers differs among instruments. For instance, while the `ft` configuration, representing a fully unfrozen network, is ideal for *tambor-hi*, it does not perform as well for *cuica* or *gongue-lo*. As expected, given the absence of unfrozen layers close to the musical surface, the $\texttt{ft}_{\texttt{Conv1}}$ configuration lags behind other finetuned configurations for most instruments. In

some situations, its performance is even inferior to the baseline.

For the *Cuica* dataset, the $\texttt{ft}_{\texttt{Conv3}}$ model excels with the highest F-measure mean of 0.971, marking a substantial leap from the baseline (`bsl`) which has an F-measure mean of 0.477—an improvement of almost 50 p.p.. Meanwhile, in the *Gonge-Lo* dataset, the top F-measure of 0.993 is shared between $\texttt{ft}_{\texttt{Conv2}}$ and $\texttt{ft}_{\texttt{Conv3}}$ models, showing a marked improvement over the baseline's F-measure of 0.508. This indicates that both finetuned models strike a harmonious balance between Precision and Recall in this dataset.

Our findings provide a new perspective on the discussion about the impact of layers that need finetuning, particularly in relation to our initial rationale. We initially believed that to optimise a network for a specific instrument, only the layers closest to the musical surface, in addition to the final output layer, require recalibration. The results present a more nuanced picture. Indeed, when we keep the shallow layers frozen (e.g., $\texttt{ft}_{\texttt{Conv1}}$), there is a clear underperformance. However, the benefits of unfreezing the shallowest layer are not as straightforward as initially thought. The configuration $\texttt{ft}_{\texttt{Conv2}}$, which allows updating the weights of the first convolutional layer (`Conv1`) (and the final output layer), does not correspond to the best configuration for any instrument. Furthermore, when we extend our analysis to the complete set of results, depicted in Table A.3[45], the best results across 4-in-5 instrument-adapted networks come from configurations that encompass finetuning of some or all TCN dilation levels. These observations suggest that the optimal finetuning strategy may be more complex than simply focusing on the layers closest to the musical surface.

Additional key observations include the optimal freeze configuration being instrument-dependent, and that unfreezing TCN layers is generally beneficial (except for the *tarol*). For voicing instruments, full-network finetuning (`ft`) ranks among the top-performing configurations, while it worsens time-keeping instruments' analysis. Lastly, the baseline consistently exhibits higher Recall, suggesting that finetuned models are more conservative in their predictions, leading to fewer true positives being detected.

Transductive Transfer Learning Results

In this exploratory study, we aim to briefly examine a novel transductive scenario that, to the best of our knowledge, has not been previously investigated: the knowledge

[45] Presenting a more detailed set of results, including mean values for F-measure, Precision, Recall, and counts of True Positives (TP), False Positives (FP), and False Negatives (FN) for all configurations, can be found in Appendix A.

transfer from beat tracking to onset detection. In this section, we summarise the main general outcomes and results, as depicted in Figure 6.4.

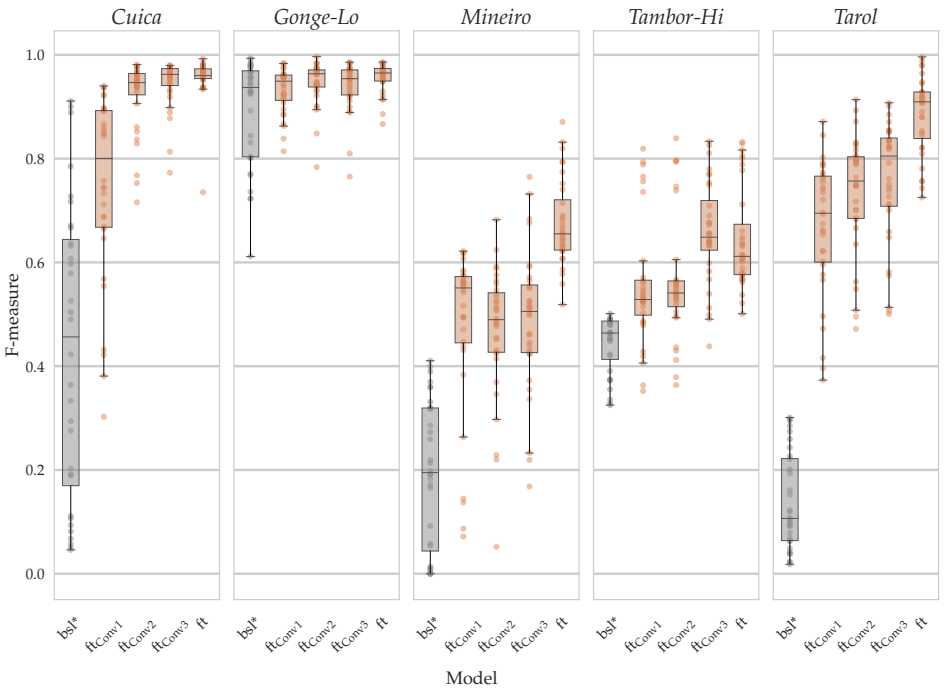


Figure 6.4: Distribution of F-measure scores by model configuration for the *Maracatu* datasets (*Transductive* Transfer Learning).

The finetuned configurations display varying degrees of accuracy for the diverse "terno" instruments. Time-keeping instruments, such as *cuica* and *gongue-lo*, have higher baseline (`bsl*`) accuracy since their onsets tend to align with the beats. Adaptation benefits all instruments, which enables us to assert that transduction is successful in adapting a beat tracking model through finetuning for high accuracy in onset detection.

However, comparing these values to the baseline (`bsl*`) is not particularly meaningful due to their distinct targets. Similarly, we do not delve deeply into the mean F-measure values across datasets because of their limited significance (these values can be found in Table A.4). Our primary focus is to determine whether this setting allows us to obtain values comparable to those in the previous inductive setting. To facilitate this assessment, we concentrate on a direct comparison summarised in Table 6.3.

The inductive setting yields the highest F-measure scores across all instruments, as expected due to the base network being (pre)trained on the same target — onset detection. However, in the transductive setting (which features a considerably larger network), some of the best-performing models manage to achieve not only higher accu-

racies than the inductive baseline (bsl), but also comparable results to the top models in the inductive setting. For time-keeping instruments, *cuica* and *gonge-lo*, performance is almost identical across both transfer learning scenarios, with differences of 1.6 p.p. and 3.7 p.p., respectively. On the other hand, voicing instruments exhibit progressively larger disparities in F-measure values: 11.3 p.p. for *tarol*, 27.5 p.p. for *mineiro*, and the largest difference of 32.2 p.p. for *tambor-hi*. The latter two instruments do not achieve comparable scores to the inductive baseline, indicating that the transferability of features between the two tasks is not as effective in these cases.

Table 6.3: Summary comparison of the three-best performing models (according to F-measure mean scores) for inductive and transductive transfer learning scenarios.

	Inductive Transfer Learning			*Transductive Transfer Learning*		
Dataset	**Best Model**	**F-measure**	**Baseline**	**Best Model**	**F-measure**	**Baseline**
Cuica	$\texttt{ft}_{\texttt{Tcn16}}$	**0.985**		ft	0.955	
	$\texttt{ft}_{\texttt{Conv3}}$	0.971	0.477	$\texttt{ft}_{\texttt{Tcn16}}$	0.948	0.429
	$\texttt{ft}_{\texttt{Conv2}}$	0.775		$\texttt{ft}_{\texttt{Tcn512}}$	0.952	
Gonge-Lo	$\texttt{ft}_{\texttt{Tcn2}}$	**0.998**		ft	0.956	
	$\texttt{ft}_{\texttt{Conv2}}$	0.993	0.508	$\texttt{ft}_{\texttt{Tcn2}}$	0.944	0.892
	$\texttt{ft}_{\texttt{Conv3}}$	0.993		$\texttt{ft}_{\texttt{Tcn512}}$	0.952	
Mineiro	$\texttt{ft}_{\texttt{Tcn16}}$	**0.972**		$\texttt{ft}_{\texttt{Tcn8}}$	0.790	
	$\texttt{ft}_{\texttt{Conv3}}$	0.945	0.946	$\texttt{ft}_{\texttt{Tcn1024}}$	0.953	0.193
	$\texttt{ft}_{\texttt{Tcn8}}$	0.958		$\texttt{ft}_{\texttt{Tcn4}}$	0.774	
Tambor-Hi	ft	**0.978**		$\texttt{ft}_{\texttt{Tcn1}}$	0.723	
	$\texttt{ft}_{\texttt{Conv3}}$	0.951	0.965	$\texttt{ft}_{\texttt{Tcn2}}$	0.708	0.443
	$\texttt{ft}_{\texttt{Tcn16}}$	0.968		$\texttt{ft}_{\texttt{Tcn512}}$	0.637	
Tarol	$\texttt{ft}_{\texttt{Conv3}}$	**0.997**		ft	0.884	
	$\texttt{ft}_{\texttt{Conv2}}$	0.996	0.993	$\texttt{ft}_{\texttt{Tcn1024}}$	0.848	0.139
	$\texttt{ft}_{\texttt{Conv1}}$	0.949		$\texttt{ft}_{\texttt{Tcn512}}$	0.831	

We underline that the networks used in these two settings have significantly different sizes: TCNv1 having 21,890 parameters, while TCNv2 has 116,302 parameters. Furthermore, it is important to note that although both networks share the same layer names, they effectively represent different temporal (and frequency) receptive fields, as illustrated in the auxiliary table A.5. This factor should be carefully considered when interpreting the results.

Indeed, as previously observed, when certain timbres are learned during the base training, their initial performance tends to be higher. However, it is intriguing to see that in some cases, the starting baseline is higher in the transductive setting than in the

inductive setting, such as with the *cuica* and particularly the *gongue-lo*. The range of known temporal periods may play a crucial role in this behaviour, as the transductive baseline (`bsl*`) has been *primed* for a beat tracking task, which involves different timings compared to those of onset detection. Thus, in this context, the baseline (`bsl`) is effectively tracking the beat on the *maracatu* signals with tempi in the 165 bpm–180 bpm vicinity (for "marcha" and "samba" sub-genres, respectively), corresponding to an inter-beat interval of roughly 333–363 ms. This range aligns closely with the temporal extent of both *cuica* and *gongue-lo* waveforms, but not with the remaining instruments[46]. It is important to note, however, that given the exploratory nature of this study, the preceding considerations are highly speculative in nature and would require further investigation.

Similar to the previous analysis, a more comprehensive set of results can be viewed in Table A.4. From these observations, we can identify some patterns given the wider improvement range of voicing instruments. The accuracy increases as more layers are finetuned up to the 3rd or 4th dilation level, after which no further enhancements are observed when including deeper layers. This pattern, however, does not hold for the *tarol*, where retraining deeper levels results in better performance.

Discussion

In this study, we have explored the efficacy of transfer learning in the context of musical onset detection, focusing on *Maracatu de baque solto*. We have systematically evaluated the finetuning of different groups of layers to bring more information to this setting, building upon the successful finetuning approach in our previous study on inductive transfer learning [Davies et al., 2020]. Additionally, we have investigated a novel transductive transfer learning scenario, which, to the best of our knowledge, has not been previously addressed in the context of beat tracking and onset detection.

From our analysis of inductive transfer learning, we observe that adaptation benefits all instruments, as the best finetuned configuration consistently outperforms the baseline. The improvement is more pronounced for time-keeping instruments (*cuica* and *gongue-lo*). The optimal set of frozen layers differs among instruments, and we find that unfreezing TCN layers is generally beneficial.

In the transductive transfer learning setting, we find that adaptation is effective for all instruments. Time-keeping instruments exhibit almost identical performance across

[46] These specific values are based on an informal inspection of instruments' waveforms that led us to the following approximate temporal spans: *Cuica*: 384–428 ms; *Gonge-Lo* - 376–400 ms, *Tarol*: 77–107 ms, *Mineiro*: 90–180 ms and *Tambor-Hi*: 120–230 ms.

both transfer learning scenarios, while voicing instruments show progressively larger disparities in F-measure values, such as for *mineiro* and *tambor-hi*.

From these findings, we can draw several insights:

- Transfer learning proves to be an effective approach for onset detection in *Maracatu* rhythms, providing improvements in accuracy across both time-keeping and voicing instruments.

- Finetuning is essential for achieving optimal performance in both inductive and transductive transfer learning settings. However, the optimal finetuning configuration depends on the specific instrument, justifying a tailored strategy to the choice of the set of layer weights to be updated during finetuning;

- In the transductive scenario, the contrasting baseline accuracies suggest that the transfer of knowledge between beat tracking and onset detection tasks might be influenced by the temporal periods of the instruments. Further investigation is needed to better understand this behaviour.

As a closing remark, it is crucial to recognise the exploratory nature of this study and the inherent limitations that accompany it. The results presented were achieved with only minimal adjustments to several aspects of the experimental pipeline, including the training optimisation parameters. Moreover, we have not fully explored all available results, and additional experimentation would be advisable for a deeper understanding of certain aspects. This experimental study has been inspired by previous works on microtiming analysis conducted on the *Maracatu* dataset [Davies et al., 2020; Fonseca et al., 2021], and we encourage readers to consult these studies for a more detailed analysis[47].

[47] In these studies, the authors build upon our finetuning approach, incorporating time-correction for onset locations to address the TCN's 10 ms temporal resolution. They concentrated on a single finetuning configuration, in which all layers, except for the shallowest and the last output layer, were frozen (corresponding to our $\texttt{ft}_{\texttt{Conv2}}$ configuration). Focusing on microtiming analysis, they conducted an in-depth assessment of onset detection accuracy using tolerance windows ranging from ± 1 ms up to ± 25 ms, in 1 ms increments.

6.1.2 Beat Tracking in Colombian *Bambuco*

Bambuco is a Colombian traditional music genre well-known for its rhythmic complexity produced by heavy syncopation and odd accents, on top of a certain degree of rhythmic freedom in the form of tempo variations and micro-timing [Cano et al., 2021]. Some of these characteristics are presented concisely in Figure 6.5. Its most distinctive feature is arguably its polymeter rhythm, which arises from the superposition or alternation of musical elements in two meters: a simple metre (3/4) and a compound one (6/8). This phenomenon is known as "hemiola"[48] (or by the equivalent Latin term "sesquialtera"[49]), and despite being relatively common in other south-American musical genres [Schechter et al., 1985], has helped to establish this genre as a challenging case for metre and beat-tracking computational analysis.

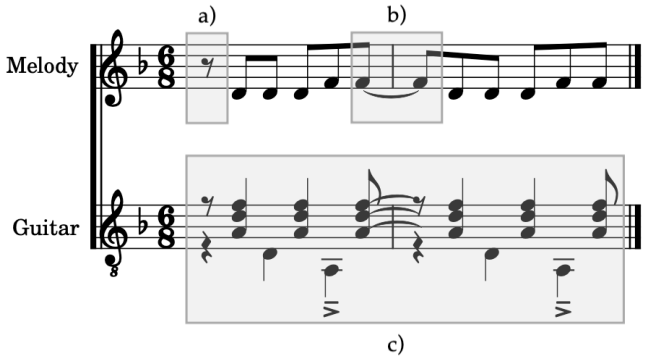


Figure 6.5: Colombian *Bambuco* example. a) downbeat in a rest; b) caudal syncopation; and c) guitar accompaniment pattern that suggests 6/8 at the top voice and 3/4 at the bass voice; (adapted from Cano et al. [2021]).

As illustrated by Figures 6.6 and 6.5, the beats' locations do not align, with the exception of the downbeat. This indicates a close relationship between the tasks of metre analysis and beat tracking. Essentially, it implies that we can deduce the metric interpretation from the placement of the beats. These properties make *Bambuco* an ideal case for assessing the effectiveness of our user-driven approach. More specifically, while our approach primarily targets beat tracking, it also informs metre analysis due to the interconnected nature of these rhythmical facets.

[48] the Greek term for the ratio 3/2.

[49] Cano et al. emphasise the lack of consensus among authors regarding specific definitions for rhythmic behaviours such as *birhythmia*, *sesquialtera*, and *hemiola*. In our research context, we use *sesquialtera* and *hemiola* interchangeably, without differentiating between vertical rhythmic superpositions and rhythmic alternations.

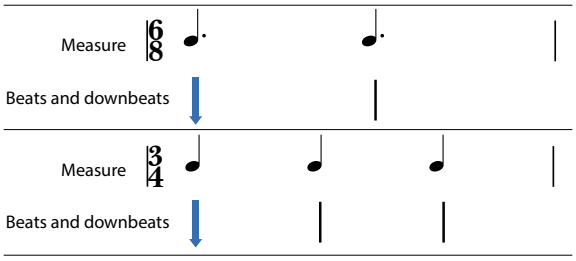


Figure 6.6: Polymeter structure in Colombian *Bambuco*, with the hierarchical relationship between metre, measures, and beats. In both 3//4 and 6//8, the beats are indicated with vertical lines, and the downbeat with a blue arrow. (adapted from Cano et al. [2021]).

Methodology

The dataset we use is the "Rhythm Set" of the ACMUS-MIR (V1.1) [Mora-Ángel et al., 2019]. It is composed of more than 70 short musical excerpts, with two different sets of ground-truth annotations corresponding to the two predominant meters[50]: the simple 3/4 and the compound 6/8. In this section, these will be referred to as *Bambuco (simple)* and *Bambuco (compound)*, respectively.

Table 6.4: Composition of the *Bambuco* dataset.

Dataset	# Files	Type	Full Dataset (mm:ss)	Mean/File (mm:ss)	Min/File (mm:ss)	Max/File (mm:ss)
Bambuco	73	Variable	00:20:59	00:17	00:07	00:28

The rest of our methodology aligns with the approaches used in Chapter 5, briefly summarised as follows. The finetuning region corresponds to 25% of each file's duration. The results shown are averaged over three complete finetuning iterations. Our focus remains on the primary set of configurations, highlighting the contrast between the baseline and the primary finetuned model (`ft+da`), as well as the DBN parameterisation configurations (`ft+da+pt`, `ft+da+tg`, `ft+da+tg+pt`). Finally, the results are presented in two sets: the first includes the finetuned segment of the input signal (`fullRes`), and the second excludes it (`testRes`).

Results

The results shown are averaged over three complete finetuning iterations. All the main finetuning configurations showed some improvement in accuracy compared to the

[50] Mora-Ángel et al. acknowledge that defining a unique metre in the *Bambuco* genre can be difficult or incorrect. As a result, beat annotations for the dataset were conducted separately for the two predominant meters.

baseline (`bsl`): moderate for the *Bambuco–simple* dataset, and more noticeable for the *Bambuco–compound* dataset.

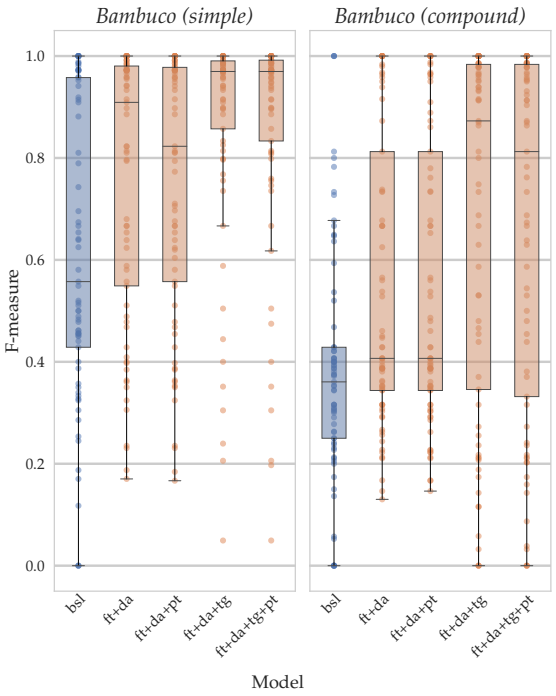


Figure 6.7: Distribution of F-measure scores by model configuration for the *Bambuco* datasets.

We first consider the `testRes` results quantitatively (as depicted in Table 6.5), where we exclude the finetuned region from the analysis. For the simple metre dataset, the average F-measure improvement reaches nearly 24.7p.p. for the best performing configuration (`ft+da+tg`), up to a score of 0.868. Similar benefits can be observed across the "looser" metrics (i.e. the AMLc and AMLt), whereas the "stricter" ones demonstrate even greater accuracy improvements — ranging from 38.3p.p. in the case of CMLc to 45.0p.p. for CMLt.

Lower starting accuracies are observed in the compound metre dataset, as expected given the under-representation of this type of metre in the baseline training datasets. In contrast, there is more room for performance improvement, realised through accuracy enhancements that range from a minimum of 29.8p.p. for the F-measure to a maximum between 50.2p.p. and 53.4p.p. (corresponding to approximately 450%) for the metrics that require correct metrical levels (CMLc and CMLt), for the same best-performing configuration `ft+da+tg`. The key to this improved performance appears to be the combination of techniques, as their isolated contributions (`ft`, `pt` and `tg`) are rather modest when compared to their combined performance.

Table 6.5: Mean of the F-measure, CMLc, CMLt, AMLc and AMLt scores across the *Bambuco (simple)* and *Bambuco (compound)* datasets for the various configurations. (`testRes`).

Dataset	Model	F-measure	CMLc	CMLt	AMLc	AMLt
Bambuco (simple)	bsl	0.621	0.319	0.366	0.631	0.692
	ft+da	0.760	0.511	0.591	0.691	0.773
	ft+da+pt	0.752	0.496	0.574	0.683	0.763
	ft+da+tg	**0.868**	0.702	**0.831**	0.734	0.862
	ft+da+tg+pt	0.866	**0.706**	0.830	**0.740**	**0.862**
	ft	0.722	0.446	0.528	0.688	0.779
	pt	0.610	0.312	0.354	0.635	0.693
	tg	0.785	0.597	0.746	0.665	0.821
Bambuco (compound)	bsl	0.383	0.114	0.114	0.262	0.276
	ft+da	0.546	0.297	0.311	0.404	0.419
	ft+da+pt	0.547	0.303	0.316	0.401	0.416
	ft+da+tg	**0.681**	**0.616**	**0.648**	**0.657**	**0.714**
	ft+da+tg+pt	0.665	0.594	0.627	0.635	0.695
	ft	0.482	0.198	0.206	0.313	0.324
	pt	0.382	0.113	0.113	0.261	0.278
	tg	0.368	0.301	0.319	0.445	0.544

Table 6.6: Mean of the E-measure score and sum of the #det, #ins, #del, #shf and #ops scores across the *Bambuco (simple)* and *Bambuco (compound)* datasets for the various configurations. (`testRes`).

Dataset	Model	E-measure	#det	#ins	#del	#shf	#ops
Bambuco (simple)	bsl	0.554	1,266	810	*29*	336	1,175
	ft+da	0.714	1,740	450	50	*222*	722
	ft+da+pt	0.704	1,725	441	50	247	737
	ft+da+tg	**0.854**	**2,126**	*12*	76	276	*362*
	ft+da+tg+pt	0.852	2,115	*12*	74	285	371
	ft	0.669	1,634	521	46	256	824
	pt	0.544	1,248	803	31	361	1,195
	tg	0.772	1,878	21	79	513	613
Bambuco (compound)	bsl	0.333	643	304	284	708	1,296
	ft+da	0.499	936	211	239	506	960
	ft+da+pt	0.499	933	212	237	509	960
	ft+da+tg	**0.668**	**1,183**	*55*	37	*418*	*510*
	ft+da+tg+pt	0.650	1,159	63	*35*	432	530
	ft	0.427	833	239	296	584	1,118
	pt	0.332	643	302	290	710	1,302
	tg	0.355	642	89	39	924	1,052

Finally, we focus on the annotation-correction workflow perspective, as presented in Tables 6.6 and 6.7. In the more stringent evaluation (`testRes`), we observe increments in the *Edit-measure* (E-measure) that roughly double those shown before for the F-measure (Table 6.5). Specifically, the best performing configuration (`ft+da+tg`) for the simple and complex metre versions of this dataset (`ft+da+tg`) yield an improvement towards the baseline of 30.0 p.p. and 33.4 p.p. (corresponding to 54% and 100%), respectively. Additionally, a comparison of the number of operations required to correct the beat estimates between the same two configurations shows that a significant proportion (approximately 2/3) are eliminated through the finetuning process.

Similar findings are obtained when taking into account the full extension of the files (`fullRes`), as shown in Table 6.7. Additionally, we include the *annotation efficiency* (Ae) results, which show us that the best performing configuration (`ft+da+tg`) output requires approximately 13.9% fewer correction operations than the baseline algorithm's output, per user annotation. This suggests that the finetuning process has led to a modest improvement in the algorithm's performance, and the user's annotation effort has been reasonably well-utilised.

Table 6.7: Mean of the E-measure and Ae scores and sum of the #det, #ins, #del, #shf and #ops scores across the *Bambuco (simple)* and *Bambuco (compound)* datasets for the various configurations. (`fullRes`)

Dataset	Model	E-measure	#det	#ins	#del	#shf	#ops	Ae
Bambuco (simple)	bsl	0.556	1,756	1,110	*60*	440	1,610	–
	ft+da	0.726	2,439	602	91	*265*	957	0.631
	ft+da+pt	0.718	2,428	588	94	291	972	0.599
	ft+da+tg	**0.869**	**2,990**	*12*	138	306	*455*	**1.139**
	ft+da+tg+pt	0.866	2,978	*12*	137	316	465	1.130
	ft	0.683	2,306	703	88	297	1,088	0.473
	pt	0.544	1,726	1,102	*60*	478	1,640	–
	tg	0.768	2,582	24	139	700	863	–
Bambuco (compound)	bsl	0.338	899	424	410	947	1,781	–
	ft+da	0.509	1,319	285	340	665	1,292	0.743
	ft+da+pt	0.513	1,322	285	332	663	1,282	0.785
	ft+da+tg	**0.685**	**1,665**	*63*	62	*541*	*667*	**1.814**
	ft+da+tg+pt	0.671	1,640	73	61	557	691	1.757
	ft	0.441	1,188	322	416	759	1,499	0.383
	pt	0.335	895	422	416	953	1,791	–
	tg	0.364	898	107	*59*	1,265	1,431	–

Summary of the Results The beat tracking results for the Colombian Bambuco dataset reveal accuracy improvements with finetuning configurations when compared to the baseline. For the simple metre dataset, the average F-measure improvement reaches nearly 24.7 p.p. for the best-performing configuration `ft+da+tg`, with an absolute score of 0.868. Stricter metrics demonstrate greater accuracy improvements, with absolute scores of 0.830 for CMLc (a 38.3 p.p. improvement) and 0.838 for CMLt (a 45 p.p. improvement). In the compound metre dataset, the accuracy enhancements range through a minimum of 29.8 p.p. for the F-measure, with an absolute score of 0.798, and a maximum between 50.2 p.p. and 53.4 p.p. for CMLc and CMLt, with absolute scores of 0.835 and 0.817, respectively. The best performing configuration (`ft+da+tg`) yields E-measure improvements towards the baseline of 30.0 p.p. and 33.4 p.p. for the simple and complex metre versions, respectively, reducing the number of operations required to correct the beat estimates by approximately 2/3.

6.1.3 Beat Tracking in Uruguayan *Candombe*

Candombe, an African-origin rhythm, is prominent in Uruguay and, to a lesser extent, in other South American countries [Schechter et al., 1985]. Musically, as illustrated in Figure 6.8, it is characterised by the interplay of three percussion instruments: the *chico*, the *repique*, and the *piano*, with an additional time-line pattern called *clave*, shared by the three drums [Jure and Rocamora, 2016]. This combination produces a typical rhythmic structure consisting of a four-beat measure evenly divided into 16 tatums, typically played at a tempo of about 110–150 bpm.

Candombe distinguishes itself from other rhythms by two features that connect it to Afro-Atlantic music traditions [Nunes et al., 2015]: a) the pulse pattern emphasises the second tatum rather than the one on the beat, and b) the clave divides the 16-tatum cycle irregularly (3+3+4+2+4), with only two of its five strokes synchronised with the beat. Moreover, in actual performances, the primary pattern of *repique* leans towards a triplet feeling, and although the *chico* drum establishes the metrical foundation, its pattern is suggested to exhibit a contraction of inter-onset intervals (IOIs) [Jure and Rocamora, 2016].

These unique characteristics of *Candombe* present challenges for both untrained listeners and standard beat-tracking algorithms, making it a challenging test case for evaluating our user-driven approach.

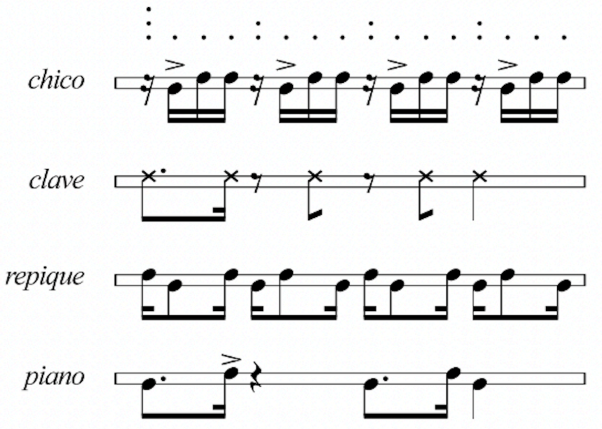


Figure 6.8: Interaction of the main *Candombe* patterns and the three levels of the resulting metric structure (adapted from Nunes et al. [2015]).

Methodology

In this study, we employ the *Candombe* dataset [Jure et al., 2015], which comprises 35 complete songs, a unique attribute in contrast to the majority of datasets that are

typically composed of short musical segments.

Table 6.8: Composition of the *Candombe* dataset.

Dataset	# Files	Type	Full Dataset (hh:mm:ss)	Mean/File (mm:ss)	Min/File (mm:ss)	Max/File (mm:ss)
Candombe	35	Variable	02:27:42	04:13	01:56	12:00

The remaining methodological elements largely mirror the prior procedures, summarised briefly in Section 6.1.2. Key aspects include maintaining the finetuning region at 25% of each file's duration, averaging results over three complete finetuning iterations, and segmenting the results into two sets: excluding and including the finetuned segment of the input signal. Furthermore, we have included the individual configurations of the user-driven techniques (`ft`, `pt`, and `tg`) in the results, recognizing their relevance to the subsequent discussion.

Results

The overall results are summarised in Figure 6.9a. Accuracy scores improve drastically for all the *main* (`ft+da`, `ft+da+pt`, `ft+da+tg` and `ft+da+tg+pt`) finetuning configurations when contrasted to the baseline (`bsl`).

Interestingly, the "simple" finetuning (`ft`) configuration stands out above the others, which is why we have included the results corresponding to the isolated use of user-driven techniques, such as `ft`, `pt`, and `tg` in Figure 6.9. As discussed in Section 5.3.1, this is the second instance where incorporating data augmentation does not appear to enhance the effectiveness of the finetuning strategy. Both *TapCorrect* and *Candombe* datasets consist of full-length files with an average duration of about 4 minutes. Given the parameterisation used in this specific experimental study (i.e., a finetuning length of 25%), the finetuning region covers a considerable amount of data, averaging over 1 minute.

These observations suggest that there might be an upper limit to the amount of data required for finetuning, beyond which further data does not yield additional benefits. The root cause could involve music signals with constant tempo, for which data augmentation is not beneficial, or musical textures with soft onsets, where data augmentation might unintentionally produce extra observations (i.e., false positives) near the actual onset (i.e., true positive).

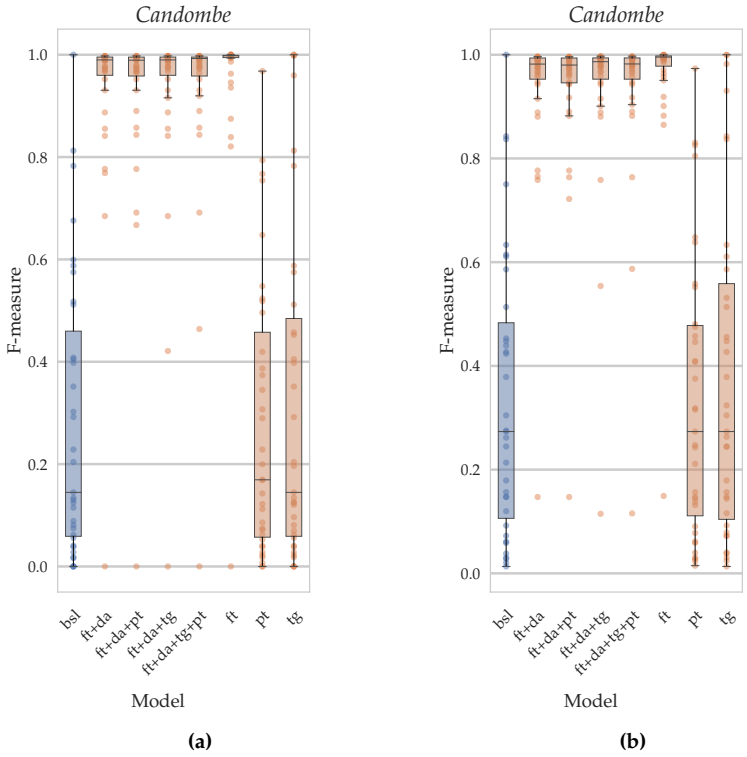


Figure 6.9: Distribution of F-measure scores by model configuration for the *Candombe* dataset: (a) `testRes` (b) `fullRes`.

Furthermore, these results show us another peculiarity: a particular file for which we obtain zero F-measure scores for all the possible configurations (depicted in Figure 6.9a). An auditory inspection of this piece ("Magarinos" by *Proyecto 1992*) revealed that it has a progressive (as opposed to repetitive) structure, meaning that the finetuning region is not repeated in any form in the rest of the file. Thus, when the finetuning region is excluded from analysis (as in Figure 6.9a), accuracy is zero ($F_m = 0$), otherwise (as in Figure 6.9b), there is very low accuracy ($0.1 \leqslant F_m \leqslant 0.2$). In effect, the benefits of finetuning are totally coupled with the existence of some sort of musical repetition in the analysed music; particularly, there is the need for reoccurrence of the musical elements of the finetuned section in the remaining audio. If so, the results are improved through finetuning; otherwise, there is no advantage, as happens in the present case.

In quantitative terms (as presented in Table 6.9), we see an improvement of the F-measure from 0.280 to 0.952 for the best-performing configuration (`ft`). Equally expressive results are obtained across all other accuracy metrics, with upswings ranging from a 60.2 p.p.(175%) for the AMLt to 74.0 p.p.(590%) for the CMLc. On all accounts, these are highly promising results, in terms of beat-tracking accuracy, especially considering that we are excluding the finetuned region from the evaluation.

Table 6.9: Mean of the F-measure, CMLc, CMLt, AMLc and AMLt scores across the *Candombe* dataset for the various configurations. (`testRes`).

Dataset	Model	F-measure	CMLc	CMLt	AMLc	AMLt
Candombe	bsl	0.280	0.125	0.264	0.168	0.344
	ft+da	0.927	0.775	0.906	0.813	0.943
	ft+da+pt	0.923	0.768	0.894	0.814	0.940
	ft+da+tg	0.921	0.783	0.922	0.783	0.922
	ft+da+tg+pt	0.923	0.787	0.924	0.787	0.924
	ft	**0.952**	**0.866**	**0.945**	**0.868**	**0.946**
	pt	0.279	0.117	0.262	0.161	0.343
	tg	0.318	0.186	0.345	0.190	0.369

If we compare our findings to those reported by Nunes et al. [2015], it becomes apparent that our approach surpasses both "general-purpose"[51] and *Candombe*–specific beat trackers. It is crucial, however, to note that these results cannot be compared directly due to the differing methodologies used. Nonetheless, these results are still relevant and warrant consideration, especially in cases where our observations contrast with other informed approaches. A recent study that reported results for this dataset was conducted by Maia et al. [2022]. A maximum F-measure of 0.996 was reported for the *Candombe* dataset using the "BayesBeat" approach. This score surpasses our maximum of 0.956 (achieved when considering the full extent – `fullRes` – of the signal), yet, a direct comparison is challenging, given the different evaluation methodologies[52].

To conclude our analysis of beat-tracking in *Candombe*, we turn to the annotation-correction workflow point-of-view. Once more, the outcomes are highly informative, as evidenced by Table 6.10: the E-measure metric attains a peak of 0.948 for the optimal configuration (`ft`). Such a high value condenses the information depicted in the specific figures: while for the baseline beat tracker generates 3,960 correct detections (#det), the finetuned beat tracker produces 12,889 accurate ones. This signifies a relevant advancement in beat-tracking performance, with an increase from approximately 28% correct detections to 90% of the test annotations in the dataset (`testRes`). As a result, the effort needed to correct the beat estimates is substantially reduced, diminishing from the 10,447 baseline correction operations (#ops) to approximately one-seventh, or

[51] The beat-tracking algorithms proposed by Ellis [2007]; Dixon [2001b]; Oliveira et al. [2012]; Klapuri et al. [2006].

[52] In the same study, a F-measure of 0.995 is reported in the best scenario for the `TCN-FT/A` configuration. In technical terms, this equates to our initial finetuning approach (as reported in Chapter 4), where Böck and Davies [2020] TCN is retrained using a 10 s annotated region, with a 50% split between finetuning and validation goals. However, unlike our per-file finetuning, this approach uses groups of 10 s cross-dataset snippets for generalised adaptation. These notable results support an even more streamlined dataset-wide annotation approach, further validating the effectiveness of our base method.

Table 6.10: Mean of the E-measure score and sum of the #det, #ins, #del, #shf and #ops scores across the *Candombe* dataset for the various configurations. (`testRes`).

Dataset	Model	E-measure	#det	#ins	#del	#shf	#ops
Candombe	bsl	0.267	3,960	2,400	28	8,019	10,447
	ft+da	0.915	12,391	1,504	140	484	2,128
	ft+da+pt	0.910	12,285	1,614	135	480	2,229
	ft+da+tg	0.915	12,521	*1,227*	144	631	2,001
	ft+da+tg+pt	0.917	12,537	*1,227*	141	614	1,982
	ft	**0.948**	**12,889**	1,304	44	*187*	*1,534*
	pt	0.267	3,967	2,359	34	8,053	10,446
	tg	0.317	4,894	1,296	*12*	8,189	9,497

more precisely, 1,534 annotation-corrections for the finetuned beat tracker.

When taking into account the full extension of the files (`fullRes`, shown in Table 6.11), the number of operations required for a user to correct the obtained beat detections is reduced in more than 11,000 annotations; from 12,912 for the baseline to 1,904 for the finetuned configuration `ft`. Given that the *Candombe* dataset comprises a total of 19,136 beat annotations, this corresponds to curtailing more than half (i.e. 57%) the number of annotations required to cover this dataset fully. Certainly, these counts come at the expense of user-annotations, which amount to a total of 4,757 beat annotations; nonetheless, these outcomes are highly telling. As seen by the Ae column, through finetuning (`ft`), each user-annotation (done to inform the finetuning) equates to a saving of 155.7% fewer correction operations than the baseline.

Table 6.11: Mean of the E-measure and Ae scores and sum of the #det, #ins, #del, #shf and #ops scores across the *Candombe* dataset for the various configurations. (`fullRes`)

Dataset	Model	E-measure	#det	#ins	#del	#shf	#ops	Ae
Candombe	bsl	0.319	6,316	2,901	92	9,919	12,912	–
	ft+da	0.919	16,688	1,885	181	561	2,632	2.439
	ft+da+pt	0.915	16,575	1,997	178	563	2,739	2.420
	ft+da+tg	0.922	16,892	*1,504*	190	739	2,437	2.444
	ft+da+tg+pt	0.923	16,903	*1,504*	188	726	2,421	2.449
	ft	**0.949**	**17,330**	1,565	98	*242*	*1,904*	**2.557**
	pt	0.319	6,319	2,821	100	9,996	12,917	–
	tg	0.369	7,498	1,567	*90*	10,071	11,728	–

Summary of the Results:

The beat tracking results for the *Candombe* dataset demonstrate substantial improvements in accuracy when fine-tuning configurations are compared to the baseline. The simple finetuning configuration (`ft`) outperforms others, improving the F-measure from 0.280 to 0.952, an absolute increase of 0.672, which puts the finetuned performance at roughly 3.4 times the performance of the baseline. This improvement is mirrored across continuity-based metrics, resulting in increases ranging from 60.2 p.p. for AMLt to 74.0 p.p. for CMLc. The annotation-correction workflow analysis indicates a significant reduction in the number of operations required to correct beat estimates. With the optimal configuration (ft), correction operations are reduced from 10,447 to 1,534, an absolute reduction of 8,913 operations, making the fine-tuned beat tracker clearly more efficient in practice.

6.2 An Extremely Challenging Case of Beat Tracking

Having previously dealt with a wide variety of difficult cases for beat-tracking, we choose an exceptionally challenging example in the current experimental scenario. The phenomenon of *polytempo*, which has been introduced in Chapter 2, is virtually absent from mainstream music genres such as pop, folk, classical, and others; as a result, it is not represented in publicly available datasets used to train state-of-the-art beat-tracking models. In addition, its nature poses an (almost) insurmountable hurdle for general-purpose beat-trackers: the existence of concurrent and isochronous pulses in the same piece of music.

Despite being an uncommon compositional technique—as demonstrated by the limitations of current music software tools in handling such complexities [Renney and Gaster, 2019; Hunt, 2020]—this phenomenon is primarily found in Western music, specifically within the so-called *avant-garde*. Charles Ives's 'Symphony no. 4' is considered the earliest formalised and non-trivial work that features polytempo [Galvão, 2014]. Later on, other composers such as Conlon Nancarrow, György Ligeti, and Iannis Xenakis also explored this approach [Taylor, 2003]. Nancarrow, in particular, often composed for *Player Piano*, allowing the realisation of complex rhythmic variations that surpassed the abilities of a human pianist. A peculiar manifestation of *polytempo* is Steve Reich' *phasing*, a compositional technique where two or more identical phrases or motifs are played simultaneously but at slightly different *tempi*, creating a gradual shift in phase between them [Holmes, 2008]. In 1972's *Clapping Music* (for two performers), Reich has the musicians change to the next rotation on the downbeat of a measure, without gradually speeding up. In contrast, in *Piano Phase– for two pianos (or marimbas)*, Reich instructs the second pianist to gradually increase in speed so that the patterns go slowly out of phase.

In *Piano Phase*, Reich brought this compositional technique to live performance (a rendition of the original score is shown in Figure 6.10), with a thorough set of instructions for performance [Christensen, 2004], which we briefly summarise:

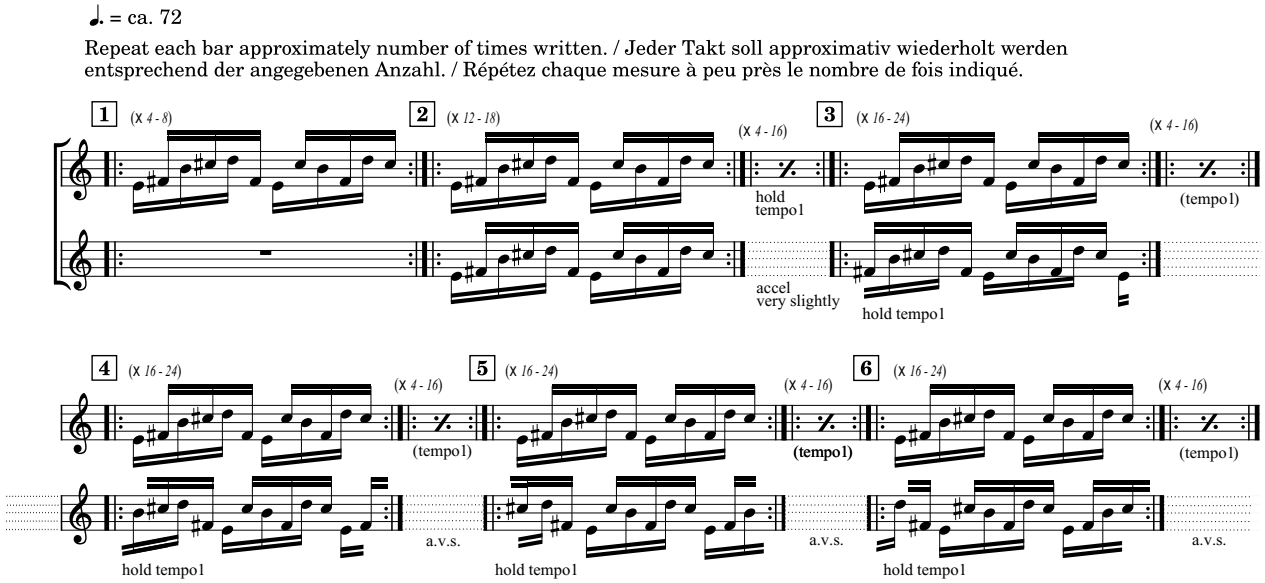


Figure 6.10: Steve Reich *Piano Phase*: reproduction of the original Score.

1. *One performer starts, the other fades in unison (bars* 1–2*), and both continue playing the pattern over and over again*;
2. *The first performer keeps a constant tempo. The other performer gradually increases his tempo, until he is one note ahead of the first performer (bar* 3*)*;
3. *After playing in synchronisation for a while, the second performer again begins increasing his tempo, and the phase shifting process starts again (bars* 3-4*)*;
4. *In the first part of the piece, this procedure is repeated twelve times.*

This composition demands a high level of skill and proficiency from the performers, making it suitable for only a select group of musicians. Similarly, the task of manually annotating beats within this composition presents significant challenges that can only be adequately addressed by individuals with a high level of proficiency in music. For those without such expertise, the task may prove to be excessively demanding. The selection of Steve Reich's *Piano Phase* for analysis presents our approach to beat-tracking with a formidable challenge; to our knowledge, this is the first reported endeavour at beat-tracking a polytemporal composition.

Preparation

The musical composition *Piano Phase* poses challenges to those involved in the process of beat-tracking, i.e. musicians, human annotators, and computational beat-trackers.

Additionally, the lack of *polytempo*–suited tools upstream, particularly those related to MIDI sequencing or musical notation [Hunt, 2020], further complicates the analysis and the reporting process. As a result, it was not feasible to generate a faithful rendition of this piece using traditional MIDI-related tools. To address these limitations, and making use of the procedural nature of Reich's composition, a simplified version of *Piano Phase* was generated using *PureData*[53] (*Pd*) software.

Our *Pd* patch (see Appendix A.4) generated two streams of 12 MIDI notes corresponding to the main motif depicted in Figure 6.10. These streams were played at slightly different *tempi*, with a tempo of approximately[54] 72 bpm for stream A and 73 bpm for stream B. The audio resulting from this process, with a duration of 2 minutes, was obtained by utilizing a MIDI-driven piano synthesizer. Furthermore, we also produced solo renditions of streams A and B for evaluation purposes.

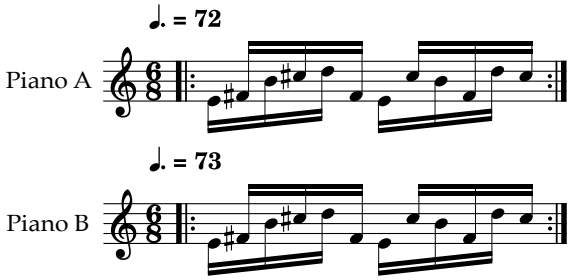


Figure 6.11: Musical score of the simplified version of *Piano Phase*.

At the same time, the ground-truth beat annotation for each of the streams was created assuming a 6/8 time signature, thereby adopting the dotted quarter note (♩.) as the unit of musical time (i.e. the beat) — as inferred, but not explicitly stated in the original score[55].

Methodology

The main experimental goal is to evaluate whether or not our beat-tracker can synchronise with the different tempi present in the music, and to what extent. To this end, we used our internal dataset (described in Table 6.12), whose main files are

[53] `https://puredata.info/`

[54] We followed the composer's instructions in the original score (♩.=72) for stream A and assigned a value of ♩.=73 for stream B. These correspond to the inter-onset-interval of 138.8 ms and 136.9 ms, respectively for patch A and patch B, which we rounded for the nearest integer in Pd `metro` implementation.

[55] In 1972's *Clapping Music for Two Performers* "Directions for Performance", Reich explains the rationale for the lack of metric notation: *"(...) the downbeat always falls on a new beat of the unchanging pattern. No other accents should be made. It is for this reason that a time signature of* 6/4 *or* 12/8 *is not given –to avoid metrical accents."*

pianophaseM_A and pianophaseM_B, corresponding to the mixed audio (M:A+B) and the ground-truth annotations of streams A and B, respectively. Additionally, in order to evaluate any potential disparities in the beat tracking effectiveness between both streams, we also incorporated the individual renditions of stream A and B (pianophaseA_A and pianophaseB_B) in our analysis.

Table 6.12: Composition of the *PianoPhase* dataset.

Filename	Audio	Ground Truth	File Length (mm:ss)
pianophaseM_A	A + B	A	02:00
pianophaseM_B	A + B	B	02:00
pianophaseA_A	A	A	02:00
pianophaseB_B	B	B	02:00

Consistent with the methodology outlined in previous sections, we used a finetuning region that comprised 25% of the total duration of each file in our experiment. We utilised the main set of configurations (as defined in Table 5.2): ft+da, ft+da+pt, ft+da+tg, ft+da+pt+tg, and the baseline configuration, denoted by bsl. In this experiment, we report the full-length results (fullRes), which include the finetuned portion of the input signal.

Results

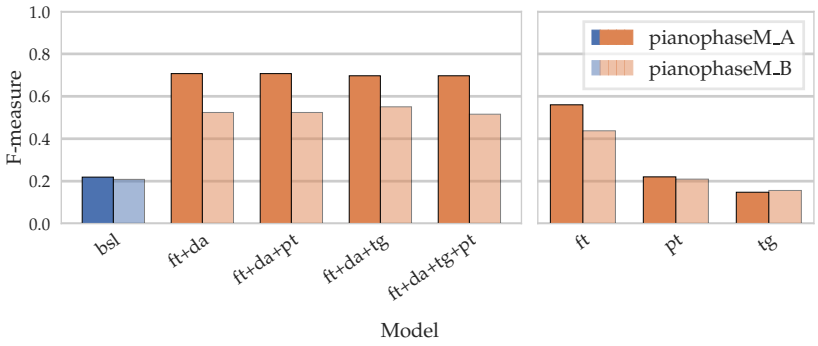


Figure 6.12: Ablation for the *main* set of configurations for the files pianophaseM_A and pianophaseM_B.

Before discussing the results, it is important to note that comparing our approach with any other beat tracker, including the current state of the art, may not be entirely fair. Beat trackers are primarily designed to identify a single tempo, rendering them unsuitable for analysing music with concurrent tempi. Furthermore, DNN-based

approaches have not encountered remotely similar examples in conventional training datasets. Nevertheless, despite the inherent limitations of this comparison, evaluating performance against the baseline remains the most appropriate method for assessing the effectiveness of our approach.

Overall, we observe significant improvements across all finetuning configurations compared to the baseline approach for both streams (`A` and `B`). Figure 6.12 demonstrates a consistent, albeit unexpected, poorer adaptation to stream `B` compared to stream `A`, with a difference of 2 p.p. The available data is insufficient to provide an explanation, and further investigation is warranted.

Examining the configurations, finetuning (`ft`) appears responsible for the majority of the improvement in final performance, as the isolated use of DBN-parameterisations (`pt` and `tg`) does not enhance beat-tracking accuracy when compared to the baseline (`bsl`).

Nevertheless, the best configuration for testing mixed audio with annotations for stream `B` (`pianophaseM_B`) combines finetuning with data augmentation and the tempo guide DBN parameterisation (`ft+da+tg`). However, it should be noted that, given the small-sized nature of this study, we cannot draw any particular significance from minor accuracy differences.

For stream `A`, as depicted in Figure 6.13, the F-measure increases from 0.219 to approximately 0.700 across most configurations. This substantial improvement in performance corresponds to a reduction from 171 correction operations for the baseline to just 47 operations in the best-performing configurations (`ft+da` and `ft+da+pt`). Figure 6.14 presents the results for finetuning to stream `B`, revealing an enhancement in the F-measure from 0.208 to 0.513 in the worst case and 0.551 in the best case (`ft+da+tg`), marking an improvement of more than 3 p.p. in any scenario.

Upon examining the different music segments, we find that for stream `A`, synchronisation (between the algorithm and the correct stream) is lost around two-thirds of the way through bar 5. At this point, the prediction function appears somewhat noisy, with correct beat positions still being recovered intermittently until after the start of bar 6. It completely loses focus until nearly bar 8. Interestingly, the baseline exhibits better performance in this region, between bars 6 and 8. The algorithm then recovers sync with the correct stream pulse until the end of the test file.

Regarding the adaptation to stream `B`, we observe that synchronisation is lost around one-third of the way through bar 4. At this point, the prediction function appears weak, with correct beat positions still being recovered (hence the superior performance of `ft+da+tg`) by the use of `tg` until approximately the start of bar 5.

Subsequently, we see a complete loss of beat tracking capability, even though the beat activation function displays increasingly stronger peaks, albeit with slightly inaccurate positioning.

Considering that we have 12 main patterns (i.e., primary relations between streams A and B, as depicted in the score of Figure 6.10), it would be valuable to repeat this experiment with larger music excerpts that encompass at least one meta-loop (i.e., bars 1–12), as well as exploring different finetuning regions and sizes. This would provide additional insights and a more comprehensive understanding of the specificities of the adaptation to polytempi.

In conclusion, although limited in scope, these preliminary findings indicate that our finetuned approach performs significantly better than the current state of the art in this particularly challenging musical example. This showcases significant potential for atypical musical examples that are not only unrepresented in annotated datasets but also defy mainstream musical standards, pushing the limits of beat-tracking methods.

Summary of the Results:

The beat tracking results for the polytempo case show significant improvements across all configurations compared to the baseline approach for both streams (A and B). The simple finetuning configuration (ft) is primarily responsible for the majority of the improvement in final performance. For stream A, the F-measure increases from 0.219 to approximately 0.700 across most configurations, corresponding to a reduction from 171 correction operations for the baseline to just 47 operations in the best-performing configurations (ft+da and ft+da+pt). For stream B, the F-measure is enhanced from 0.208 to 0.513 in the worst case and 0.551 in the best case (ft+da+tg), marking an improvement of more than 3 p.p. in any scenario. These initial findings, while exploratory in nature, indicate a promising trend: our approach shows substantial improvements over the current state-of-the-art methods when faced with this especially challenging musical example.

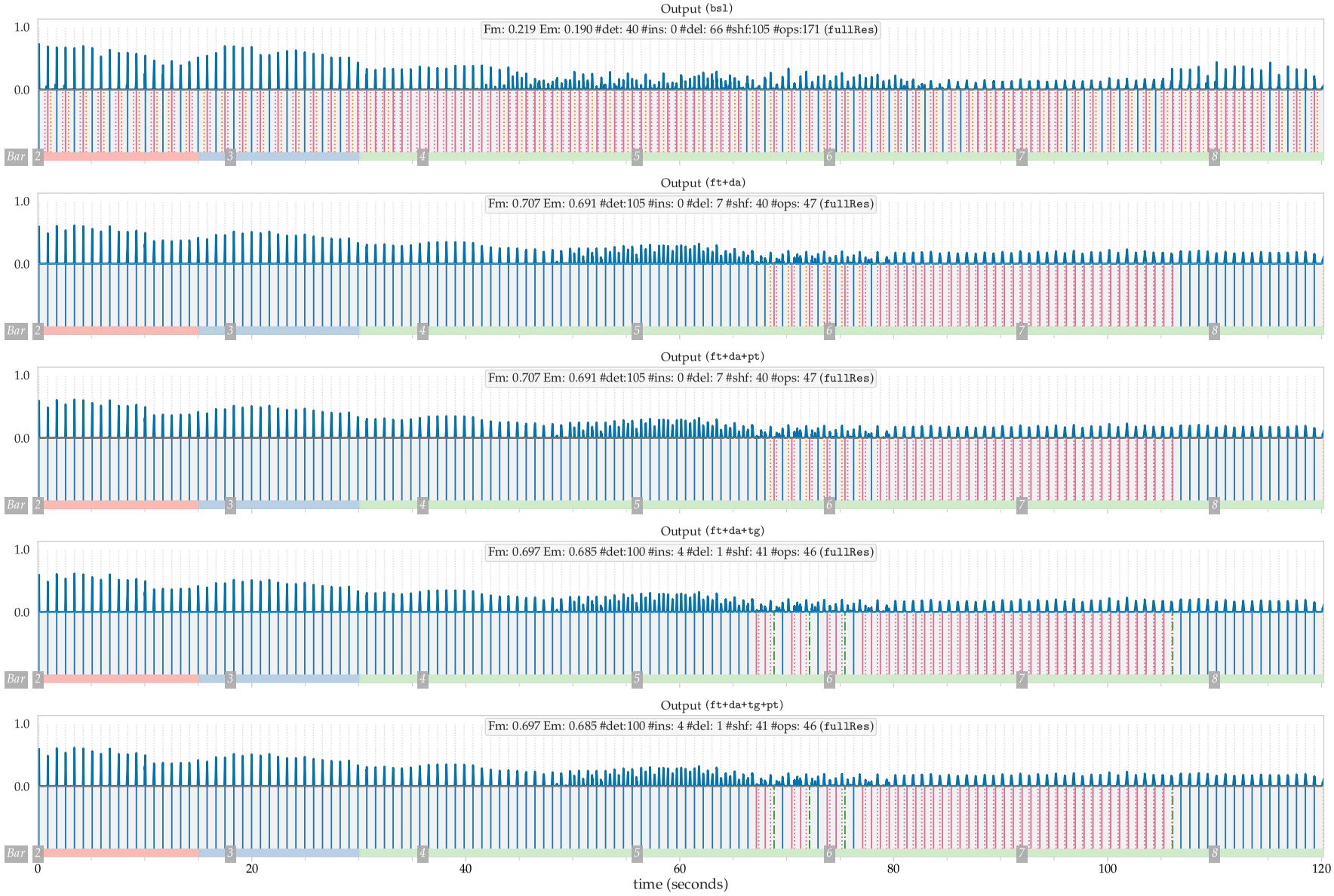


Figure 6.13: Detailed analysis for `pianophaseM_A` (mixed audio and annotations for stream `A`).

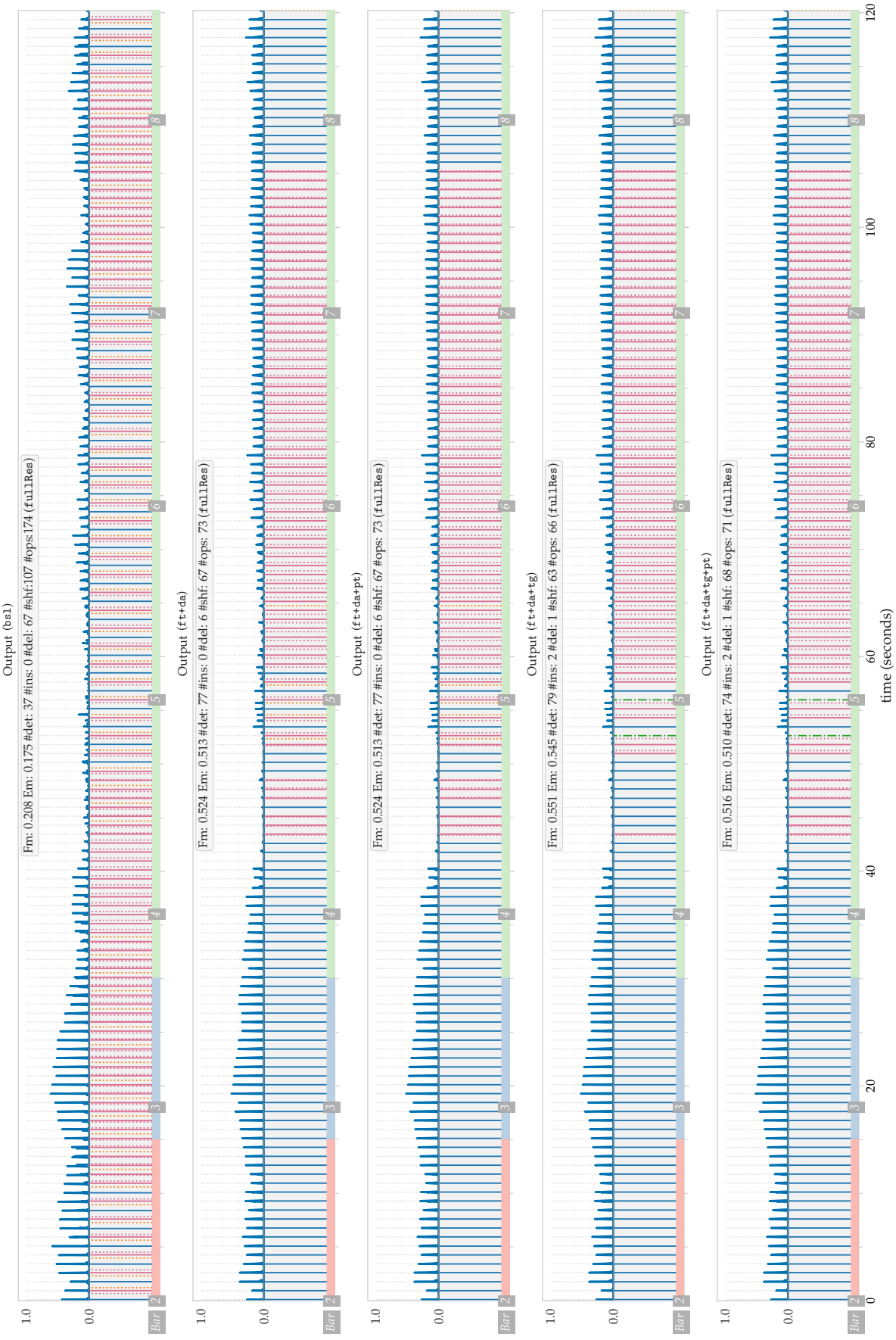


Figure 6.14: Detailed analysis for `pianophaseM_B` (mixed audio and annotations for stream B).

6.3 Summary

In this chapter, we have expanded the application of our approach to computational rhythm analysis across a variety of diverse and challenging contexts. Starting with the Brazilian *Maracatu* rhythmic ensemble, we used transfer learning to enhance musical onset detection performance. By systematically examining finetuning strategies within an inductive setting, we evaluated the effects of retraining different layer sets. This exploration resulted in improved performance and shed light on the impact of various retraining regimes, corroborating previous studies. Furthermore, we included the transductive setting in our analysis, which as far as we are aware, is the first time it has been explored in the context of beat tracking and onset detection.

Progressing into beat tracking within the Colombian *Bambuco* genre, known for its challenging bi-metric nature, we found that all finetuning configurations outperformed the baseline. The most effective configuration made significant strides across multiple evaluation metrics. This suggests the aptitude of our approach in handling intricate rhythmic structures like those of *Bambuco*, and also highlights its potential for the task of metre detection. We further applied our methodology to the Uruguayan *Candombe*. Beat tracking results pointed to significant accuracy enhancements, reinforcing the potential of our approach in navigating complex rhythmic scenarios.

In the second section, we embraced a typical creative-MIR use-case, creating a rendition of a contemporary minimalist piece that explores the limits of computational beat tracking by presenting a highly challenging piece for analysis. The selection of Steve Reich's *Piano Phase* for analysis posed a formidable challenge to our approach, introducing what we believe to be one of the first attempts at beat tracking a *polytempo* composition. Results demonstrated significant improvements across all user-centric setups relative to the state of the art. Whilst our findings are exploratory, they underline the versatility of our method in tackling unconventional musical examples that defy mainstream standards, thereby pushing the boundaries of beat tracking.

7

Conclusion

Throughout this thesis, we have been investigating the incorporation of user input and preferences to enhance beat tracking in complex musical environments. Our focus has been directed towards human-in-the-loop strategies, aiming to improve the adaptability of existing Music Information Retrieval (MIR) techniques. The ultimate objective is to establish a context and content-sensitive solution that effectively tackles the challenging rhythmic cases frequently seen in research areas such as creative-MIR and ethnomusicology.

This dissertation started with a comprehensive background review and domain characterisation, outlined in Chapter 2. This chapter introduces crucial music concepts, presents the task of beat tracking, and discusses its associated challenges. We reviewed key classical and contemporary advancements in beat tracking, with an emphasis on the specific audio characteristics or contexts that currently limit the accuracy of the analysis. Recognising the evolving role of the user in Music Information Retrieval (MIR) and Machine Learning (ML), we situated our work within this broader landscape.

Shifting to empirical investigation, we examined the relationship between a user's perception of beat tapping difficulty and the assessment of expressive timing at the signal level, a long-standing difficulty in beat tracking. This led to the development of a

user-driven reparameterisation method for a leading-edge, data-driven beat tracker. As detailed in Chapter 3, this technique effectively customised the beat-tracking algorithm, thereby enhancing accuracy across a variety of musical contexts. Importantly, this was achieved without the need for exhaustive training of the underlying model.

In Chapter 4, we proposed a novel strategy for harnessing user annotations of a brief music segment through a transfer learning approach. Specifically, we focused on finetuning the state-of-the-art model, thereby adapting it *in-situ*. This strategy led to significant improvements in performance across various reference datasets, exhibiting robust adaptability to a variety of musical timbres and expressions.

To effectively gauge the impact of our proposed strategies, we introduced two novel user-centric evaluation metrics in Chapter 4. The *E-Measure* and *Annotation Efficiency* metrics offer a new perspective on beat tracking model evaluation, focusing on the user's annotation workflow and the effort required to achieve a ground-truth annotation. These metrics facilitated a more accurate and practical assessment of the techniques developed.

In Chapter 5, we consolidated the user-driven reparameterisation and *in-situ* fine-tuning into a holistic approach. This combined strategy utilises both user annotations and preferences to dynamically customise the underlying beat-tracking algorithm, offering improved robustness to various musical contexts.

We have illustrated the benefits of harnessing user knowledge for facilitating *in-situ* adaptation of leading-edge MIR methods. Our work addresses complex musical contexts without necessitating extensive model retraining, establishing a user-centric approach that bolsters the precision of MIR techniques. By expanding the adaptability to a diverse range of musical signals, our approach contributes to the advancement of computational rhythm analysis.

Through these contributions, we have demonstrated the potential of utilising user knowledge to enable *in-situ* adaptation of top-tier MIR methods. Our approach addresses challenging musical scenarios without the need for extensive model training. This user-centric strategy enhances the accuracy of MIR methods, broadening the range of musical signals they can adapt to, thereby pushing the boundaries of computational rhythm analysis. This strategy provides a pragmatic answer to the question: *How can we adapt when even state-of-the-art techniques fall short?* The outcome of our research unveils the potential for the development of learning techniques capable of swiftly adjusting to new content with minimal user-input. In conclusion, these insights reinforce our original thesis: user knowledge can indeed be harnessed for *in-situ* adaptability of leading MIR methods, bolstering their resilience to challenging musical scenarios.

The following sections of this chapter outline our main contributions to computational rhythm analysis in challenging musical conditions. We conclude by discussing some promising directions for future work which bring to light implications of our work in a broader setting.

7.1 Summary of Contributions

We summarise our key contributions and principal outcomes as follows:

- We introduced a user-driven reparameterisation approach for an advanced data-driven beat tracker, improving accuracy without necessitating extensive model training (Chapter 3). This methodology harnessed user insight about musical expressive timing to customise the algorithm's post-processing Dynamic Bayesian Network, enhancing the analysis of highly-expressive music.

- We proposed a finetuning method that adapted the state-of-the-art beat-tracker using a short beat-annotated region (Chapter 4). This method resulted in enhanced performance in complex musical settings by exposing the model to a limited set of user-annotated data. It significantly improved performance across reference datasets, demonstrating adaptability to diverse musical timbres and expressions. Remarkably, we achieved a mean F-measure improvement ranging from 7% to over 16% for the challenging SMC dataset, and saw a 2/3 reduction in the total number of correction operations for the best configurations compared to the state-of-the-art performance.

- Combining user-driven reparameterisation and *in-situ* finetuning, our integrated approach to user-centric adaptation expanded the adaptability of the state-of-the-art beat tracker to various musical contexts (Chapter 5). This strategy considerably outperformed the baseline in all datasets, with particularly sharp gains in the analysis of non-Western music datasets. Our top-performing configuration showed an average F-measure increase of approximately 25% for the *Bambuco* dataset, while continuity-based metrics (CMLc and CMLt) demonstrated gains of 38 to 45%, respectively.

- We developed two novel user-centric metrics for beat-tracking evaluation, each focusing on the user's perspective within an annotation workflow (Chapter 4):

- The *E-Measure* is our adaptation of the well-established F-measure to more accurately capture the edit operations common in beat annotation workflows. This includes correct detections, deletions, insertions, and uniquely, the *shift* operation — our explicit addition that improves the alignment of algorithm's performance evaluation to the user annotation workflow.
- The *Annotation Efficiency* (Ae) metric assesses the improved performance in relation to user effort compared to the baseline approach. It quantifies the reduction in correction operations achieved through the finetuning process, normalised by the number of user annotations. This metric offers a quantitative insight into the practical benefits of our method.

In addition to these contributions, we developed the `beatflow` library, an open-source tool that models the beat-tracking annotation workflow, providing complementary user-centric metrics and a visualisation module to graphically depict evaluation results in terms of correction edit operations (Chapter 4).

In conclusion, our work has demonstrated the potential of user knowledge in enhancing the *in-situ* adaptability of top-tier MIR methods, improving their robustness against challenging musical conditions. It has also opened a pathway towards more accurate analysis of complex musical pieces, thereby extending the possibilities of computational rhythm analysis.

Finally, our research resulted in the development and open sourcing of the `beatflow` library, which models the beat-tracking annotation workflow. It provides complementary user-centric metrics and a visualisation module that graphically depicts evaluation results in terms of correction edit operations, thereby enhancing the practical efficiency and applicability of our approach.

In conclusion, our user-centric approach demonstrated the potential of leveraging user knowledge for *in-situ* adaptability of top-tier MIR methods, thereby improving robustness to challenging musical conditions. Our contributions have not only enhanced the accuracy and adaptability of MIR methods but also presented a path towards highly accurate analysis of complex musical pieces, thus pushing the boundaries of computational rhythm analysis.

7.2 Future Work

In this thesis, we have demonstrated the potential of human-in-the-loop computational rhythm analysis for challenging music signals. However, we believe that there are many

more possibilities to explore in this field, and several directions for future research can build upon the work presented here. In this section, we briefly discuss some of these potential future developments.

We will first focus on immediate extensions and improvements that stem directly from our ongoing work.

Advancing User-Centric Metrics: As discussed in Section 4.3, our study has advanced user-centric metrics to better model user effort in the context of beat annotation. However, there remain several key areas for future work, from short-term to long-term paths for research:

- Developing an expanded set of metrics to encompass the diverse aspects of the process under investigation. Depending on the scenario, the primary focus might be on the absolute number of corrections avoided through finetuning, the relative improvement in correction operations, among other aspects, according to the specific context.
- Refining the existing method for matching estimated events or beats with potential operations, which currently uses a greedy strategy. Our aim is to explore global optimisation techniques, drawing from established graph-based strategies such as those used in `mir_eval` [Raffel et al., 2014].
- Modelling the personalised nature of annotation workflows. Within this long-term perspective, we plan to represent edit operations through user-based cost profiles. This will allow us to account for the varying effort required by different users to perform the same correction operation, thus enabling the derivation of annotation metrics using specific user templates.

By probing these additional aspects and formulating an extended set of metrics, we expect to strengthen our capacity to model the full annotation workflow in an adaptive setting. Improving both aspects will entail considerable updates at the code level. To reflect our commitment to open research and easier tool adaptation, we plan to update our shared base code and transition towards a class-based design. This transition will incidentally simplify the use and adoption of our evaluation metrics and visualisation software, thus fostering wider research application.

Optimizing Finetuning Region Selection: The exploratory work in Section B.1 suggests that alternative strategies for selecting the finetuning region could lead to better results. For instance, an initial beat tracking analysis and visualisation could help users identify regions with significant deviations from the ground truth data. Additionally, novel approaches for network adaptation that observe the entire piece,

such as semi-supervised learning, could help overcome limitations associated with finetuning based only on partial input observations. This research direction finds support in HITL [Dudley and Kristensson, 2018] and brings enhanced user guidance in selecting the optimal finetuning region.

Exploring Advanced Retraining Strategies: Our results, as shown in B.2, indicate the potential for performance improvement through the exploration of advanced optimisation techniques for retraining the model. An investigation into discriminative finetuning, gradual unfreezing [Howard and Ruder, 2018], and alternative retraining optimisation techniques could yield promising advancements. A more systematic evaluation of layers to be frozen and input-dependent finetuning, which automatically determines layers to finetune per target instance [Guo et al., 2018], is also worth considering.

Adapting the Architecture: In our study, detailed in section 4.5.2 and elaborated throughout this thesis, we have shown that model finetuning helps adapt to the unique attributes of individual musical signals. This results in a beat activation function with more pronounced peaks. With such clear peaks in beat likelihood, the need for the Dynamic Bayesian Network (DBN) may be obviated in certain scenarios. Future research could explore the feasibility of bypassing the DBN when clear beat activation peaks are present. This modification could augment control, streamline operations and potentially boost overall performance. It also opens up the possibility of selecting disjoint finetuning regions, which is not possible with the current DBN post-processing. Additionally, this approach could spur alternative strategies for extracting beat positions from the TCN output. For instance, smoothing the BAT over a broader interval around each beat could potentially handle distinct signals such as the polytempo analysed in Section 6.2.

Expanding Rhythm Analysis Tasks: Our work presented in 6 has demonstrated the applicability of our approach to additional MIR tasks, such as onset detection and, indirectly, metre determination. A logical and straightforward extension would involve exploring tasks closely associated with beat tracking, like downbeat detection and tempo tracking, capitalising on the multi-task capabilities of the underlying state of the art. Moreover, broader scope tasks, where rhythm analysis plays a pivotal role, such as structural segmentation, could benefit from our adaptive solution as a means to address the inherent subjectivity and ambiguity.

After discussing extensions directly related to our current research, we now consider wider future work that branches out more broadly from our core work. This broader view resonates with the principles of *Open Research*. In embracing these principles, we

highlight their importance and align our efforts with the collective goals of enhancing collaboration, enriching the research community, and fostering developments that will benefit both researchers and practitioners in a cooperative ecosystem.

Promote Creation/Improvement of Challenging Open Datasets: The MIR community currently faces significant challenges related to the scarcity of open datasets [Bittner et al., 2019]. This issue is particularly pertinent in the context of beat tracking, as the *SMC* dataset – the only resource offering specifically challenging examples – provides short-duration snippets. To foster advancements in a field increasingly reliant on extensive data, addressing this scarcity is essential. Our approach holds potential for semi-automatic annotation, that we can further explore. For instance, adopting a single dataset-wide finetuning step, capitalising on the expected music similarity within a dataset, as has been done in previous research [Fiocchi et al., 2018; Maia et al., 2022]. Thus, our approach could serve as a tool for creating new open datasets featuring full-length demanding musical examples, minimising user effort. This would allow us to contribute directly and indirectly to the goal of enriching the pool of MIR annotated challenging data.

Streamlining Annotation and Beat Tracking Workflows: In the progression of our work, one worthwhile consideration is the integration of our approach with established MIR libraries and tools, optimising beat tracking workflows, particularly for complex musical signals. Past research on tools for onset detection [Valero-Mas and Iñesta, 2017] and beat tracking [Driedger et al., 2019] indicates that semi- or fully-automatic strategies can reduce correction workloads. Although our approach primarily concentrates on model adaptation rather than specific tool development, the integration potential remains encouraging. However, we realise this integration calls for collaborative efforts with developers of current MIR libraries and tools, a goal to which we, as researcher/engineers, are prepared to undertake.

Additionally, exploring these broader research directions promises to enrich our work with User Experience (UX) research methodologies. A semi-automatic annotation campaign would offer opportunities for field studies, providing insights into user strategies and refining our user-centric metrics. The integration of our contributions with widely used MIR tools, on the other hand, allows for UX studies focused on user engagement and the impact of our solutions on beat tracking workflows. By pursuing these paths, we can improve our methodologies using findings from UX research, thus enhancing the applicability and relevance of our work.

Appendices

A

Appendix A – Complementary Results

A.1 Additional Results for Chapter 4

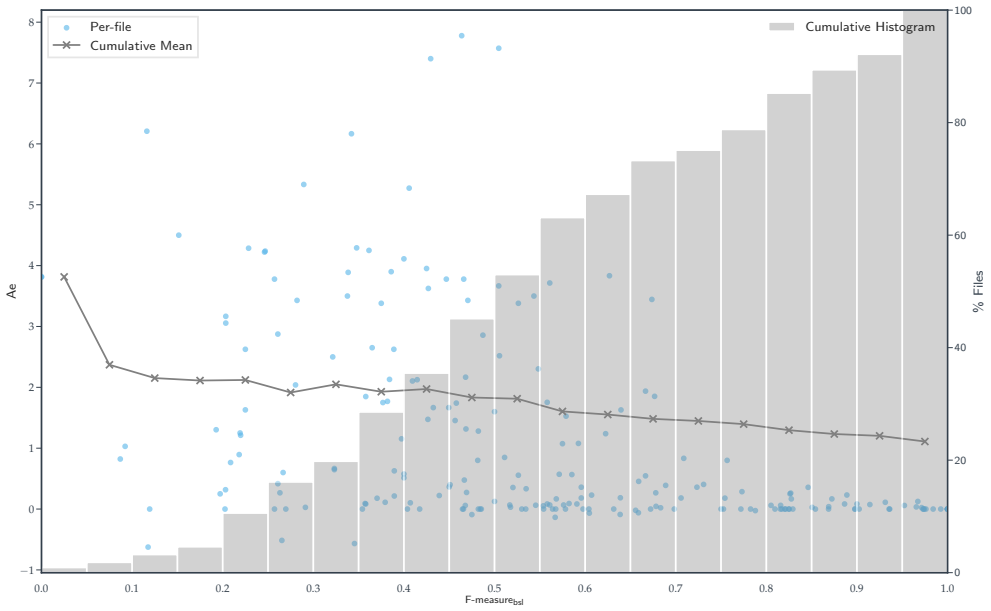


Figure A.1: Annotation efficiency (Ae) *vs.* baseline F-measure for the *SMC* dataset. The scatter plot (in blue) illustrates individual file Ae values. The line plot indicates the cumulative mean Ae computed in 20 bins across the F-measure range. As the F-measure increases, indicating better baseline performance, the Ae value tends to decrease, highlighting the diminishing returns of finetuning for files with already satisfactory baseline performance. This trend underscores the bias in cross-dataset evaluations using a simulated environment, potentially underestimating finetuning's true performance.

A.2 Additional Results for Chapter 5

Table A.1: Mean of the F-measure score across all files of the *SMC* dataset for the various configurations. The best results for each file are highlighted in *grey*. (`testRes`).

File	Model							
	bsl	**ft+da**	**ft+da+pt**	**ft+da+tg**	**ft+da+tg+pt**	**ft**	**pt**	**tg**
SMC_001	0.493	0.512	0.522	0.739	0.739	0.502	0.493	**0.783**
SMC_002	0.359	0.359	0.438	**0.441**	**0.441**	0.359	0.359	0.358
SMC_003	0.369	0.760	0.746	**0.800**	**0.800**	0.474	0.369	0.176
SMC_004	0.438	0.515	0.556	0.561	**0.659**	0.516	0.438	0.464
SMC_005	0.276	0.276	0.276	**0.410**	**0.410**	0.310	0.276	**0.410**
SMC_006	0.472	0.545	0.567	0.750	**0.798**	0.524	0.500	0.704
SMC_007	0.411	0.411	0.411	0.528	0.528	0.411	0.411	**0.542**
SMC_008	0.393	0.393	0.393	**0.846**	0.771	0.532	0.393	0.761
SMC_009	0.422	0.580	0.444	**0.852**	**0.852**	0.444	0.422	**0.852**
SMC_010	**0.895**	**0.895**	**0.895**	**0.895**	**0.895**	**0.895**	**0.895**	**0.895**
SMC_011	0.818	0.818	**0.897**	0.818	**0.897**	0.818	**0.897**	0.818
SMC_012	**0.410**	0.386	0.376	0.386	0.376	**0.410**	**0.410**	**0.410**
SMC_013	**0.863**	0.807	0.807	0.807	0.807	**0.863**	**0.863**	**0.863**
SMC_014	0.763	0.737	**0.772**	0.737	**0.772**	0.737	0.737	0.763
SMC_015	0.357	0.480	0.536	0.882	**0.971**	0.360	0.351	0.278
SMC_016	0.696	**0.725**	**0.725**	**0.725**	**0.725**	**0.725**	0.667	0.696
SMC_017	**0.833**	**0.833**	**0.833**	**0.833**	**0.833**	**0.833**	**0.833**	**0.833**
SMC_018	0.442	**0.512**	0.496	**0.512**	0.496	0.465	0.447	0.442
SMC_019	0.441	0.459	0.459	**0.700**	**0.700**	0.441	0.441	**0.700**
SMC_021	0.615	0.561	0.547	0.561	0.547	**0.622**	0.615	0.615
SMC_022	0.448	0.615	0.667	**0.730**	**0.730**	0.636	0.576	0.523
SMC_023	0.627	0.640	0.640	**1.000**	**1.000**	0.627	0.627	0.588
SMC_024	**0.473**	**0.473**	**0.473**	**0.473**	**0.473**	**0.473**	**0.473**	**0.473**
SMC_026	0.447	0.439	0.414	0.627	0.627	0.387	0.447	**0.706**
SMC_027	0.839	**0.849**	0.839	**0.849**	0.839	0.839	0.839	0.839
SMC_028	0.494	0.549	0.549	**0.822**	**0.822**	0.549	0.494	0.800
SMC_030	0.500	0.778	0.778	**0.995**	**0.995**	0.945	0.500	0.949
SMC_032	0.493	0.493	0.489	**0.495**	**0.495**	0.481	0.493	0.493
SMC_033	0.447	0.511	0.519	0.511	0.519	**0.587**	0.561	0.562
SMC_034	0.867	0.867	0.867	0.867	0.867	**0.900**	0.867	0.867
SMC_035	0.741	**0.778**	**0.778**	**0.778**	**0.778**	**0.778**	0.741	0.741
SMC_036	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_037	0.581	0.588	0.588	0.588	0.588	0.557	0.590	**0.621**

Continued on next page

Table A.1: Mean of the F-measure score across all files of the *SMC* dataset for the various configurations. The best results for each file are highlighted in *grey*. (`testRes`).

File	Model							
	bsl	**ft+da**	**ft+da+pt**	**ft+da+tg**	**ft+da+tg+pt**	**ft**	**pt**	**tg**
SMC_038	**0.793**	**0.793**	**0.793**	**0.793**	**0.793**	**0.793**	0.759	**0.793**
SMC_041	0.353	0.340	0.312	0.285	0.315	**0.548**	0.368	0.300
SMC_042	0.559	0.539	0.539	0.783	0.783	0.559	0.559	**0.826**
SMC_043	0.636	0.615	**0.644**	0.607	0.615	0.592	0.636	0.636
SMC_044	0.740	0.720	**0.767**	0.720	0.711	0.767	0.730	0.579
SMC_046	**0.625**	0.604	0.573	0.604	0.573	**0.625**	**0.625**	**0.625**
SMC_047	0.617	0.658	0.634	**0.708**	**0.708**	0.634	0.617	0.691
SMC_048	0.391	0.565	0.522	**0.688**	0.645	0.464	0.391	0.533
SMC_051	**0.742**	0.689	0.689	0.689	0.689	0.689	0.493	**0.742**
SMC_052	0.667	0.622	0.613	0.631	0.613	**0.689**	0.598	0.667
SMC_054	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_055	0.313	0.657	0.657	0.676	**0.687**	0.657	0.657	0.676
SMC_056	0.205	0.205	0.205	**0.404**	0.336	0.205	0.182	0.184
SMC_057	0.456	0.456	**0.483**	0.462	0.456	0.456	**0.483**	0.464
SMC_058	0.441	0.461	0.472	0.461	0.472	**0.492**	0.441	0.441
SMC_059	0.914	0.914	0.914	0.914	0.914	**0.924**	0.914	0.914
SMC_060	0.531	**0.562**	**0.562**	**0.562**	**0.562**	**0.562**	0.531	0.531
SMC_061	0.500	0.506	0.506	**0.739**	**0.739**	0.473	0.500	0.737
SMC_063	0.707	0.691	0.707	0.691	0.707	**0.715**	0.707	0.707
SMC_064	0.066	0.525	0.536	**0.850**	0.840	0.459	0.098	0.000
SMC_065	**0.861**	**0.861**	**0.861**	**0.861**	**0.861**	**0.861**	**0.861**	**0.861**
SMC_066	0.265	0.394	0.358	**0.622**	**0.622**	0.343	0.299	0.444
SMC_067	**0.776**	0.737	0.737	0.737	0.737	**0.776**	**0.776**	**0.776**
SMC_068	**0.886**	**0.886**	**0.886**	**0.886**	**0.886**	**0.886**	**0.886**	**0.886**
SMC_069	0.762	0.738	0.738	0.738	0.738	0.762	**0.819**	0.762
SMC_071	0.763	0.789	**0.886**	0.789	**0.886**	0.781	0.868	0.763
SMC_072	0.889	0.889	0.871	**0.903**	0.871	0.889	0.871	**0.903**
SMC_073	0.857	0.899	**0.928**	0.899	**0.928**	0.886	0.870	0.857
SMC_074	0.652	0.772	0.787	0.777	**0.826**	0.719	0.697	0.652
SMC_075	**0.560**	0.304	0.207	0.304	0.207	0.507	**0.560**	**0.560**
SMC_076	0.366	0.346	0.328	0.528	0.537	0.579	0.366	**0.718**
SMC_079	0.569	0.585	0.585	0.754	**0.762**	0.555	0.588	0.759
SMC_080	0.250	0.360	0.360	**0.485**	**0.485**	0.317	0.286	0.258
SMC_082	0.640	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	0.640	**0.880**
SMC_084	0.082	0.063	0.063	0.000	0.000	**0.122**	0.082	0.047
SMC_085	0.269	0.256	0.244	**0.295**	**0.295**	0.269	0.269	0.269

Continued on next page

Table A.1: Mean of the F-measure score across all files of the *SMC* dataset for the various configurations. The best results for each file are highlighted in *grey*. (`testRes`).

File	Model							
	bsl	**ft+da**	**ft+da+pt**	**ft+da+tg**	**ft+da+tg+pt**	**ft**	**pt**	**tg**
SMC_086	0.656	**0.719**	**0.719**	**0.719**	**0.719**	**0.719**	**0.719**	0.656
SMC_087	0.945	**0.989**	**0.989**	**0.989**	**0.989**	**0.989**	0.945	0.945
SMC_088	**0.561**	0.526	0.526	0.526	0.526	0.526	**0.561**	**0.561**
SMC_089	0.115	0.116	0.115	0.114	0.114	0.103	**0.116**	0.115
SMC_092	**0.976**	0.957	0.957	0.957	0.957	**0.976**	**0.976**	**0.976**
SMC_093	0.526	0.610	0.533	0.810	**0.845**	0.547	0.568	0.688
SMC_095	0.974	**0.982**	**0.982**	**0.982**	**0.982**	0.947	0.947	0.974
SMC_096	**0.923**	0.917	0.917	0.917	0.917	**0.923**	**0.923**	**0.923**
SMC_098	**0.681**	**0.681**	**0.681**	**0.681**	**0.681**	**0.681**	**0.681**	**0.681**
SMC_099	0.286	0.386	0.271	**0.464**	0.310	0.280	0.286	**0.464**
SMC_100	0.448	0.483	**0.517**	0.357	0.381	0.460	0.414	0.448
SMC_101	**0.673**	**0.673**	**0.673**	0.614	0.640	**0.673**	**0.673**	0.614
SMC_103	0.333	0.453	0.480	0.453	0.480	0.480	0.361	**0.545**
SMC_104	**0.531**	0.434	0.487	0.444	0.476	**0.531**	**0.531**	**0.531**
SMC_105	0.170	0.292	0.251	0.299	**0.314**	0.234	0.170	0.250
SMC_106	**0.613**	0.570	0.559	0.570	0.559	0.525	**0.613**	**0.613**
SMC_109	**0.613**	0.602	0.575	0.602	**0.613**	0.581	0.562	**0.613**
SMC_111	0.453	0.447	0.502	0.621	**0.633**	0.356	0.480	0.414
SMC_113	0.357	0.552	0.529	**0.857**	**0.857**	0.532	0.393	**0.857**
SMC_114	**0.710**	**0.710**	**0.710**	**0.710**	**0.710**	**0.710**	**0.710**	**0.710**
SMC_116	0.197	**0.236**	0.235	0.181	0.172	0.116	0.200	0.116
SMC_117	0.242	0.242	0.242	0.208	0.208	**0.268**	0.242	0.204
SMC_118	0.290	0.209	0.221	0.222	0.222	0.292	0.290	**0.333**
SMC_119	0.207	0.316	0.316	**0.436**	**0.436**	0.207	0.207	0.218
SMC_120	**0.965**	**0.965**	**0.965**	**0.965**	**0.965**	**0.965**	**0.965**	**0.965**
SMC_121	0.192	0.154	0.154	**0.217**	0.136	0.128	0.192	0.061
SMC_124	**0.485**	0.450	0.470	0.450	0.470	0.303	0.424	**0.485**
SMC_126	0.875	**0.906**	**0.906**	**0.906**	**0.906**	0.875	0.875	0.875
SMC_127	0.435	0.462	0.530	**0.543**	**0.543**	0.455	0.471	0.292
SMC_130	0.392	0.481	0.481	0.450	**0.495**	0.392	0.392	0.444
SMC_133	0.697	**0.738**	0.728	**0.738**	0.728	0.727	0.697	0.697
SMC_135	0.704	**0.761**	**0.761**	**0.761**	**0.761**	0.704	**0.761**	0.704
SMC_137	0.423	0.510	0.531	**0.696**	**0.696**	0.624	0.423	0.536
SMC_139	0.421	0.456	0.456	**0.684**	**0.684**	0.456	0.421	0.632
SMC_140	0.753	0.666	0.666	0.304	0.304	0.753	**0.762**	0.312
SMC_142	0.587	**0.596**	0.587	**0.596**	0.587	0.587	0.560	0.587

Continued on next page

Table A.1: Mean of the F-measure score across all files of the *SMC* dataset for the various configurations. The best results for each file are highlighted in *grey*. (`testRes`).

File	Model							
	bsl	**ft+da**	**ft+da+pt**	**ft+da+tg**	**ft+da+tg+pt**	**ft**	**pt**	**tg**
SMC_143	0.500	**0.523**	**0.523**	**0.523**	**0.523**	**0.523**	0.500	0.500
SMC_146	0.519	0.519	0.519	0.519	0.491	0.519	0.519	**0.556**
SMC_147	0.263	0.263	0.263	**0.754**	**0.754**	0.263	0.263	0.533
SMC_148	0.296	0.239	**0.298**	0.254	0.254	0.272	0.296	0.222
SMC_149	0.583	0.583	0.587	0.388	0.388	0.583	**0.595**	0.388
SMC_150	0.953	**0.975**	**0.975**	**0.975**	**0.975**	0.963	0.953	0.953
SMC_151	0.711	**0.762**	0.711	0.723	0.735	0.736	0.711	0.708
SMC_152	0.595	0.493	0.595	0.625	**0.675**	0.600	0.571	0.595
SMC_153	0.547	**0.917**	**0.917**	**0.917**	**0.917**	0.859	0.547	0.547
SMC_154	0.538	0.538	**0.603**	0.549	0.549	0.577	0.538	0.588
SMC_157	**0.263**	**0.263**	**0.263**	**0.263**	**0.263**	**0.263**	**0.263**	**0.263**
SMC_158	0.200	0.200	0.200	**0.235**	**0.235**	0.200	0.200	**0.235**
SMC_159	0.484	0.608	0.608	0.608	0.608	0.484	**0.622**	0.484
SMC_161	**0.610**	**0.610**	**0.610**	**0.610**	**0.610**	**0.610**	**0.610**	**0.610**
SMC_166	0.789	0.895	**0.921**	0.895	**0.921**	0.868	0.763	0.789
SMC_167	0.255	0.405	0.405	**0.667**	0.549	0.431	0.213	0.529
SMC_168	0.298	0.397	0.397	0.516	**0.538**	0.298	0.298	0.387
SMC_169	0.452	0.429	0.452	0.625	0.620	0.435	0.452	**0.641**
SMC_170	**0.882**	**0.882**	**0.882**	**0.882**	**0.882**	**0.882**	**0.882**	**0.882**
SMC_171	0.500	0.526	0.526	0.745	0.745	0.500	0.500	**0.784**
SMC_172	**0.274**	0.194	0.214	0.194	0.194	0.194	0.205	**0.274**
SMC_173	0.533	0.567	0.533	0.700	0.700	0.533	0.567	**0.750**
SMC_174	0.310	0.302	**0.333**	0.302	**0.333**	0.310	0.310	0.310
SMC_175	0.222	0.252	0.252	**0.298**	**0.298**	0.229	0.222	0.200
SMC_176	0.250	0.250	0.271	0.312	0.286	0.260	**0.317**	0.281
SMC_178	0.478	0.542	0.500	**0.750**	**0.750**	0.514	0.522	**0.750**
SMC_179	0.453	0.453	0.446	0.500	**0.571**	0.453	0.415	**0.571**
SMC_181	0.656	0.418	0.436	0.430	0.437	0.634	0.623	**0.746**
SMC_182	0.771	0.771	**0.841**	0.771	**0.841**	0.771	**0.841**	0.771
SMC_184	0.727	**0.742**	**0.742**	**0.742**	**0.742**	0.727	0.727	0.727
SMC_187	0.167	0.261	0.261	**0.387**	0.366	0.261	0.167	0.323
SMC_188	0.657	0.657	0.657	**0.979**	**0.979**	0.657	0.657	**0.979**
SMC_190	0.678	0.764	0.764	**0.800**	**0.800**	0.678	0.678	0.760
SMC_192	0.238	**0.323**	**0.323**	0.261	0.261	0.312	0.238	0.138
SMC_193	**0.812**	**0.812**	**0.812**	**0.812**	**0.812**	**0.812**	**0.812**	**0.812**
SMC_194	0.118	0.197	**0.208**	0.197	**0.208**	0.118	0.118	0.118

Continued on next page

Table A.1: Mean of the F-measure score across all files of the *SMC* dataset for the various configurations. The best results for each file are highlighted in *grey*. (`testRes`).

File	Model							
	bsl	**ft+da**	**ft+da+pt**	**ft+da+tg**	**ft+da+tg+pt**	**ft**	**pt**	**tg**
SMC_195	0.185	0.185	0.185	0.252	0.252	0.185	0.197	**0.255**
SMC_197	0.240	0.392	0.392	**0.647**	0.627	0.392	0.240	0.353
SMC_198	0.804	0.759	0.759	0.590	0.590	**0.823**	0.804	0.571
SMC_199	**0.324**	**0.324**	**0.324**	**0.324**	**0.324**	**0.324**	**0.324**	**0.324**
SMC_202	**0.211**	**0.211**	**0.211**	**0.211**	**0.211**	**0.211**	**0.211**	**0.211**
SMC_203	**0.242**	**0.242**	**0.242**	0.222	0.197	**0.242**	**0.242**	0.222
SMC_204	0.273	**0.386**	0.351	0.309	0.312	0.273	0.273	0.247
SMC_205	**0.861**	**0.861**	**0.861**	**0.861**	**0.861**	**0.861**	**0.861**	**0.861**
SMC_206	0.375	0.510	**0.526**	0.449	0.493	0.431	0.417	0.375
SMC_207	0.541	0.775	0.775	0.775	0.775	0.619	0.541	**0.829**
SMC_208	0.519	0.654	0.654	0.706	**0.719**	0.519	0.519	0.667
SMC_209	**0.648**	0.562	0.571	0.578	0.590	0.620	0.629	**0.648**
SMC_211	**0.625**	0.565	0.565	0.565	0.565	0.582	0.564	**0.625**
SMC_212	0.120	0.211	**0.229**	0.125	0.125	0.122	0.120	0.182
SMC_213	0.222	0.487	**0.516**	0.487	0.452	0.222	0.222	0.222
SMC_214	0.148	0.148	**0.185**	0.167	0.167	0.148	**0.185**	0.167
SMC_215	0.211	0.369	0.382	**0.409**	0.390	0.375	0.214	0.226
SMC_216	0.590	0.678	0.656	0.678	0.656	**0.689**	0.600	0.590
SMC_217	0.393	0.429	0.429	**0.840**	**0.840**	0.429	0.525	**0.840**
SMC_219	0.621	0.628	0.605	0.628	0.605	**0.644**	0.598	0.621
SMC_220	0.716	**0.804**	0.798	**0.804**	0.798	0.796	0.716	0.716
SMC_221	0.306	0.306	0.306	0.440	0.440	0.333	0.306	**0.480**
SMC_222	0.237	0.136	0.185	**0.691**	0.641	0.237	0.237	0.522
SMC_223	0.431	0.426	0.426	0.227	0.227	0.433	**0.523**	0.304
SMC_224	0.129	0.129	**0.161**	0.129	**0.161**	0.129	**0.161**	0.129
SMC_225	0.357	0.231	0.346	0.754	**0.800**	0.357	0.357	**0.800**
SMC_226	0.704	0.741	0.741	0.741	**0.742**	0.741	0.704	0.704
SMC_227	**0.615**	0.611	0.611	0.611	0.611	0.594	**0.615**	**0.615**
SMC_229	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_232	0.541	0.583	**0.588**	0.500	0.500	0.571	0.541	0.483
SMC_235	**0.459**	**0.459**	**0.459**	**0.459**	**0.459**	**0.459**	**0.459**	**0.459**
SMC_236	**0.724**	0.635	0.538	0.512	0.512	0.585	0.596	0.455
SMC_237	0.576	**0.606**	**0.606**	**0.606**	**0.606**	0.556	0.576	0.576
SMC_239	0.200	0.204	0.204	**0.229**	**0.229**	0.200	0.200	0.222
SMC_241	0.389	**0.444**	**0.444**	**0.444**	**0.444**	0.417	0.389	0.389
SMC_242	0.632	**0.754**	0.737	**0.754**	0.737	0.719	0.632	0.632

Continued on next page

Table A.1: Mean of the F-measure score across all files of the *SMC* dataset for the various configurations. The best results for each file are highlighted in *grey*. (`testRes`).

File	Model							
	bsl	**ft+da**	**ft+da+pt**	**ft+da+tg**	**ft+da+tg+pt**	**ft**	**pt**	**tg**
SMC_243	0.476	0.508	0.500	**0.692**	**0.692**	0.487	0.469	0.627
SMC_244	0.177	0.177	0.177	**0.408**	**0.408**	0.177	0.231	**0.408**
SMC_248	0.489	**0.533**	0.511	**0.533**	0.511	0.489	0.489	0.489
SMC_249	0.825	0.804	**0.836**	0.804	**0.836**	0.815	0.825	0.825
SMC_251	0.548	0.548	0.548	0.583	0.583	0.548	0.548	**0.589**
SMC_252	0.956	**1.000**	**1.000**	**1.000**	**1.000**	0.970	0.956	0.956
SMC_253	0.758	**0.788**	**0.788**	**0.788**	**0.788**	**0.788**	0.758	0.758
SMC_254	0.437	0.196	0.213	0.194	0.186	**0.451**	0.447	0.356
SMC_255	0.762	0.730	0.730	**0.800**	**0.800**	0.730	0.762	**0.800**
SMC_256	0.400	0.382	0.476	0.517	**0.539**	0.400	0.400	0.508
SMC_257	**0.612**	0.592	**0.612**	0.370	0.370	0.605	**0.612**	0.370
SMC_258	0.400	0.328	0.329	0.313	0.313	**0.471**	0.400	0.400
SMC_259	**0.421**	**0.421**	**0.421**	**0.421**	**0.421**	**0.421**	**0.421**	**0.421**
SMC_260	0.454	0.482	**0.632**	0.516	0.553	0.539	0.539	0.442
SMC_261	0.526	0.475	**0.632**	0.356	0.356	0.526	0.448	0.364
SMC_262	0.500	0.724	0.730	0.702	**0.758**	0.701	0.500	0.710
SMC_263	0.484	0.484	0.495	**0.667**	**0.667**	0.516	0.379	**0.667**
SMC_264	**0.571**	0.486	0.505	0.486	0.505	0.505	0.543	**0.571**
SMC_265	0.392	0.863	**0.899**	0.409	0.408	0.457	0.388	0.387
SMC_266	0.211	0.238	0.238	0.238	**0.286**	0.250	0.246	0.256
SMC_269	0.667	0.632	0.632	**0.885**	**0.885**	0.630	0.667	0.654
SMC_271	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_272	0.643	0.643	0.643	0.953	0.953	0.643	0.643	**0.958**
SMC_273	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_274	0.000	0.988	0.988	0.988	0.988	**0.993**	0.000	0.000
SMC_275	0.506	0.678	0.678	**1.000**	**1.000**	0.513	0.506	0.408
SMC_276	0.667	**0.995**	**0.995**	**0.995**	**0.995**	0.667	0.667	**0.995**
SMC_277	0.495	0.667	0.667	**1.000**	**1.000**	0.667	0.495	**1.000**
SMC_278	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_279	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_280	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_281	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_282	**0.979**	**0.979**	**0.979**	**0.979**	**0.979**	**0.979**	**0.979**	**0.979**
SMC_283	**0.976**	**0.976**	**0.976**	**0.976**	**0.976**	**0.976**	**0.976**	**0.976**
SMC_284	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_285	0.000	**0.992**	**0.992**	**0.992**	**0.992**	**0.992**	0.000	0.000

Continued on next page

Table A.1: Mean of the F-measure score across all files of the *SMC* dataset for the various configurations. The best results for each file are highlighted in *grey*. (`testRes`).

File	Model							
	bsl	**ft+da**	**ft+da+pt**	**ft+da+tg**	**ft+da+tg+pt**	**ft**	**pt**	**tg**
SMC_286	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_287	**0.990**	**0.990**	**0.990**	**0.990**	**0.990**	**0.990**	**0.990**	**0.990**
SMC_288	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
SMC_289	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

Table A.2: Mean of the F-measure score across all files of the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations). The best results for each file are highlighted in *grey*. (`testRes`).

File	bsl	$Optimal_R$	Model
SMC_001	0.493	**0.783**	tg
SMC_002	0.359	**0.441**	ft+da+tg
SMC_003	0.369	**0.800**	ft+da+tg
SMC_004	0.438	**0.717**	ft+da+tg+pt
SMC_005	0.276	**0.410**	ft+da+tg
SMC_006	0.472	**0.786**	ft+da+tg
SMC_007	0.411	**0.542**	ft+da+tg
SMC_008	0.393	**1.000**	ft+da+tg
SMC_009	0.422	**0.852**	ft+da+tg
SMC_010	**0.895**	**0.895**	bsl
SMC_011	0.818	**0.897**	ft+da+pt
SMC_012	**0.410**	**0.410**	bsl
SMC_013	**0.863**	**0.863**	bsl
SMC_014	0.763	**0.842**	ft+da+pt
SMC_015	0.357	**0.971**	ft+da+tg+pt
SMC_016	0.696	**0.725**	ft+da
SMC_017	**0.833**	**0.833**	bsl
SMC_018	0.442	**0.512**	ft+da
SMC_019	0.441	**0.700**	ft+da+tg
SMC_021	0.615	**0.622**	ft
SMC_022	0.448	**0.730**	ft+da+tg
SMC_023	0.627	**1.000**	ft+da+tg
SMC_024	**0.473**	**0.473**	bsl
SMC_026	0.447	**0.706**	tg
SMC_027	**0.839**	**0.839**	bsl
SMC_028	0.494	**0.822**	ft+da+tg

Continued on next page

Table A.2: Mean of the F-measure score across all files of the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations). The best results for each file are highlighted in *grey*. (`testRes`).

File	bsl	$Optimal_R$	Model
SMC_030	0.500	**0.995**	ft+da
SMC_032	0.493	**0.500**	ft+da+tg
SMC_033	0.447	**0.562**	ft
SMC_034	0.867	**0.900**	ft
SMC_035	0.741	**0.778**	ft+da
SMC_036	**1.000**	**1.000**	bsl
SMC_037	0.581	**0.621**	tg
SMC_038	**0.793**	**0.793**	bsl
SMC_041	0.353	**0.576**	ft
SMC_042	0.559	**0.826**	tg
SMC_043	**0.636**	**0.636**	bsl
SMC_044	0.740	**0.767**	ft
SMC_046	**0.625**	**0.625**	bsl
SMC_047	0.617	**0.691**	ft+da+tg
SMC_048	0.391	**0.645**	ft+da+tg
SMC_051	**0.742**	**0.742**	bsl
SMC_052	0.667	**0.689**	ft
SMC_054	**1.000**	**1.000**	bsl
SMC_055	0.313	**0.687**	ft+da+tg+pt
SMC_056	0.205	**0.343**	ft+da+tg
SMC_057	0.456	**0.483**	ft+da+pt
SMC_058	0.441	**0.500**	ft
SMC_059	**0.914**	**0.914**	bsl
SMC_060	0.531	**0.562**	ft+da
SMC_061	0.500	**0.737**	tg
SMC_063	0.707	**0.756**	ft
SMC_064	0.066	**0.850**	ft+da+tg
SMC_065	**0.861**	**0.861**	bsl
SMC_066	0.265	**0.622**	ft+da+tg
SMC_067	**0.776**	**0.776**	bsl
SMC_068	**0.886**	**0.886**	bsl
SMC_069	0.762	**0.819**	pt
SMC_071	0.763	**0.868**	ft+da+pt
SMC_072	0.889	**0.903**	ft+da+tg
SMC_073	0.857	**0.928**	ft+da+pt
SMC_074	0.652	**0.804**	ft+da+tg+pt
SMC_075	**0.560**	**0.560**	bsl

Continued on next page

Table A.2: Mean of the F-measure score across all files of the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations). The best results for each file are highlighted in *grey*. (`testRes`).

File	bsl	$Optimal_R$	Model
SMC_076	0.366	**0.718**	tg
SMC_079	0.569	**0.759**	ft+da+tg
SMC_080	0.250	**0.485**	ft+da+tg
SMC_082	0.640	**0.880**	ft+da
SMC_084	0.082	**0.122**	ft
SMC_085	0.269	**0.308**	ft+da+tg
SMC_086	0.656	**0.719**	ft+da
SMC_087	0.945	**0.989**	ft+da
SMC_088	**0.561**	**0.561**	bsl
SMC_089	0.115	**0.167**	ft
SMC_092	**0.976**	**0.976**	bsl
SMC_093	0.526	**0.853**	ft+da+tg+pt
SMC_095	**0.974**	**0.974**	bsl
SMC_096	**0.923**	**0.923**	bsl
SMC_098	**0.681**	**0.681**	bsl
SMC_099	0.286	**0.464**	ft+da+tg
SMC_100	0.448	**0.517**	ft+da+pt
SMC_101	**0.673**	**0.673**	bsl
SMC_103	0.333	**0.545**	tg
SMC_104	**0.531**	**0.531**	bsl
SMC_105	0.170	**0.365**	ft+da+tg+pt
SMC_106	**0.613**	**0.613**	bsl
SMC_109	**0.613**	**0.613**	bsl
SMC_111	0.453	**0.621**	ft+da+tg
SMC_113	0.357	**0.857**	ft+da+tg
SMC_114	**0.710**	**0.710**	bsl
SMC_116	0.197	**0.244**	ft+da
SMC_117	0.242	**0.281**	ft
SMC_118	0.290	**0.333**	tg
SMC_119	0.207	**0.436**	ft+da+tg
SMC_120	**0.965**	**0.965**	bsl
SMC_121	0.192	**0.529**	ft+da+tg
SMC_124	0.485	**0.507**	ft+da+pt
SMC_126	0.875	**0.906**	ft+da
SMC_127	0.435	**0.558**	ft+da+tg
SMC_130	0.392	**0.510**	ft+da
SMC_133	0.697	**0.738**	ft+da

Continued on next page

Table A.2: Mean of the F-measure score across all files of the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations). The best results for each file are highlighted in *grey*. (`testRes`).

File	bsl	$Optimal_R$	Model
SMC_135	0.704	**0.761**	ft+da
SMC_137	0.423	**0.696**	ft+da+tg
SMC_139	0.421	**0.684**	ft+da+tg
SMC_140	0.753	**0.762**	pt
SMC_142	**0.587**	**0.587**	bsl
SMC_143	0.500	**0.523**	ft+da
SMC_146	0.519	**0.556**	tg
SMC_147	0.263	**0.739**	ft+da+tg
SMC_148	0.296	**0.385**	ft+da
SMC_149	0.583	**0.595**	pt
SMC_150	0.953	**0.963**	ft+da
SMC_151	0.711	**0.787**	ft
SMC_152	0.595	**0.658**	ft+da+tg+pt
SMC_153	0.547	**0.896**	ft+da
SMC_154	0.538	**0.615**	ft+da+pt
SMC_157	**0.263**	**0.263**	bsl
SMC_158	0.200	**0.235**	ft+da+tg
SMC_159	0.484	**0.622**	pt
SMC_161	**0.610**	**0.610**	bsl
SMC_166	0.789	**0.921**	ft+da+pt
SMC_167	0.255	**0.647**	ft+da+tg
SMC_168	0.298	**0.581**	ft+da+tg
SMC_169	0.452	**0.641**	tg
SMC_170	**0.882**	**0.882**	bsl
SMC_171	0.500	**0.784**	tg
SMC_172	**0.274**	**0.274**	bsl
SMC_173	0.533	**0.750**	tg
SMC_174	0.310	**0.333**	ft+da+pt
SMC_175	0.222	**0.333**	ft+da+tg
SMC_176	0.250	**0.317**	pt
SMC_178	0.478	**0.750**	ft+da+tg
SMC_179	0.453	**0.571**	ft+da+tg+pt
SMC_181	0.656	**0.746**	tg
SMC_182	0.771	**0.841**	ft+da+pt
SMC_184	0.727	**0.750**	ft+da
SMC_187	0.167	**0.387**	ft+da+tg
SMC_188	0.657	**0.979**	ft+da+tg

Continued on next page

Table A.2: Mean of the F-measure score across all files of the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations). The best results for each file are highlighted in *grey*. (`testRes`).

File	bsl	$Optimal_R$	Model
SMC_190	0.678	**0.800**	ft+da+tg
SMC_192	0.238	**0.323**	ft+da
SMC_193	**0.812**	**0.812**	bsl
SMC_194	0.118	**0.197**	ft+da
SMC_195	0.185	**0.255**	tg
SMC_197	0.240	**0.647**	ft+da+tg
SMC_198	0.804	**0.830**	ft
SMC_199	**0.324**	**0.324**	bsl
SMC_202	**0.211**	**0.211**	bsl
SMC_203	**0.242**	**0.242**	bsl
SMC_204	0.273	**0.379**	ft+da
SMC_205	**0.861**	**0.861**	bsl
SMC_206	0.375	**0.549**	ft+da+pt
SMC_207	0.541	**0.829**	tg
SMC_208	0.519	**0.745**	ft+da+tg+pt
SMC_209	**0.648**	**0.648**	bsl
SMC_211	**0.625**	**0.625**	bsl
SMC_212	0.120	**0.192**	ft+da+pt
SMC_213	0.222	**0.516**	ft+da+pt
SMC_214	0.148	**0.185**	ft+da+pt
SMC_215	0.211	**0.407**	ft+da+tg
SMC_216	0.590	**0.689**	ft+da
SMC_217	0.393	**0.840**	ft+da+tg
SMC_219	0.621	**0.644**	ft+da
SMC_220	0.716	**0.824**	ft+da
SMC_221	0.306	**0.480**	tg
SMC_222	0.237	**0.711**	ft+da+tg
SMC_223	0.431	**0.523**	pt
SMC_224	0.129	**0.161**	ft+da+pt
SMC_225	0.357	**0.800**	ft+da+tg+pt
SMC_226	0.704	**0.755**	ft+da+tg+pt
SMC_227	**0.615**	**0.615**	bsl
SMC_229	**1.000**	**1.000**	bsl
SMC_232	0.541	**0.588**	ft+da
SMC_235	**0.459**	**0.459**	bsl
SMC_236	**0.724**	**0.724**	bsl
SMC_237	0.576	**0.606**	ft+da

Continued on next page

Table A.2: Mean of the F-measure score across all files of the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations). The best results for each file are highlighted in *grey*. (`testRes`).

File	bsl	$Optimal_R$	Model
SMC_239	0.200	**0.229**	ft+da+tg
SMC_241	0.389	**0.444**	ft+da
SMC_242	0.632	**0.763**	ft+da
SMC_243	0.476	**0.692**	ft+da+tg
SMC_244	0.177	**0.408**	ft+da+tg
SMC_248	0.489	**0.543**	ft+da
SMC_249	**0.825**	**0.825**	bsl
SMC_251	0.548	**0.589**	ft+da+tg
SMC_252	0.956	**1.000**	ft+da
SMC_253	0.758	**0.788**	ft+da
SMC_254	0.437	**0.447**	ft
SMC_255	0.762	**0.800**	ft+da+tg
SMC_256	0.400	**0.567**	ft+da+tg+pt
SMC_257	**0.612**	**0.612**	bsl
SMC_258	0.400	**0.471**	ft
SMC_259	**0.421**	**0.421**	bsl
SMC_260	0.454	**0.632**	ft+da+pt
SMC_261	0.526	**0.632**	ft+da+pt
SMC_262	0.500	**0.752**	ft+da
SMC_263	0.484	**0.667**	ft+da+tg
SMC_264	**0.571**	**0.571**	bsl
SMC_265	0.392	**0.863**	ft+da
SMC_266	0.211	**0.286**	ft+da+tg+pt
SMC_269	0.667	**0.885**	ft+da+tg
SMC_271	**1.000**	**1.000**	bsl
SMC_272	0.643	**0.958**	tg
SMC_273	**1.000**	**1.000**	bsl
SMC_274	0.000	**0.993**	ft+da
SMC_275	0.506	**1.000**	ft+da+tg
SMC_276	0.667	**0.995**	ft+da
SMC_277	0.495	**1.000**	ft+da+tg
SMC_278	**1.000**	**1.000**	bsl
SMC_279	**1.000**	**1.000**	bsl
SMC_280	**1.000**	**1.000**	bsl
SMC_281	**1.000**	**1.000**	bsl
SMC_282	**0.979**	**0.979**	bsl
SMC_283	**0.976**	**0.976**	bsl

Continued on next page

Table A.2: Mean of the F-measure score across all files of the *SMC* dataset for the baseline (`bsl`) and a simulated *optimal* model (selected from the *reported* set of configurations). The best results for each file are highlighted in *grey*. (`testRes`).

File	bsl	$Optimal_R$	Model
SMC_284	**1.000**	**1.000**	bsl
SMC_285	0.000	**0.992**	ft+da
SMC_286	**1.000**	**1.000**	bsl
SMC_287	**0.990**	**0.990**	bsl
SMC_288	**1.000**	**1.000**	bsl
SMC_289	**1.000**	**1.000**	bsl

A.3 Additional Results for Chapter 6

A.3.1 *Section 6.1.1 – Onset Detection in Brazilian Maracatu*

In the context of onset detection in MIR, Table A.3 displays the performance of a baseline model and the full set of available configurations (ft_Conv1, ..., ft_Conv3, and ft_Tcn1, ..., ft_Tcn1024, ft) in terms of precision and recall for five different datasets (Cuica, Gonge-Lo, Mineiro, Tambor-Hi, and Tarol).

In summary, the finetuned configurations generally perform better than the baseline model in detecting a higher proportion of relevant onsets and a larger part of the relevant onsets present in the audio signal. However, there are a few cases where the baseline model achieves the best metric within a specific dataset, highlighting the importance of optimizing the models for each particular use case. Moreover, the best results across 4-in-5 instrument-adapted networks come from configurations that encompass finetuning of some or all TCN dilation levels. These observations suggest that the optimal finetuning strategy may be more complex than simply focusing on the layers closest to the musical surface.

Table A.3: Mean of the F-measure, Precision and Recall scores and sum of the #TP, #FP and #FN scores across the *Cuica*, *Gonge-Lo*, *Mineiro*, *Tambor-Hi* and *Tarol* datasets for the baseline and the finetuned (with different sets of frozen layers) approaches tested on the `TCNv1` network.

Dataset	Model	F-measure	Precision	Recall	#TP	#FP	#FN
Cuica	bsl	0.477	0.314	0.997	4,579	10,363	15
	ft_{Conv1}	0.467	0.323	0.862	3,955	8,860	639
	ft_{Conv2}	0.775	0.638	**0.999**	**4,588**	2,754	*6*
	ft_{Conv3}	0.971	0.949	0.996	4,572	257	22
	ft_{Tcn1}	0.981	0.968	0.995	4,565	162	29
	ft_{Tcn2}	0.983	0.975	0.992	4,551	123	43
	ft_{Tcn4}	0.984	**0.986**	0.982	4,501	*66*	93
	ft_{Tcn8}	0.984	0.982	0.986	4,522	90	72
	ft_{Tcn16}	**0.985**	0.984	0.985	4,521	72	73
	ft_{Tcn32}	0.948	0.910	0.992	4,551	441	43
	ft_{Tcn64}	0.964	0.937	0.994	4,560	307	34
	ft_{Tcn128}	0.953	0.918	0.993	4,560	420	34
	ft_{Tcn256}	0.982	0.972	0.992	4,549	137	45
	ft_{Tcn512}	0.975	0.959	0.993	4,560	183	34
	$ft_{Tcn1024}$	0.971	0.953	0.992	4,554	221	40
	ft	0.890	0.810	0.993	4,560	1,137	34
Gonge-Lo	bsl	0.508	0.345	**1.000**	**4,723**	9,478	*0*
	ft_{Conv1}	0.532	0.379	0.942	4,437	8,008	286

Continued on next page

Table A.3: Mean of the F-measure, Precision and Recall scores and sum of the #TP, #FP and #FN scores across the *Cuica*, *Gonge-Lo*, *Mineiro*, *Tambor-Hi* and *Tarol* datasets for the baseline and the finetuned (with different sets of frozen layers) approaches tested on the TCNv1 network.

Dataset	Model	F-measure	Precision	Recall	#TP	#FP	#FN
	ft_{Conv2}	0.993	0.987	0.999	4,717	67	6
	ft_{Conv3}	0.993	0.989	0.998	4,714	52	9
	ft_{Tcn1}	0.995	0.993	0.998	4,712	32	11
	ft_{Tcn2}	**0.998**	0.999	0.997	4,708	2	15
	ft_{Tcn4}	0.998	0.998	0.997	4,711	6	12
	ft_{Tcn8}	0.997	0.996	0.998	4,713	16	10
	ft_{Tcn16}	0.996	0.994	0.998	4,714	22	9
	ft_{Tcn32}	0.998	**1.000**	0.996	4,706	*1*	17
	ft_{Tcn64}	0.997	0.995	0.998	4,714	19	9
	ft_{Tcn128}	0.994	0.990	0.998	4,715	42	8
	ft_{Tcn256}	0.983	0.968	0.999	4,720	148	3
	ft_{Tcn512}	0.993	0.988	0.998	4,715	51	8
	$ft_{Tcn1024}$	0.988	0.977	0.999	4,718	101	5
	ft	0.966	0.936	0.999	4,718	297	5
Mineiro	bsl	0.946	0.906	0.991	17,744	1,890	174
	ft_{Conv1}	0.929	0.907	0.954	16,939	1,702	979
	ft_{Conv2}	0.946	0.937	0.957	17,028	1,089	890
	ft_{Conv3}	0.945	0.926	0.966	17,203	1,291	715
	ft_{Tcn1}	0.944	0.945	0.945	16,790	870	1,128
	ft_{Tcn2}	0.952	0.932	0.974	17,420	1,157	498
	ft_{Tcn4}	0.943	0.901	**0.993**	**17,783**	1,884	*135*
	ft_{Tcn8}	0.958	0.951	0.967	17,264	812	654
	ft_{Tcn16}	**0.972**	**0.960**	0.985	17,652	*725*	266
	ft_{Tcn32}	0.955	0.926	0.987	17,692	1,325	226
	ft_{Tcn64}	0.959	0.940	0.981	17,562	1,032	356
	ft_{Tcn128}	0.959	0.944	0.976	17,450	935	468
	ft_{Tcn256}	0.951	0.932	0.975	17,409	1,161	509
	ft_{Tcn512}	0.945	0.926	0.969	17,313	1,321	605
	$ft_{Tcn1024}$	0.940	0.929	0.954	17,011	1,213	907
	ft	0.950	0.944	0.957	17,111	965	807
Tambor-Hi	bsl	0.965	0.935	0.998	13,353	817	22
	ft_{Conv1}	0.955	0.916	0.998	13,354	1,073	21
	ft_{Conv2}	0.937	0.884	**0.999**	**13,357**	1,512	*18*
	ft_{Conv3}	0.951	0.909	0.998	13,351	1,163	24
	ft_{Tcn1}	0.957	0.920	0.998	13,355	1,024	20
	ft_{Tcn2}	0.967	0.938	0.998	13,350	785	25
	ft_{Tcn4}	0.952	0.911	0.998	13,355	1,142	20
	ft_{Tcn8}	0.962	0.930	0.997	13,340	888	35
	ft_{Tcn16}	0.968	0.940	0.997	13,347	753	28
	ft_{Tcn32}	0.967	0.939	0.997	13,343	758	32
	ft_{Tcn64}	0.968	0.941	0.997	13,345	731	30
	ft_{Tcn128}	0.971	0.948	0.996	13,324	647	51

Continued on next page

Table A.3: Mean of the F-measure, Precision and Recall scores and sum of the #TP, #FP and #FN scores across the *Cuica*, *Gonge-Lo*, *Mineiro*, *Tambor-Hi* and *Tarol* datasets for the baseline and the finetuned (with different sets of frozen layers) approaches tested on the TCNv1 network.

Dataset	Model	F-measure	Precision	Recall	#TP	#FP	#FN
	ft_{Tcn256}	0.974	0.952	0.997	13,334	617	41
	ft_{Tcn512}	0.975	0.956	0.995	13,307	570	68
	$ft_{Tcn1024}$	0.978	**0.962**	0.994	13,296	479	79
	ft	**0.978**	0.962	0.996	13,325	*477*	50
Tarol	bsl	0.993	0.987	1.000	18,577	182	8
	ft_{Conv1}	0.949	0.906	**1.000**	**18,580**	2,055	*5*
	ft_{Conv2}	0.996	0.993	1.000	18,577	114	8
	ft_{Conv3}	**0.997**	**0.994**	0.999	18,574	*81*	11
	ft_{Tcn1}	0.994	0.989	1.000	18,577	153	8
	ft_{Tcn2}	0.996	0.993	0.999	18,577	105	8
	ft_{Tcn4}	0.992	0.986	0.999	18,575	229	10
	ft_{Tcn8}	0.992	0.985	0.999	18,576	236	9
	ft_{Tcn16}	0.990	0.981	1.000	18,577	311	8
	ft_{Tcn32}	0.991	0.983	1.000	18,578	278	7
	ft_{Tcn64}	0.987	0.975	1.000	18,577	422	8
	ft_{Tcn128}	0.991	0.984	1.000	18,577	246	8
	ft_{Tcn256}	0.992	0.985	1.000	18,577	243	8
	ft_{Tcn512}	0.990	0.982	0.999	18,576	274	9
	$ft_{Tcn1024}$	0.991	0.983	0.999	18,576	265	9
	ft	0.989	0.979	1.000	18,577	322	8

Table A.4: Mean of the F-measure, Precision and Recall scores and sum of the #TP, #FP and #FN scores across the *Cuica*, *Gonge-Lo*, *Mineiro*, *Tambor-Hi* and *Tarol* datasets for the baseline and the finetuned (with different sets of frozen layers) approaches tested on the TCNv2 network.

Dataset	Model	F-measure	Precision	Recall	#TP	#FP	#FN
Cuica	bsl*	0.429	0.832	0.324	1,436	*126*	3,158
	ft_{Conv1}	0.749	0.775	0.742	3,263	899	1,331
	ft_{Conv2}	0.921	0.926	0.927	4,103	305	491
	ft_{Conv3}	0.944	**0.952**	0.942	4,203	189	391
	ft_{Tcn1}	0.944	0.937	0.957	4,294	252	300
	ft_{Tcn2}	0.943	0.933	0.961	4,320	276	274
	ft_{Tcn4}	0.945	0.931	0.966	4,359	286	235
	ft_{Tcn8}	0.946	0.930	0.969	4,385	291	209
	ft_{Tcn16}	0.948	0.928	0.975	4,427	301	167
	ft_{Tcn32}	0.948	0.926	0.976	4,438	308	156
	ft_{Tcn64}	0.949	0.926	0.978	4,456	310	138
	ft_{Tcn128}	0.950	0.927	0.978	4,460	303	134
	ft_{Tcn256}	0.950	0.930	0.976	4,444	289	150
	ft_{Tcn512}	0.952	0.932	0.976	4,447	279	147

Continued on next page

Table A.4: Mean of the F-measure, Precision and Recall scores and sum of the #TP, #FP and #FN scores across the *Cuica*, *Gonge-Lo*, *Mineiro*, *Tambor-Hi* and *Tarol* datasets for the baseline and the finetuned (with different sets of frozen layers) approaches tested on the TCNv2 network.

Dataset	Model	F-measure	Precision	Recall	#TP	#FP	#FN
	$ft_{Tcn1024}$	0.953	0.936	0.974	4,432	262	162
	ft	**0.955**	0.935	**0.979**	**4,471**	267	*123*
Gonge-Lo	bsl*	0.892	**0.960**	0.851	3,891	*159*	832
	ft_{Conv1}	0.932	0.921	0.949	4,381	355	342
	ft_{Conv2}	0.946	0.949	0.949	4,383	222	340
	ft_{Conv3}	0.940	0.941	0.945	4,356	255	367
	ft_{Tcn1}	0.940	0.944	0.942	4,339	244	384
	ft_{Tcn2}	0.944	0.946	0.947	4,395	236	328
	ft_{Tcn4}	0.946	0.945	0.951	4,442	238	281
	ft_{Tcn8}	0.950	0.943	0.961	4,491	248	232
	ft_{Tcn16}	0.946	0.941	0.956	4,478	256	245
	ft_{Tcn32}	0.947	0.942	0.956	4,480	249	243
	ft_{Tcn64}	0.947	0.942	0.957	4,486	251	237
	ft_{Tcn128}	0.949	0.943	0.960	4,492	242	231
	ft_{Tcn256}	0.948	0.945	0.956	4,460	235	263
	ft_{Tcn512}	0.952	0.945	0.964	4,511	235	212
	$ft_{Tcn1024}$	0.953	0.945	0.965	4,521	238	202
	ft	**0.956**	0.944	**0.971**	**4,554**	241	*169*
Mineiro	bsl*	0.193	**0.992**	0.114	2,063	*8*	15,855
	ft_{Conv1}	0.476	0.986	0.327	5,391	77	12,527
	ft_{Conv2}	0.466	0.963	0.315	5,306	136	12,612
	ft_{Conv3}	0.487	0.951	0.338	5,465	251	12,453
	ft_{Tcn1}	0.620	0.985	0.474	7,599	89	10,319
	ft_{Tcn2}	0.757	0.970	0.635	10,548	269	7,370
	ft_{Tcn4}	0.774	0.963	0.662	10,990	361	6,928
	ft_{Tcn8}	**0.790**	0.968	**0.681**	**11,371**	328	*6,547*
	ft_{Tcn16}	0.727	0.963	0.598	9,883	328	8,035
	ft_{Tcn32}	0.760	0.964	0.640	10,750	349	7,168
	ft_{Tcn64}	0.748	0.959	0.625	10,488	399	7,430
	ft_{Tcn128}	0.722	0.960	0.591	9,903	371	8,015
	ft_{Tcn256}	0.702	0.949	0.567	9,575	464	8,343
	ft_{Tcn512}	0.678	0.948	0.536	9,095	442	8,823
	$ft_{Tcn1024}$	0.661	0.949	0.515	8,747	409	9,171
	ft	0.675	0.954	0.531	8,996	380	8,922
Tambor-Hi	bsl*	0.443	**0.998**	0.286	3,742	*5*	9,633
	ft_{Conv1}	0.555	0.989	0.396	4,722	45	8,653
	ft_{Conv2}	0.565	0.992	0.405	4,840	35	8,535
	ft_{Conv3}	0.656	0.982	0.501	6,115	112	7,260
	ft_{Tcn1}	**0.723**	0.986	**0.578**	**7,170**	97	*6,205*
	ft_{Tcn2}	0.708	0.992	0.559	6,871	54	6,504
	ft_{Tcn4}	0.704	0.985	0.555	6,916	98	6,459

Continued on next page

Table A.4: Mean of the F-measure, Precision and Recall scores and sum of the #TP, #FP and #FN scores across the *Cuica*, *Gonge-Lo*, *Mineiro*, *Tambor-Hi* and *Tarol* datasets for the baseline and the finetuned (with different sets of frozen layers) approaches tested on the TCNv2 network.

Dataset	Model	F-measure	Precision	Recall	#TP	#FP	#FN
Tarol	ft_{Tcn8}	0.647	0.985	0.489	6,028	84	7,347
	ft_{Tcn16}	0.650	0.988	0.491	6,111	68	7,264
	ft_{Tcn32}	0.646	0.988	0.487	6,025	62	7,350
	ft_{Tcn64}	0.637	0.987	0.478	5,876	69	7,499
	ft_{Tcn128}	0.639	0.987	0.479	5,919	68	7,456
	ft_{Tcn256}	0.630	0.986	0.470	5,800	70	7,575
	ft_{Tcn512}	0.637	0.987	0.478	5,890	70	7,485
	$ft_{Tcn1024}$	0.638	0.987	0.479	5,872	67	7,503
	ft	0.643	0.988	0.485	5,947	62	7,428
	bsl*	0.139	**0.992**	0.078	1,238	*9*	17,347
	ft_{Conv1}	0.669	0.984	0.520	8,830	101	9,755
	ft_{Conv2}	0.734	0.977	0.598	10,340	183	8,245
	ft_{Conv3}	0.757	0.978	0.629	10,799	196	7,786
	ft_{Tcn1}	0.756	0.981	0.626	10,752	171	7,833
	ft_{Tcn2}	0.809	0.985	0.695	12,079	137	6,506
	ft_{Tcn4}	0.837	0.985	0.735	12,909	147	5,676
	ft_{Tcn8}	0.827	0.984	0.722	12,680	163	5,905
	ft_{Tcn16}	0.827	0.989	0.719	12,557	105	6,028
	ft_{Tcn32}	0.785	0.988	0.662	11,427	102	7,158
	ft_{Tcn64}	0.746	0.989	0.614	10,336	86	8,249
	ft_{Tcn128}	0.807	0.990	0.694	11,886	92	6,699
	ft_{Tcn256}	0.824	0.989	0.718	12,336	98	6,249
	ft_{Tcn512}	0.831	0.990	0.727	12,521	92	6,064
	$ft_{Tcn1024}$	0.848	0.990	0.751	13,028	103	5,557
	ft	**0.884**	0.990	**0.807**	**14,215**	111	*4,370*

Temporal Receptive Field Computation

We directly adapted the models developed by Davies and Böck [2019]; Böck and Davies [2020], with minor modifications such as the inclusion of an extra dilation level for TCNv1. The input layers of both TCNv1 and TCNv2 possess similar characteristics.

They utilize a 16-dimensional feature vector, derived from the log-magnitude spectrogram of the input audio signal, as the input for their dilated convolution operations rather than raw audio. The spectrogram is computed with a window and FFT size of 2048 samples, a hop size of 441 samples (equating to 100 frames per second for audio sampled at 44,100 Hz), and filtered through a bank of overlapping triangular filters with 12 bands per octave, ranging in frequency from 30 to 17,000 Hz.

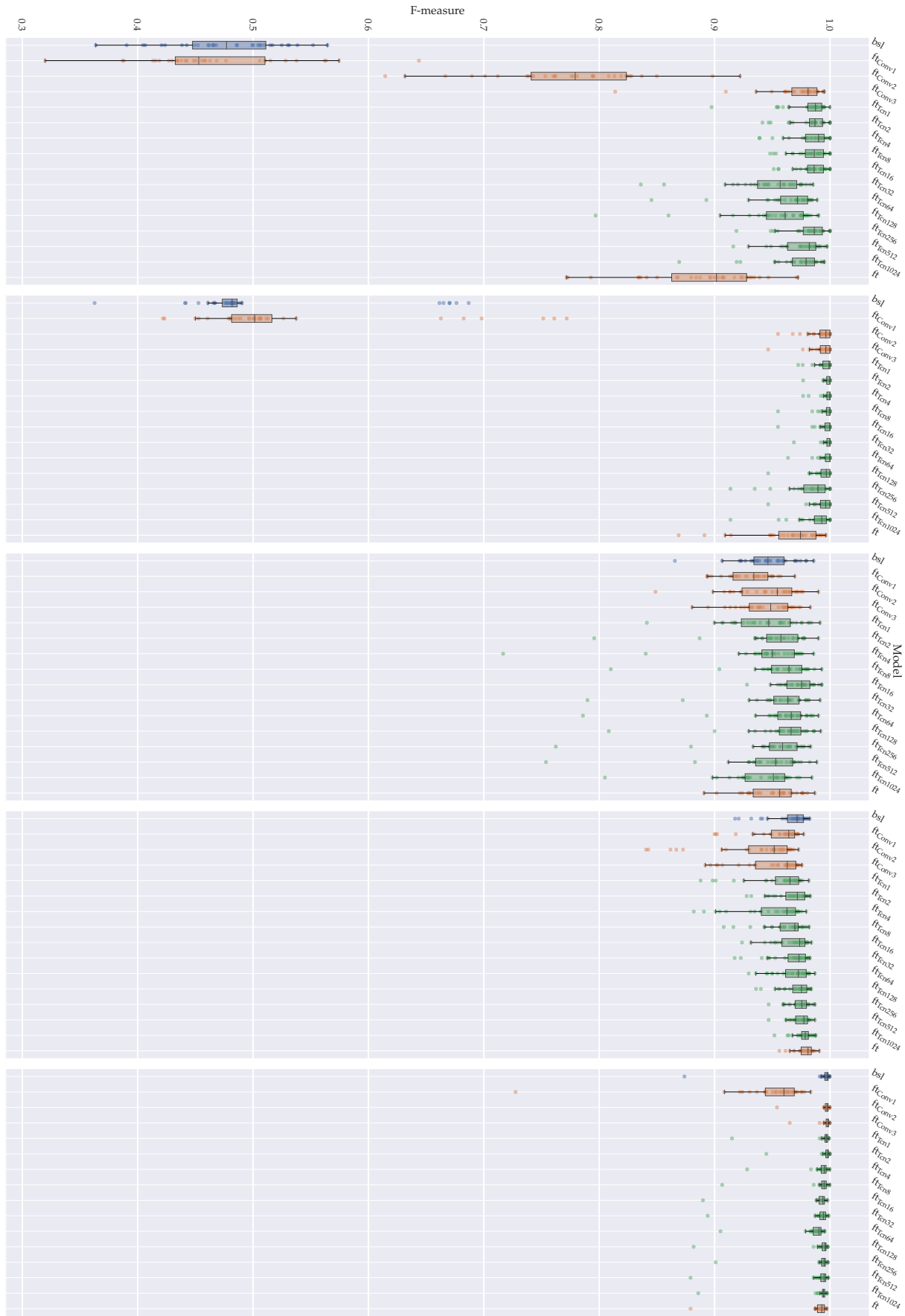


Figure A.2: Distribution of F-measure scores by all model configurations for the *Maracatu* datasets (*Inductive* Transfer Learning).

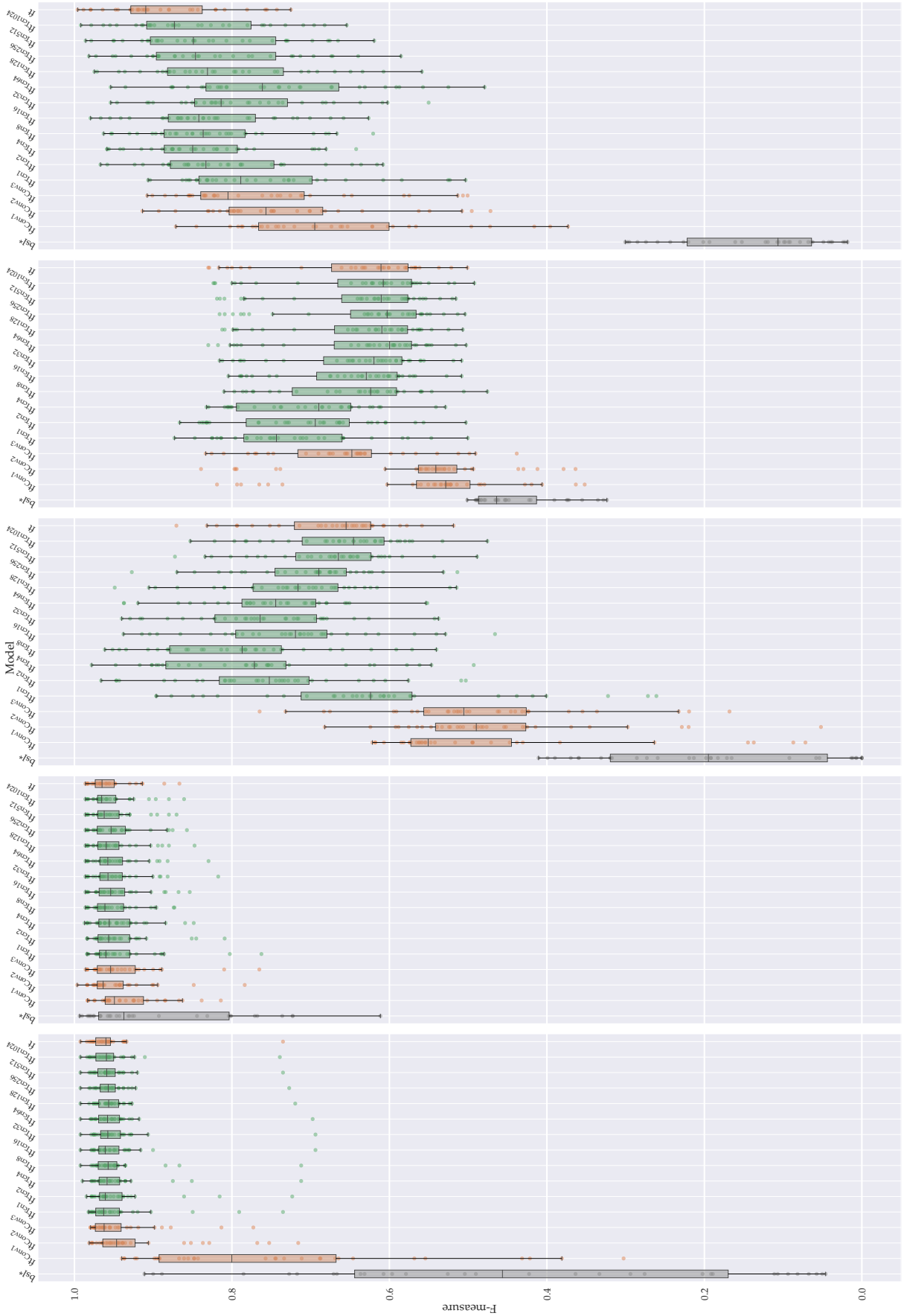


Figure A.3: Distribution of F-measure scores by all model configurations for the *Maracatu* datasets (*Transductive* Transfer Learning).

Subsequently, alternating convolutional and max pooling layers are applied to 5-frame slices, reducing the dimensionality in both time and frequency to a single dimension. The characteristics of these convolutional blocks are as follows:

- **TCNv1**: 16 filters with kernel sizes of 3x3 for the first two and 1x8 for the last layer. The first two layers involve max pooling layers that apply pooling solely in the frequency direction over 3 bins.
- **TCNv2**: 20 filters with kernel sizes of 3x3, 1x10, and 3x3, respectively, and intermediate max pooling layers that apply pooling only in the frequency direction over 3 frequency bins.

In both architecture, a dropout rate of 0.1 is used, along with the exponential linear unit (ELU) as the activation function.

To compute the temporal resolution, we follow:

$$rf(Tcn_i) = rf(Tcn_{i-1}) + [(f-1) \times d_i + 1] \tag{A.1}$$

where $rf(Tcn_i)$ represents the receptive field at the output of the i-th TCN layer, f is the filter temporal size used in the TCN layers, and d_i is the dilation rate for the i-th TCN layer. Also, $rf(Tcn_0) = rf(Conv)$, i.e., the receptive field at the beginning of the TCN block is given by the output of the convolutional block.

The following Table A.5 presents a comprehensive comparison of temporal receptive fields between the TCNv1 and TCNv2 architectures. By examining each layer, we can identify the distinct differences in the receptive fields, expressed in terms of frames and milliseconds, for both architectures.

Table A.5: Comparison of temporal receptive fields in for TCNv1 and TCNv2 architectures.

	TCNv1		**TCNv2**	
Layer	*frames*	*ms*	*frames*	*ms*
input	1	10	1	10
ft_{Conv1}	3	30	3	30
ft_{Conv2}	5	50	5	50
ft_{Conv3}	12	120	7	70
ft_{Tcn1}	16	160	11	110
ft_{Tcn2}	20	200	15	150
ft_{Tcn4}	24	240	19	190
ft_{Tcn8}	28	280	23	230
ft_{Tcn16}	32	320	27	270
ft_{Tcn32}	36	360	31	310
ft_{Tcn64}	40	400	35	350
ft_{Tcn128}	44	440	39	390
ft_{Tcn256}	48	480	43	430
ft_{Tcn512}	52	520	47	470
$ft_{Tcn1024}$	56	560	51	510

A.3.2 *Section 6.2 – An Extremely Challenging Case of Beat Tracking*

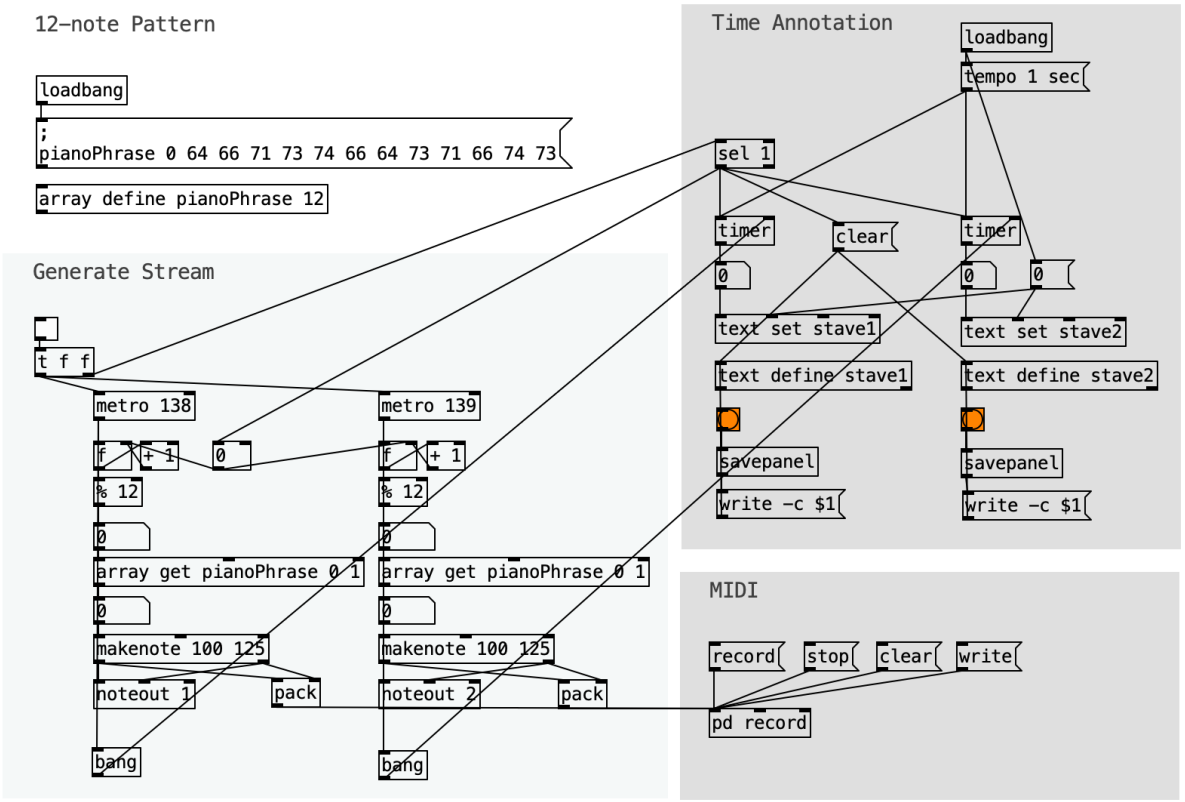


Figure A.4: *Pd* patch - generation of a simplified version of Steve Reich *Piano Phase*.

B

Appendix B – Supplementary Experiments

This appendix offers additional details on selected aspects of our user-driven approach, including the selection of the finetuning region, the optimisation of the finetuning process, and the time cost of retraining. These insights, while not central to the main thesis, provide further context to our methodology and identify potential areas for further exploration. To facilitate the distillation of key concepts, each section concludes with a *main takeaway* that encapsulates the main points discussed.

B.1 Impact of the Selection of the Finetuning Region

In this section, we further explore an essential element of our user-driven beat tracking approach: the selection of the region for user annotation. This region must embody the features of the remainder of the musical piece, as the objective of finetuning lies in enabling the network to adapt to the specific audio signal. While user discretion would be ideal for this selection, considering their beat tracking objectives and the attributes of the music, standardisation for systematic evaluation is impractical given the boundless diversity of music.

Despite the importance of this selection, our focus is not on constructing an exhaustive systematisation of potential options, but on highlighting the significant impact of the finetuning region selection on the effectiveness of our beat tracking technique.

Our dissertation outlined the rationale behind our choices for finetuning region in terms of size and location:

- *Size*: our evaluation in Chapter 4 utilised two adjacent 5 s regions for training and validation. Yet, for Chapter 5, we opted to accommodate variable file lengths across datasets by choosing a segment corresponding to 25% of each file's length, rather than a standardised 10 s region. This strategy balances the need for representative finetuning regions with our objective of minimising user annotation effort.

- *Location*: our strategy aimed to provide a fair basis for comparison with the baseline network. However, this deterministic assignment may be suboptimal if the musical content of the remaining piece differs significantly from the finetuning region.

Two finetuning strategies are explored, *Sequential* and *Percent*. The *Sequential* strategy progresses through 10 iterations, maintaining a constant retraining region length while the starting position incrementally increases, resembling a sliding window traversing each file. The *Percent* strategy also proceeds over 10 iterations but fixes the retraining region's starting position and gradually extends the length of the region. We also examine two training configurations: one includes a validation region half the size of the retraining region, while the other has no validation set and the finetuning process terminates once the pre-defined number of epochs is reached.

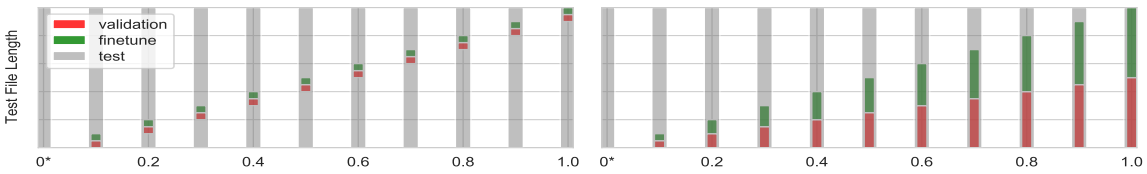
For expediency, we conducted this experiment using a small dataset comprising 9 files from the *SMC* dataset and 6 files from the *TapCorrect* dataset. The distinct characteristics of both datasets enhance our investigation: *SMC*, composed of shorter 40-seconds snippets, provides challenging beat tracking cases, whereas *TapCorrect*, with full-length files averaging over 4 minutes, includes more straightforward scenarios. In Table B.1, we list the specific files being used.

Table B.1: Lists of files for the *sub_smc* and *sub_tap* datasets.

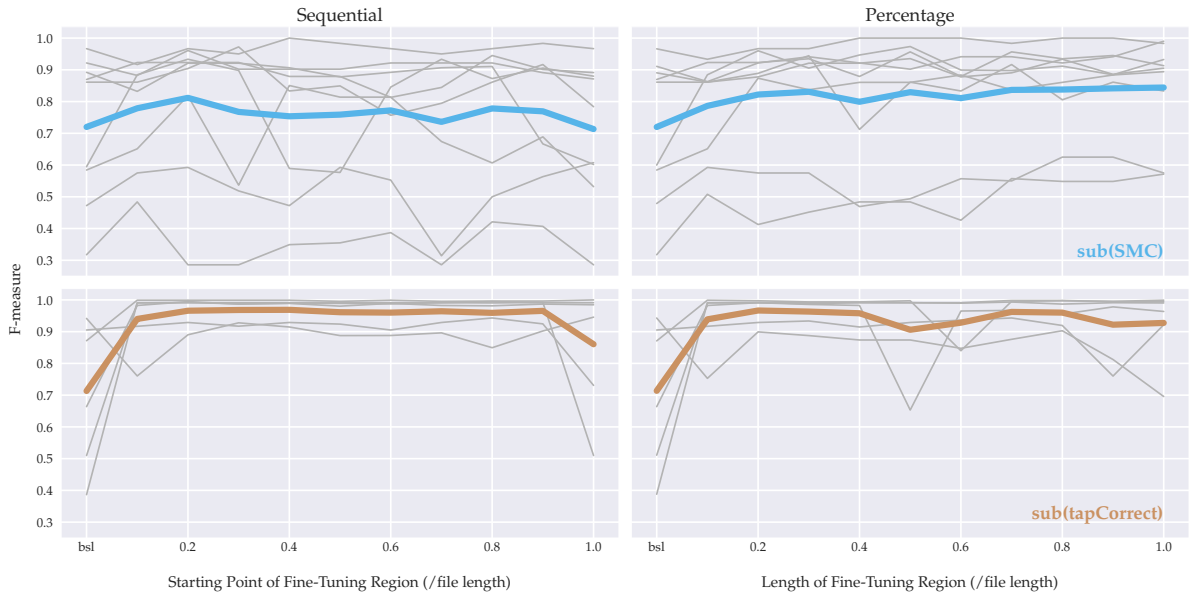
sub_smc	***sub_tap***
SMC_003	006_youtube_I3gHugP6bPE
SMC_009	014_youtube_53FGAgfYVsw
SMC_064	055_youtube_LzLt9X4hoIE
SMC_071	068_youtube_a8LcePfI6Hs
SMC_073	071_youtube_WBQ03PXMpqY
SMC_082	088_youtube_D1ACUFvxAGQ
SMC_168	
SMC_207	
SMC_252	

Our preliminary analysis, as shown in Figures B.1 and B.2, reveals a clear connection between the diversity of the finetuning regions and the resultant performances. Another

clear observation is the increase in accuracy with the percentage strategy until reaching an optimal point at 20% (with validation) or 30% (without validation) of the file length.



(a) Finetuning regions.

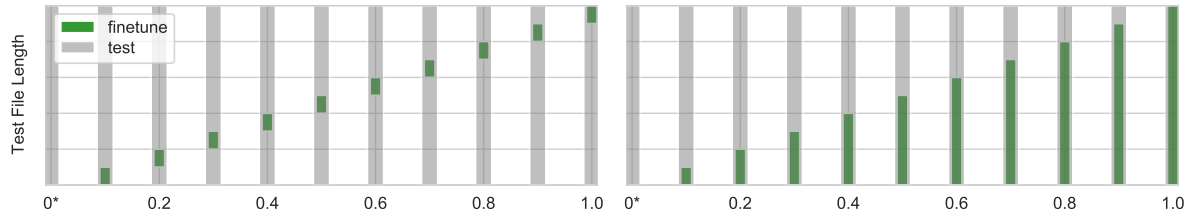


(b) Mean F-measure over the dataset.

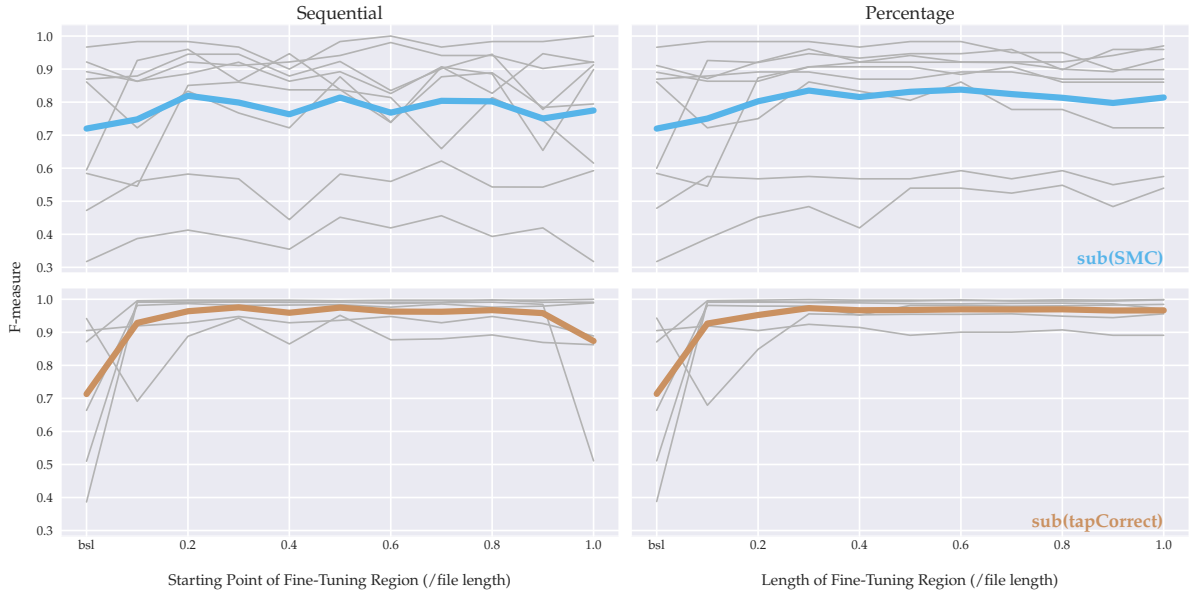
Figure B.1: Impact of the selected finetuning region in the overall accuracy: with validation. (left) regions selected in a *sequential* way; (right) regions selected as a *percentage*. These results comprise the full extension of the files (`fullRes`).

Interestingly, utilising the end portion of the music pieces for finetuning generally results in lower accuracy, potentially due to insufficient rhythmic content in these regions. Additionally, *SMC*-related files display broader accuracy ranges compared to *TapCorrect* files, reflecting the disparate attributes of these subsets.

Despite being highly exploratory, these findings underline that the selection of different finetuning regions clearly impacts the overall performance. Our basic strategy recommends an informed choice of the finetuning region by the user. However, we acknowledge the potential for additional optimisation in this process.



(a) Finetuning regions.



(b) Mean F-measure across dataset.

Figure B.2: Impact of the selected finetuning region in the overall accuracy: without validation. (left) regions selected in a *sequential* way; (right) regions selected as a *percentage*. These results comprise the full extension of the files (`fullRes`).

Main Takeaway

The selection of the finetuning region significantly impacts beat tracking performance and presents considerable potential for further optimisation.

B.2 Finetuning Optimisation Overview

In this supplementary section, we present a succinct review of the finetuning process for our TCN-based network. Figure B.3 and Figure B.4 illustrate the model performance indicators, namely training loss, validation loss, and learning rate curves, for the *TapCorrect* dataset, with and without data augmentation. This is followed by a brief analysis of these figures, highlighting the key insights and patterns observed during the process.

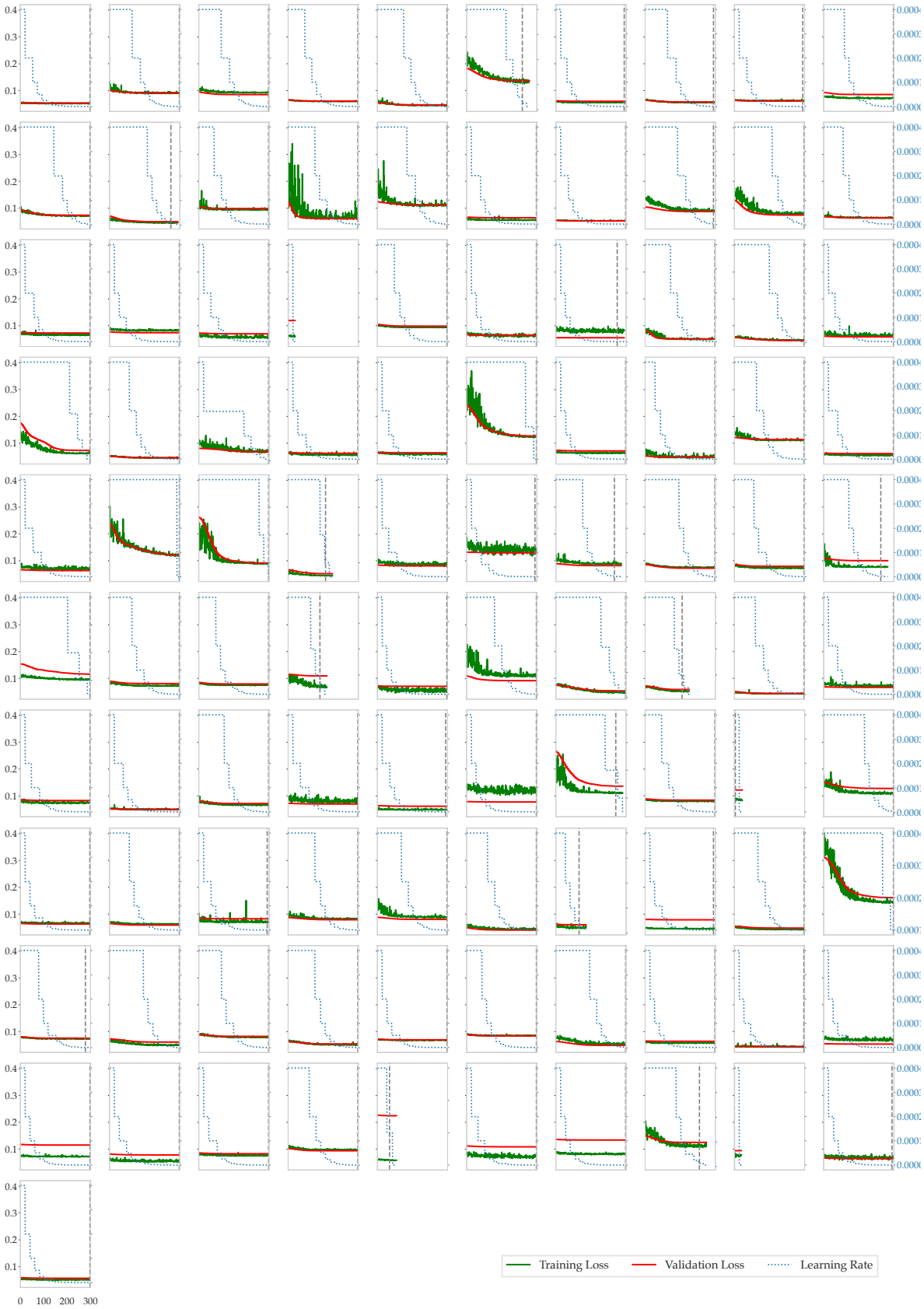


Figure B.3: Optimisation overview: without data augmentation. Training loss, validation loss, and learning rate curves for the finetuning process on the *TapCorrect* dataset.

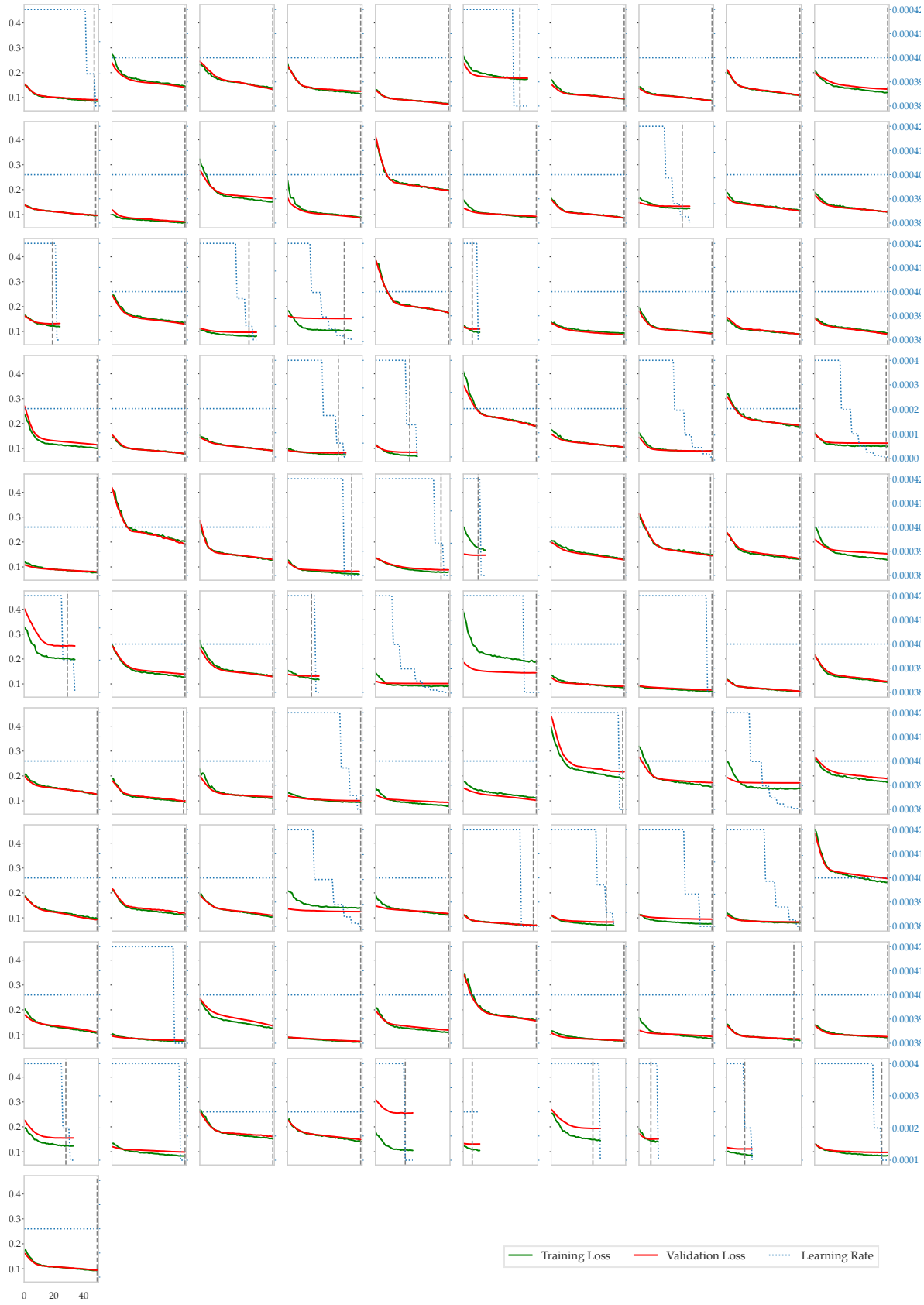


Figure B.4: Optimisation overview: with data augmentation. Training loss, validation loss, and learning rate curves for the finetuning process on the *TapCorrect* dataset.

We observe that in some instances the training loss remains essentially static, whereas in other cases, it experiences either gradual or abrupt reductions. The validation loss also presents varied behaviours, showing significant reductions in some files and minimal changes in others. Notably, when data augmentation is employed, the training loss exhibits greater variability compared to the validation loss.

The convergence behaviour of the finetuning process is deeply influenced by the particular file in use (or rather, by the relation between the characteristics of the finetuning region and the remaining signal). Some files demand the maximum allocation of epochs (50 with data augmentation, 300 without) to trigger early stopping, while others reach this threshold within fewer than 10 epochs. Moreover, variations in validation loss improvement can be observed between epochs.

Distinct adaptations in the learning rate are observed across different files throughout the training process, attributable to the *reduction on plateau* learning rate schedule.

Investigating the relationship between these observed convergence patterns and the specific data utilised offers a promising approach to adapt the training regime to the audio's characteristics and assist the user in selecting the finetuning region. This investigation could be coupled with an analysis of how different neural network layers impact the audio's high-level properties, potentially providing additional insights for enhancing the finetuning process.

Main Takeaway

The variability in the convergence of finetuning across different files suggests a potential for human-in-the-loop optimisation and justifies further investigation of underlying correlations.

B.3 Training Time

The duration of model finetuning is a key consideration, particularly in terms of practical implementation. For our human-in-the-loop strategy to be viable, adaptation must occur swiftly. However, we did not delve deeply into this aspect in our principal study, as it became clear that the time requirement for finetuning was insignificant.

Throughout our research process, we conducted numerous full-dataset finetuning experiments and, after the initial exploratory stage, we began to keep detailed records. These records encompassed various factors, including overall time spent, but also encapsulated tasks beyond finetuning, such as dataset access, debugging information

output, and evaluation, thus not strictly representing finetuning time. Despite the imprecise nature of these estimates, the consistently minimal time cost of finetuning seemed to obviate the need for further exploration. It approximated a 1:1 relation with the musical audio's duration, a ratio that seems entirely acceptable, especially considering that beat annotation in challenging signals is far more time-consuming.

To support this claim, we ran a streamlined finetuning cycle on the first five files of the *SMC* dataset. This process took about 160.47 seconds with data augmentation and 131.94 seconds without. These times correspond to finetuning 200 seconds of audio (i.e.5 files of 40 s each), indicating that our method's time requirement is around 0.7 per unit of audio time.

Main Takeaway

The time required for finetuning is shorter than the duration of the musical audio, which solidifies our approach's practicality as a swift, *in-situ* beat tracking tool.

C

Appendix C – Music Reference

Throughout this thesis, various musical works have been referenced to underscore specific themes, exemplify particular points, or to steer discussions. A majority of these pieces are drawn from the *SMC* dataset [Holzapfel et al., 2012b], with additional selections from the *Hainsworth* [Hainsworth and Macleod, 2004] and *Candombe* [Nunes et al., 2015] datasets. Additionally, two pieces were specifically chosen by us to complement our discussions and provide a broader musical context. To provide the exact versions of the works, we first used `Shazam`[56] for identification. Results are in Table C.1. We then located exact compositions on `Spotify`[57] or `YouTube`[58], ensuring readers access the referenced auditory material.

For readers unfamiliar with these works, or for those desiring to accompany our analysis, Table C.1 offers a detailed list of these compositions. Each entry provides pertinent details, such as the author and performer, along with direct links to the platforms mentioned above, facilitating immediate access to the auditory experience of each piece. In the cases of *Blue Moon* and *Proyecto 1992 - Magarinos*, we were unable to identify the exact version of the piece on any streaming platform. As a result, we have provided access to the specific files used in our experiments within our GitHub repository[59]. In a similar manner, our simplified version of *Piano Phase* is also distributed within the same repository.

[56] `https://www.shazam.com/`
[57] `https://www.spotify.com/`
[58] `https://www.youtube.com/`
[59] `https://github.com/asapsmc/MusicReferenceTable`

Song Title	Author	Performer	Dataset	File	Listen	Info	Appears in
Blue Moon	Rodgers and Hart (*1934 standard*)		*Hainsworth*	hains134	–	–	4.5.2
Choros N.1	Heitor Villa-Lobos	Kyuhee Park	–	Audio	Spotify		4.5.2
"Faust" Fantaisie Brillante	Henryk Wienawski	The Budapest Strings Chamber Orchestra	*SMC*	SMC_013	Spotify	Shazam	5.3.3
Gymnopédie N.3, II Movement	Erik Satie (Orch. by Debussy)	The Utah Symphony Orchestra	*SMC*	SMC_010	Spotify	Shazam	5.3.3
Liebestraum N.3	Franz Liszt	The Budapest Strings Chamber Orchestra	*SMC*	SMC_005	Spotify	Shazam	5.3.3
Carmen Fantasy, Op.25: IV. Allegro Moderato	Georges Bizet	The Budapest Strings Chamber Orchestra	*SMC*	SMC_002	YouTube	Shazam	5.3.3
Étude N.4	Leo Brouwer	Timo Korhonen	*SMC*	SMC_003	Spotify	Shazam	5.3.3
VI. Closing	Philip Glass	The Philip Glass Ensemble	*SMC*	SMC_008	Spotify	Shazam	5.3.3
Ghosts of Things to Come	Clint Mansell	Clint Mansell & Kronos Quartet	*SMC*	SMC_064	Spotify	Shazam	5.3.3
Montreal	Autechre	Autechre	*SMC*	SMC_285	Spotify	Shazam	5.3.3
Magarinos		Proyecto 1992	*Candombe*	Audio			6.1.3
Piano Phase (Simplified)	Steve Reich		–	Audio			6.2

Table C.1: Comprehensive list of musical works referenced throughout the thesis, with details and listening links.

References

Adams, E. (2018). What is Rhythmic Dissonance? *The Eagle Feather*, 14(1):1–22. Cited on pp. 24 and 26.

Agawu, K. and Agawu, V. K. (1995). *African Rhythm: A Northern Ewe Perspective*. Cambridge University Press. Cited on pp. 19 and 22.

Al-Maskari, A., Sanderson, M., and Clough, P. (2007). The relationship between IR effectiveness measures and user satisfaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 773–774. ACM. Cited on p. 54.

Allen, P. E. and Dannenberg, R. B. (1990). Tracking Musical Beats in Real Time. In *Proceedings of the International Computer Music Conference*, pages 140–143. Computer Music Association. Cited on p. 29.

Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2014). Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, 35(4):105–120. Cited on pp. 45 and 48.

Andersen, K. and Knees, P. (2016). Conversations with Expert Users in Music Retrieval and Research Challenges for Creative MIR. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 122–128. Cited on p. 5.

Arom, S. (1989). Time Structure in the Music of Central Africa: Periodicity, Meter, Rhythm and Polyrhythmics. *Leonardo Journal*, 22(1):91–99. Cited on p. 38.

Bååth, R. and Madison, G. (2012). The subjective difficulty of tapping to a slow beat. In *Proceedings of the 12th International Conference on Music Perception and Cognition (ICMPC)*. Cited on p. 60.

Baraldi, F. B. (2022). Envy and "corporeal lockdown" in Maracatu de baque solto (Brazil). In *Proceedings of "The Healing and Emotional Power of Music and Dance (HELP-MD)" Symposium*, pages 47–50. Cited on p. 133.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58(December 2019):82–115. Cited on p. 47.

Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. (2005). A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047. Cited on p. 25.

Bello, J. P., Rowe, R., Guedes, C., and Toussaint, G. (2015). Five Perspectives on Musical Rhythm. *Journal of New Music Research*, 441(1):1–2. Cited on p. 5.

Benadon, F. (2004). Towards a theory of tempo modulation. In *Proceedings of the 8th International Conference on Music Perception and Cognition (ICMPC)*, pages 563–566. Causal Productions, Sydney, Australia. Cited on p. 35.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 591–596. Cited on p. 45.

Bessoni e Silva, G. P. (2021). Maracatu de Baque Solto: de brincadeira a patrimônio cultural. *Caderno Virtual de Turismo*, 21(2):113. Cited on p. 133.

Bilmes, J. (1993). *Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm*. M.sc. thesis, Massachusetts Institute of Technology. Cited on pp. 17, 18, and 35.

Bittner, R. M., Fuentes, M., Rubinstein, D., Jansson, A., Choi, K., and Kell, T. (2019). Mirdata: Software for reproducible usage of datasets. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pages 99–106. Cited on pp. 32 and 173.

Böck, S. and Davies, M. E. P. (2020). Deconstruct, Analyse, Reconstruct: How To Improve Tempo, Beat, and Downbeat Estimation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pages 574–582. Cited on pp. 3, 7, 30, 37, 74, 76, 80, 81, 88, 89, 106, 109, 118, 137, 155, and 195.

Böck, S., Davies, M. E. P., and Knees, P. (2019). Multi-Task Learning of Tempo and Beat: Learning One To Improve the Other. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pages 486–493. Cited on pp. 69, 76, and 77.

Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., and Widmer, G. (2016a). madmom: A New Python Audio and Music Signal Processing Library. In *Proceedings of the 24th ACM International Conference on Multimedia (MM '16)*, MM '16, pages 1174–1178. ACM. Cited on pp. 42, 59, 63, 66, 67, and 68.

Böck, S., Krebs, F., and Schedl, M. (2012). Evaluating the online capabilities of onset detection methods. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 49–54. Cited on pp. 137 and 138.

Böck, S., Krebs, F., and Widmer, G. (2014). A Multi-model Approach to Beat Tracking Considering Heterogeneous Music Styles. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 603–608. Cited on pp. 31, 32, 58, and 60.

Böck, S., Krebs, F., and Widmer, G. (2016b). Joint Beat and Downbeat tracking with recurrent neural networks. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 255–261. Cited on pp. 31, 58, 59, 67, 68, 70, 76, 77, and 78.

Böck, S. and Schedl, M. (2011). Enhanced beat tracking with context-aware neural networks. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx)*, pages 135–139. Cited on pp. 29, 32, 36, 59, and 134.

Brown, H. M. (1980). Tactus. In Sadie, S., editor, *The New Grove Dictionary of Music and Musicians*, pages 357–358. Macmillan Publishers. Cited on p. 17.

Brown, J. C. (1993). Determination of the meter of musical scores by autocorrelation. *Journal of the Acoustical Society of America*, 94(4):1953–1957. Cited on pp. 27 and 28.

Burloiu, G. (2020). Adaptive Drum Machine Microtiming with Transfer Learning and RNNs. In *Extended Abstracts for the Late-Breaking Demo Session of the International Society for Music Information Retrieval Conference (ISMIR)*. Cited on p. 34.

Cannam, C., Landone, C., and Sandler, M. (2010). Sonic Visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the International Conference on Multimedia (MM '10)*, pages 1467–1468. ACM Press. Cited on pp. 49 and 95.

Cano, E., Mora-ángel, F., Gil, G. A. L., Escamilla, A., Alzate, J. F., and Betancur, M. (2020). Sesquialtera in the Colombian Bambuco : Perception and Estimation of Beat and Meter. In *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR)*, pages 409–415. Cited on p. 32.

Cano, E., Mora-Ángel, F., Gil, G. A. L., Zapata, J. R., Escamilla, A., Alzate, J. F., and Betancur, M. (2021). Sesquialtera in the Colombian Bambuco: Perception and Estimation of Beat and Meter – Extended version. *Transactions of the International Society for Music Information Retrieval*, 4(1):248–262. Cited on pp. 18, 19, 37, 146, and 147.

Cawley, G. (2011). Baseline Methods for Active Learning. *Proceedings of Active Learning and Experimental Design workshop In conjunction with AISTATS*, 16:47–57. Cited on p. 54.

Cemgil, A. T. and Kappen, B. (2003). Monte Carlo Methods for Tempo Tracking and Rhythm Quantization. *Journal of Artificial Intelligence Research*, 18:45–81. Cited on pp. 25 and 29.

Cemgil, A. T., Kappen, B., Desain, P., and Honing, H. (2000). On tempo tracking: Tempogram Representation and Kalman filtering. *Journal of New Music Research*, 29(4):259–273. Cited on pp. 25, 30, and 40.

Choi, J., Lee, J., Park, J., and Nam, J. (2019). Zero-shot learning for audio-based music classification and tagging. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pages 67–74. Cited on p. 34.

Choi, K., Fazekas, G., Cho, K., and Sandler, M. (2017a). A Tutorial on Deep Learning for Music Information Retrieval. *arXiv preprint : 1709.04396v1*. Cited on p. 31.

Choi, K., Fazekas, G., Sandler, M., and Cho, K. (2017b). Transfer learning for music classification and regression tasks. In *Proceedings of the 18th International Conference on Music Information Retrieval (ISMIR)*, pages 141–149. Cited on pp. 33 and 48.

Chor, I. (2010). Microtiming and Rhythmic Structure in Clave-Based Music. In Danielsen, A., editor, *Musical Rhythm in the Age of Digital Reproduction*, pages 37–50. Ashgate Publishing, Ltd. Cited on p. 35.

Christensen, E. (2004). Overt and hidden processes in 20th century music. *Axiomathes*, 14(1):97–117. Cited on p. 158.

Clarke, E. F. (1987). Levels of structure in the organization of musical time. *Contemporary Music Review*, 2(1):211–238. Cited on p. 21.

Clarke, E. F. (1999). Rhythm and Timing in Music. In Deutsch, D., editor, *The Psychology of Music*, chapter 13, pages 473–499. Academic Press, 2nd edition. Cited on pp. 14 and 22.

Collins, N. (2006). Towards a style-specific basis for computational beat tracking. In *Proceedings of the 9th International Conference on Music Perception and Cognition*, pages 461–467. Cited on pp. 36 and 68.

Cooper, G. W. and Meyer, L. B. (1960). *The Rhythmic Structure of Music*. The University of Chicago Press. Cited on pp. 14, 15, 17, and 20.

Cornelis, O., Lesaffre, M., Moelants, D., and Leman, M. (2010). Access to ethnic music: Advances and perspectives in content-based music information retrieval. *Signal Processing*, 90(4):1008–1031. Cited on p. 37.

Cowell, H. (1958). *New Musical Resources*. Cambridge University Press. Cited on p. 26.

Dalton, B., Johnson, D., and Tzanetakis, G. (2019). DAW-Integrated Beat Tracking for Music Production. In *Sound and Music Computing Conference (SMC)*. Cited on p. 73.

Danielsen, A. (2010). There, There and Everywhere: Three Accounts of Pulse in D'Angelo's 'Left And Right'. In Danielsen, A., editor, *Musical Rhythm in the Age of Digital Reproduction*, pages 19–36. Ashgate Publishing, Ltd. Cited on p. 35.

Dannenberg, R. B. (2005). Toward automated holistic beat tracking, music analysis, and understanding. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 366–373. Cited on p. 39.

Daudet, L., Richard, G., and Leveau, P. (2004). Methodology and Tools for the evaluation of automatic onset detection algorithms in music. *Proceedings of the 5th International Society for Music Information Retrieval Conference (ISMIR)*, pages 72–75. Cited on p. 134.

Davies, M. E. P. and Böck, S. (2014). Evaluating the Evaluation Measures for Beat Tracking. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 637–642. Cited on pp. 38, 40, and 67.

Davies, M. E. P. and Böck, S. (2019). Temporal convolutional networks for musical audio beat tracking. In *Proceedings of the 27th European Signal Processing Conference (EUSIPCO)*. Cited on pp. 4, 76, 81, 137, and 195.

Davies, M. E. P., Böck, S., and Fuentes, M. (2021). *Tempo, Beat and Downbeat Estimation*. https://tempobeatdownbeat.github.io/tutorial/intro.html. Cited on pp. 31, 39, 49, and 51.

Davies, M. E. P., Degara, N., and Plumbley, M. D. (2009). Evaluation Methods for Musical Audio Beat Tracking Algorithms. Technical Report October, Queen Mary University of London. Cited on pp. 39, 40, and 77.

Davies, M. E. P., Fuentes, M., Fonseca, J., Aly, L., Jerónimo, M., and Baraldi, F. B. (2020). Moving in Time: Computational Analysis of Microtiming in Maracatu de Baque Solto. In *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR)*. Cited on pp. 133, 136, 139, 144, and 145.

Davies, M. E. P., Hamel, P., Yoshii, K., and Goto, M. (2013). AutoMashUpper: An Automatic Multi-Song Mashup System. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*. Cited on p. 5.

Davies, M. E. P., Hamel, P., Yoshii, K., and Goto, M. (2014). AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1726–1737. Cited on pp. 3, 5, and 25.

Davies, M. E. P. and Plumbley, M. D. (2007). Context-Dependent Beat Tracking of Musical Audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020. Cited on pp. 3, 29, 41, and 134.

DeFord, R. I. (2015). *Tactus, Mensuration, and Rhythm in Renaissance Music*. Cambridge University Press. Cited on p. 17.

Degara, N., Rua, E. A., Pena, A., Torres-Guijarro, S., Davies, M. E. P., and Plumbley, M. D. (2012). Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech and Language Processing*, 20(1):278–289. Cited on p. 31.

der Nederlanden, C. M. V. B., Taylor, J. E. T., and Grahn, J. A. (2019). Neural Basis of Rhythm Perception. In Thaut, M. H. and Hodges, D., editors, *The Oxford Handbook of Music and the Brain*. Oxford University Press. Cited on p. 22.

Desain, P. (1992). A (de) composable theory of rhythm perception. *Music Perception*, 9(4):439–454. Cited on p. 26.

Desain, P. and de Vos, S. (1990). Autocorrelation and the study of musical expression. In *Proceedings of the International Computer Music Conference*, pages 357–360. Cited on p. 28.

Deutsch, D. and Feroe, J. (1981). The internal representation of pitch sequences in tonal music. *Psychological Review*, 88(6):503–522. Cited on p. 26.

Dhillon, G. S., Chaudhari, P., Ravichandran, A., and Soatto, S. (2020). A Baseline for Few-Shot Image Classification. In *Proceedings of the Eighth International Conference on Learning Representations (ICLR)*. Cited on p. 34.

Dixon, S. (2001a). An Interactive Beat Tracking and Visualisation System The Audio-Graphical User Interface. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 215–218. Cited on pp. 29, 36, and 73.

Dixon, S. (2001b). Automatic Extraction of Tempo and Beat From Expressive Performances. *Journal of New Music Research*, 30(1):39–58. Cited on pp. 25, 28, 35, 39, 41, 82, and 155.

Dixon, S. (2006). Onset detection revisited. In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx)*, pages 133–137. Cited on pp. 41 and 53.

Dixon, S. and Goebl, W. (2002). Pinpointing the beat: Tapping to expressive performances. In Stevens, C., Burnham, D., McPherson, G., Schubert, E., and Renwick, J., editors, *7th International Conference on Music Perception and Cognition*, number July, pages 2000–2003. Cited on p. 36.

Dixon, S., Pampalk, E., and Widmer, G. (2003). Classification of Dance Music by Periodicity Patterns. In *Proceedings of the 4th International Society for Music Information Retrieval Conference (ISMIR)*, pages 159–165. Cited on pp. 2 and 28.

Dorf, R. C. and Bischop, R. H. (2011). *Modern Control Systems*. Pearson, twelfth edition. Cited on p. 54.

Dörfler, M. (2001). Time-Frequency Analysis for Music Signals: A Mathematical Approach. *Journal of New Music Research*, 30(1):3–12. Cited on p. 15.

Downie, J. S. (2003). Music Information Retrieval. *Annual Review of Information Science and Technology*, 37:295–340. Cited on p. 1.

Downie, J. S., Ehmann, A. F., Bay, M., and Jones, M. C. (2010). The music information retrieval evaluation eXchange: Some observations and insights. *Studies in Computational Intelligence*, 274:93–115. Cited on p. 5.

Driedger, J., Schreiber, H., De Haas, W. B., and Müller, M. (2019). Towards automatically correcting tapped beat annotations for music recordings. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pages 200–207. Cited on pp. 50, 82, 89, 106, and 173.

Dudley, J. J. and Kristensson, P. O. (2018). A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems*, 8(2):1–37. Cited on pp. 48 and 172.

Ellis, D. P. W. (2007). Beat Tracking by Dynamic Programming. *Journal of New Music Research*, 36(1):51–60. Cited on pp. 29, 134, and 155.

Epstein, D. (1995). *Shaping Time: Music, the Brain and Performance*. Schirmer Books. Cited on p. 14.

Fails, J. A. and Olsen, D. R. (2003). Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM. Cited on p. 47.

Faisal, A. A., Selen, L. P., and Wolpert, D. M. (2008). Noise in the nervous system. *Nature Reviews Neuroscience*, 9(4):292–303. Cited on p. 82.

Fiebrink, R. and Cook, P. R. (2010). The Wekinator: a system for real-time, interactive machine learning in music. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, volume 4, pages 2005–2005. Cited on pp. 46 and 48.

Fiebrink, R., Cook, P. R., and Trueman, D. (2011). Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 147–156. ACM. Cited on pp. 46 and 48.

Fiebrink, R. A. and Caramiaux, B. (2018). *The Machine Learning Algorithm as Creative Musical Tool*, volume 1. Oxford University Press. Cited on pp. 44 and 45.

Fiocchi, D., Buccoli, M., Zanoni, M., Antonacci, F., and Sarti, A. (2018). Beat Tracking using Recurrent Neural Network: A Transfer Learning Approach. In *26th European Signal Processing Conference (EUSIPCO)*, pages 1915–1919. IEEE. Cited on pp. 34, 37, and 173.

Fletcher, H. and Munson, W. A. (1933). Loudness, Its Definition, Measurement and Calculation. *The Journal of the Acoustical Society of America*, 5(2):82–108. Cited on p. 39.

Flexer, A. (2014). On inter-rater agreement in audio music similarity. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 245–250. Cited on pp. 5 and 52.

Fonseca, J., Fuentes, M., Bonini Baraldi, F., and Davies, M. E. (2021). On the Use of Automatic Onset Detection for the Analysis of Maracatu de Baque Solto. In Correia Castilho, L., Dias, R., and Pinho, J., editors, *Perspectives on Music, Sound and Musicology.Current Research in Systematic Musicology, vol. 10.*, pages 209–225. Springer Cham. Cited on pp. 133 and 145.

Foote, J. and Uchihashi, S. (2001). The beat spectrum: a new approach to rhythm analysis. In *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, pages 881–884. IEEE. Cited on p. 29.

Fraisse, P. (1982). Rhythm and Tempo. In Deutsch, D., editor, *The Psychology of Music*, pages 149–180. Academic Press. Cited on pp. 14, 17, and 20.

Friberg, A. and Sundström, A. (2002). Swing Ratios and Ensemble Timing in Jazz Performance: Evidence for a Common Rhythmic Pattern. *Music Perception*, 19(3):333–349. Cited on p. 35.

Fu, Z., Lu, G., Ting, K. M., and Zhang, D. (2011). A Survey of Audio-Based Music Classification and Annotation. *IEEE Transactions on Multimedia*, 13(2):303–319. Cited on p. 47.

Fuentes, M., Maia, L. S., Rocamora, M., Biscainho, L. W., Crayencour, H. C., Essid, S., and Bello, J. P. (2019a). Tracking beats and microtiming in Afro-latin American music using conditional random fields and deep learning. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pages 251–258. Cited on pp. 25, 32, and 37.

Fuentes, M., McFee, B., Crayencour, H. C., Essid, S., and Bello, J. P. (2019b). A Music Structure Informed Downbeat Tracking System Using Skip-chain Conditional Random Fields and Deep Learning. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 481–485. IEEE. Cited on p. 31.

Galvão, M. (2014). *Metric Interplay: A Case Study In Polymeter, Polyrhythm, And Polytempo*. Master of fine arts thesis, University of California, Irvine. Cited on pp. 23, 24, and 158.

Gatty, R. (1912). Tempo Rubato. *The Musical Times*, 53(829):160–162. Cited on p. 21.

Gerischer, C. (2020). O Suingue Baiano : Rhythmic Feeling and Microrhythmic Phenomena in Brazilian Percussion. (December 2006). Cited on p. 35.

Gillies, M., Fiebrink, R., Tanaka, A., Garcia, J., Bevilacqua, F., Heloir, A., Nunnari, F., Mackay, W., Amershi, S., Lee, B., D'Alessandro, N., Tilmanne, J., Kulesza, T., and Caramiaux, B. (2016). Human-Centred Machine Learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3558–3565. ACM. Cited on p. 46.

Gjerdingen, R. O. (1989). Meter as a Mode of Attending: A Network Simulation of Attentional Rhythmicity in Music. *Intégral*, 3(1989):67–91. Cited on p. 18.

Gómez, E., Herrera, P., and Gómez-Martin, F. (2013). Computational Ethnomusicology: perspectives and challenges. *Journal of New Music Research*, 42(2):111–112. Cited on pp. 5 and 37.

Goto, M. (2001). An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds. *Journal of New Music Research*, 30(2):159–171. Cited on p. 36.

Goto, M. and Muraoka, Y. (1994). A beat tracking system for acoustic signals of music. In *Proceedings of the 2nd ACM International Conference on Multimedia (MULTIMEDIA '94)*, pages 365–372. ACM Press. Cited on pp. 3 and 29.

Goto, M. and Muraoka, Y. (1997). Issues in Evaluating Beat Tracking Systems. In *Working Notes of the IJCAI-97 Workshop on Issues in AI and Music - Evaluation and Assessment*, pages 9–16. Cited on p. 39.

Goto, M. and Muraoka, Y. (1999). Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. *Speech Communication*, 27(3-4):311–335. Cited on pp. 29 and 36.

Gouyon, F. (2005). *A Computational Approach to Rhythm Description - Audio Features for the Computation of Rhythm Periodicity Functions and their use in Tempo Induction and Music Content Processing*. Ph.d. thesis, Universitat Pompeu Fabra. Cited on pp. 16, 17, and 78.

Gouyon, F. and Dixon, S. (2005). A Review of Automatic Rhythm Description Systems. *Computer Music Journal*, 29(1):34–54. Cited on p. 2.

Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. (2006). An Experimental Comparison of Audio Tempo Induction Algorithms. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1832–1844. Cited on pp. 29 and 77.

Grosche, P., Müller, M., and Sapp, C. S. (2010). What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 649–654. Cited on pp. 3, 4, 33, and 58.

Gulluni, S., Buisson, O., Essid, S., and Richard, G. (2011). An interactive system for electro-acoustic music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 145–150. Cited on p. 48.

Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., and Feris, R. (2018). SpotTune: Transfer Learning through Adaptive Fine-tuning. *arXiv 1811.08737*. Cited on p. 172.

Hainsworth, S. (2004). *Techniques for the Automated Analysis of Musical Audio*. Ph.d. thesis, University of Cambridge. Cited on pp. 4, 41, 77, 84, 89, 92, and 106.

Hainsworth, S. (2006). Beat Tracking and Musical Metre Analysis. In Klapuri, A. and Davy, M., editors, *Signal Processing Methods for Music Transcription*, pages 101–129. Springer US. Cited on p. 25.

Hainsworth, S. W. and Macleod, M. D. (2004). Particle Filtering Applied to Musical Tempo Tracking. *EURASIP Journal on Advances in Signal Processing*, pages 2385–2395. Cited on pp. 69 and 209.

Handel, S. (1989). *Listening: An Introduction to the Perception of Auditory Events*. The MIT Press. Cited on pp. 14 and 15.

Hasty, C. (2020). *Meter as Rhythm*. Oxford University Press, 20th anniv edition. Cited on p. 18.

Herrera, P., Serrà, J., Laurier, C., Guaus, E., Gómez, E., and Serra, X. (2009). The Discipline formerly known as MIR. In *International Society for Music Information Retrieval (ISMIR) Conference - Special Session on The Future of MIR (fMIR)*. Cited on p. 2.

Herrera-Boyer, P. (2018). *MIRages: an account of music audio extractors, semantic description and context-awareness, in the three ages of MIR*. Ph.d. thesis, Universitat Pompeu Fabra. Cited on pp. 2 and 44.

Herrera-Boyer, P. and Gouyon, F. (2013). MIRrors: Music Information Research reflects on its future. *Journal of Intelligent Information Systems*, 41(3):339–343. Cited on p. 2.

Hockman, J. A., Bello, J. P., Davies, M. E. P., and Plumbley, M. D. (2008). Automated Rhythmic Transformation of Musical Audio. In *Proceedings of 11th International Conference on Digital Audio Effects (DAFx)*, pages 177–180. Cited on p. 77.

Hockman, J. A., Davies, M. E. P., and Fujinaga, I. (2012). One in the Jungle: Downbeat Detection in Hardcore, Jungle, and Drum and Bass. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 169–174. Cited on p. 36.

Holmes, T. (2008). *Electronic And Experimental Music*. Routledge, 3rd edition. Cited on p. 158.

Holzapfel, A. (2014). Tracking the "odd": meter inference in a culturally diverse music corpus. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 425–430. Cited on pp. 5 and 37.

Holzapfel, A., Davies, M. E. P., Zapata, J. R., Oliveira, J. L., and Gouyon, F. (2012a). On the automatic identification of difficult examples for beat tracking: Towards building new evaluation datasets. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 89–92. IEEE. Cited on p. 29.

Holzapfel, A., Davies, M. E. P., Zapata, J. R., Oliveira, J. L., and Gouyon, F. (2012b). Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 20(9):2539–2548. Cited on pp. 3, 4, 33, 35, 51, 58, 67, 78, 80, 89, 92, 106, and 209.

Holzinger, A. (2016). Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131. Cited on p. 46.

Honing, H. (2001). From Time to Time: The Representation of Timing and Tempo. *Computer Music Journal*, 25(3):50–61. Cited on p. 15.

Honing, H. (2013). Structure and Interpretation of Rhythm in Music. In Deutsch, D., editor, *The Psychology of Music*, pages 369–404. Elsevier, 3rd edition. Cited on pp. 14, 15, 16, 17, 18, 19, 20, and 21.

Honing, H. and Ladinig, O. (2009). Exposure influences expressive timing judgments in music. *Journal of Experimental Psychology: Human Perception and Performance*, 35(1):281–288. Cited on pp. 21 and 50.

Hornbæk, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, 64(2):79–102. Cited on p. 54.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, pages 328–339. Cited on pp. 79 and 172.

Humphrey, E. J. and Bello, J. P. (2012). Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 403–408. Cited on pp. 2 and 3.

Humphrey, E. J., Bello, J. P., and LeCun, Y. (2013a). Feature learning and deep architectures: new directions for music informatics. *Journal of Intelligent Information Systems*, 41(3):461–481. Cited on pp. 3 and 30.

Humphrey, E. J., Montecchio, N., Bittner, R., Jansson, A., and Jehan, T. (2017). Mining Labeled Data From Web-Scale Collections for Vocal Activity Detection in Music. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 709–715. Cited on p. 37.

Humphrey, E. J., Turnbull, D., and Collins, T. (2013b). A brief review of creative MIR. Cited on pp. 5 and 45.

Hunt, S. J. (2020). Exploring polyrhythms, polymeters, and polytempi with the universal grid sequencer framework. In *Proceedings of the 15th International Audio Mostly Conference*, pages 101–106. ACM. Cited on pp. 158 and 160.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, 9(3):90–95. Cited on p. 88.

Huron, D. (2006). *Sweet Anticipation*. The MIT Press. Cited on p. 22.

Iversen, J. R. and Patel, A. D. (2008). The Beat Alignment Test (BAT): Surveying beat processing abilities in the general population. In *Proceedings of the 10th International Conference on Music Perception and Cognition (ICMPC)*, pages 465–468. Cited on p. 66.

Jehan, T. (2005). *Creating music by listening*. Phd, MIT. Cited on pp. 18, 36, and 133.

Jin, C., Davies, M., and Campisi, P. (2017). Embedded Systems Feel the Beat in New Orleans: Highlights from the IEEE Signal Processing Cup 2017 Student Competition. *IEEE Signal Processing Magazine*, 34(4):143–170. Cited on p. 68.

Johansson, M. (2010). The Concept of Rhythmic Tolerance: Examining Flexible Grooves in Scandinavian Folk Fiddling. In Danielsen, A., editor, *Musical Rhythm in the Age of Digital Reproduction*, pages 69–84. Ashgate Publishing, Ltd. Cited on p. 35.

Jure, L., Marenco, B., Fuentes, M., Lanzaro, F., and Gómez, A. (2015). An Audio-Visual Database of Candombe Performances For Computational Musicological Studies. In *II Congreso Internacional de Ciencia y Tecnología Musical (CICTeM)*, pages 17–24. Cited on p. 152.

Jure, L. and Rocamora, M. (2016). Microtiming in the rhythmic structure of Candombe drumming patterns. In *Fourth International Conference on Analytical Approaches to World Music (AAWM 2016)*, pages 1–5. Cited on p. 152.

Kamar, E. (2016). Directions in hybrid intelligence: Complementing AI systems with human intelligence. *International Joint Conference on Artificial Intelligence (IJCAI)*, July:4070–4073. Cited on p. 46.

Karpathy, A., Johnson, J., and Fei-Fei, L. (2015). Visualizing and Understanding Recurrent Networks. *ArXiv 1506.02078*. Cited on p. 138.

Kendall, R. A. and Carterette, E. C. (1990). The Communication of Musical Expression. *Music Perception*, 8(2):129–163. Cited on p. 21.

Khoshrou, S., Cardoso, J. S., and Teixeira, L. F. (2015). Learning from evolving video streams in a multi-camera scenario. *Machine Learning*, 100(2-3):609–633. Cited on p. 54.

Kim, B. and Pardo, B. (2017). I-SED. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 553–557. ACM. Cited on p. 48.

Kim, B. and Pardo, B. (2018). A human-in-the-loop system for sound event detection and annotation. *ACM Transactions on Interactive Intelligent Systems*, 8(2). Cited on p. 48.

Klapuri, A. P., Eronen, A. J., and Astola, J. T. (2006). Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):342–355. Cited on pp. 2, 18, 29, 36, 41, and 155.

Kolinski, M. (1973). A Cross-Cultural Approach to Metro-Rhythmic Patterns. *Ethnomusicology*, 17(3):494. Cited on p. 5.

Korzeniowski, F., Böck, S., and Widmer, G. (2014). Probabilistic extraction of beat positions from a beat activation function. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 513–518. Cited on p. 31.

Krebs, F., Böck, S., Dorfer, M., and Widmer, G. (2016). Downbeat tracking using beat-synchronous features and recurrent neural networks. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 129–135. Cited on p. 31.

Krebs, F., Böck, S., and Widmer, G. (2013). Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 227–232. Cited on pp. 31, 36, and 77.

Krebs, F., Sebastian, B., and Widmer, G. (2015). An Efficient State-Space Model for Joint Tempo and Meter Tracking. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 72–78. Cited on pp. 32, 37, 60, and 78.

Krebs, H. (1987). Some Extensions of the Concepts of Metrical Consonance and Dissonance. *Journal of Music Theory*, 31(1):99. Cited on p. 22.

Krebs, H. (1999). *Fantasy Pieces: Metrical Dissonance in the Music of Robert Schumann*. Oxford University Press. Cited on p. 22.

Kubik, G. (2010). *Theory of African Music, Volume I*. Chicago Studies in Ethnomusicology. University of Chicago Press. Cited on p. 5.

Lamere, P. (2008). Social Tagging and Music Information Retrieval. *Journal of New Music Research*, 37(2):101–114. Cited on p. 45.

Large, E. W. and Kolen, J. F. (1994). Resonance and the Perception of Musical Meter. *Connection Science*, 6(2-3):177–208. Cited on pp. 18 and 29.

Laroche, J. (2001). Estimating tempo, swing and beat locations in audio recordings. In *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, number October, pages 135–138. IEEE. Cited on p. 28.

Lerdahl, F. and Jackendoff, R. (1981). On the Theory of Grouping and Meter. *The Musical Quarterly*, LXVII(4):479–506. Cited on p. 17.

Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press. Cited on pp. 14 and 15.

Li, M., Chen, X., Li, X., Ma, B., and Vitányi, P. M. (2004). The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264. Cited on p. 82.

London, J. (2001). Rhythm. In Stanley Sadie and Tyrrell, J., editors, *The New Grove Dictionary of Music and Musicians*, pages 277–302. Macmillan Publishers, second edition. Cited on p. 23.

London, J. (2004). *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press. Cited on pp. 17, 18, 19, and 23.

Longuet-Higgins, H. and Lee, C. S. (1982). The perception of musical rhythms. *Perception*, 11:115–128. Cited on p. 30.

Maia, L. S., Rocamora, M., and Fuentes, M. (2022). Adapting meter tracking models to latin american music. In *Proceedings of the 23th International Society for Music Information Retrieval Conference (ISMIR)*. Cited on pp. 155 and 173.

Mandel, M. I., Poliner, G. E., and Ellis, D. P. W. (2006). Support vector machine active learning for music retrieval. *Multimedia Systems,*, 12(1):3–13. Cited on p. 48.

Manilow, E. and Pardo, B. (2020). Bespoke Neural Networks for Score-Informed Source Separation. In *Extended Abstracts for the Late-Breaking Demo Session of the International Society for Music Information Retrieval Conference (ISMIR)*. Cited on p. 34.

Marchand, U. and Peeters, G. (2015). Swing Ratio Estimation. In *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx)*, pages 423–428. Cited on pp. 89 and 106.

McAdams, S. (1989). Psychological constraints on form-bearing dimensions in music. *Contemporary Music Review*, 4(1):181–198. Cited on p. 26.

Mcauley, J. D. (2010). Tempo and Rhythm. In Riess Jones, M., Fay, R. R., and Popper, A. N., editors, *Music Perception*, volume 36 of *Springer Handbook of Auditory Research*. Springer New York. Cited on p. 20.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165. Cited on p. 98.

McFee, B. (2018). Statistical Methods for Scene and Event Classification. In Virtanen, T., Plumbley, M. D., and Ellis, D., editors, *Computational Analysis of Sound Scenes and Events*, pages 103–146. Springer International Publishing,. Cited on p. 31.

Mcfee, B., Raffel, C., Liang, D., Ellis, D. P. W., Mcvicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th python in science conference*, pages 18–25. Cited on p. 49.

Mckinney, M. F. and Moelants, D. (2006). Ambiguity in tempo perception: What draws listeners to different metrical levels? *Music Perception*, 24(2):155–166. Cited on p. 52.

McKinney, M. F., Moelants, D., Davies, M. E. P., and Klapuri, A. (2007). Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms. *Journal of New Music Research*, 36(1):1–16. Cited on p. 40.

Meyer, L. B. (1973). *Explaining Music: Essays and Explorations*. University of California Press. Cited on p. 14.

Michon, J. A. (1978). The Making of the Present : A Tutorial Review. In Requin, J., editor, *Attention and Performance VII*, pages 89–111. Erlbaum. Cited on p. 20.

Mora-Ángel, F., Gil, G. L., Cano, E., and Grollmisch, S. (2019). ACMUS-MIR: A new annotated data set of Andean Colombian music. In *7th International Conference on Digital Libraries for Musicology (DLfM)*. Cited on pp. 51 and 147.

Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., and Fernández-Leal, Á. (2022). *Human-in-the-loop machine learning: a state of the art*, volume 56. Springer Netherlands. Cited on p. 47.

Müller, M. (2015). Tempo and Beat Tracking. In *Fundamentals of Music Processing*, pages 303–353. Springer International Publishing. Cited on pp. 1 and 25.

Nieto, O., Mysore, G. J., Wang, C.-i., Smith, J. B. L., Schlüter, J., Grill, T., and McFee, B. (2020). Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications. *Transactions of the International Society for Music Information Retrieval*, 3(1):246–263. Cited on pp. 25 and 53.

Nunes, L., Rocamora, M., Jure, L., and Biscainho, L. W. (2015). Beat and downbeat tracking based on rhythmic patterns applied to the Uruguayan candombe drumming. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 264–270. Cited on pp. 5, 37, 152, 155, and 209.

Oliveira, J. L., Davies, M. E. P., Gouyon, F., and Reis, L. P. (2012). Beat Tracking for Multiple Applications: A Multi-Agent System Architecture With State Recovery. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(10):2696–2706. Cited on p. 155.

Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing. T2009(06). Cited on p. 47.

Palmer, C. (1997). Music Performance. *Annual Review of Psychology*, 48(1):115–138. Cited on pp. 21 and 35.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359. Cited on pp. 33 and 34.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71. Cited on p. 99.

Parncutt, R. (1994). A Perceptual Model of Pulse Salience and Metrical Accent in Musical Rhythms. *Music Perception*, 11(4):409–464. Cited on p. 19.

Patel, A. D. (2006). Musical Rhythm, Linguistic Rhythm, and Human Evolution. *Music Perception*, 24(1):99–104. Cited on p. 4.

Patel, A. D. (2008). *Music, Language, and the Brain*. Oxford University Press. Cited on p. 26.

Paulus, J. and Klapuri, A. (2002). Measuring the Similarity of Rhythmic Patterns. In *Proceedings of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*. Cited on p. 28.

Peeters, G. (2021). The Deep Learning Revolution in MIR: The Pros and Cons, the Needs and the Challenges. In Kronland-Martinet, R., Ystad, S., and Aramaki, M., editors, *Perception, Representations, Image, Sound, Music - 14th International Symposium, CMMR 2019, Marseille, France, October 14-18, 2019, Revised Selected Papers*, volume 12631 of *Lecture Notes in Computer Science*, pages 3–30. Springer. Cited on pp. 3, 32, and 37.

Peeters, G. and Papadopoulos, H. (2011). Simultaneous Beat and Downbeat-Tracking Using a Probabilistic Framework: Theory and Large-Scale Evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1754–1769. Cited on p. 36.

Peeters, G. and Richard, G. (2021). Deep Learning for Audio and Music. In Benois-Pineau, J. and Zemmari, A., editors, *Multi-faceted Deep Learning*, pages 231–266. Springer International Publishing. Cited on p. 3.

Pinto, A. S., Böck, S., Cardoso, J. S., and Davies, M. E. P. (2021). User-Driven Fine-Tuning for Beat Tracking. *Electronics*, 10(13):1518. Cited on p. 50.

Pons, J., Serra, J., and Serra, X. (2019). Training Neural Audio Classifiers with Few Data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16–20. IEEE. Cited on p. 33.

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S. Y., and Sainath, T. (2019). Deep Learning for Audio Signal Processing. *IEEE Journal on Selected Topics in Signal Processing*, 13(2):206–219. Cited on p. 31.

Quinton, E., Harte, C., and Sandler, M. (2015). Extraction of Metrical Structure from Music Recordings. In *Proceedings of the 18th Int. Conference on Digital Audio Effects (DAFx-15)*, pages 1–7. Cited on p. 53.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. Cited on p. 31.

Rabiner, L. R. (1977). On the Use of Autocorrelation Analysis for Pitch Detection. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(1):24–33. Cited on p. 27.

Raffel, C., Mcfee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. W. (2014). mir_eval: A Transparent Implementation of Common MIR Metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 367–372. Cited on pp. 49 and 171.

Renney, N. and Gaster, B. R. (2019). Digital Expression and Representation of Rhythm. In *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound*, pages 9–16. ACM. Cited on p. 158.

Repp, B. H. (1992). Diversity and commonality in music performance: An analysis of timing microstructure in Schumann's "Träumerei". *The Journal of the Acoustical Society of America*, 92(5):2546–2568. Cited on p. 14.

Repp, B. H. (1994). On Determining the Basic Tempo of an Expressive Music Performance. *Psychology of Music*, 22(2):157–167. Cited on pp. 14 and 20.

Repp, B. H. (2005). Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12(6):969–992. Cited on p. 50.

Repp, B. H. (2006). Musical Synchronization. In Altenmüller, E., Wiesendanger, M., and Kesselring, J., editors, *Music, motor control, and the brain*, pages 55–76. Oxford University Press. Cited on p. 22.

Rex Hartson, H. (1998). Human–computer interaction: Interdisciplinary roots and trends. *Journal of Systems and Software*, 43(2):103–118. Cited on p. 46.

Richard S. Sutton, A. G. B. (2014). *Reinforcement Learning: An Introduction*. The MIT Press, second edition. Cited on p. 47.

Roads, C. (2001). *Microsound*. The MIT Press. Cited on p. 15.

Rogers, Y., Sharp, H., and Preece, J. (2023). *Interaction Design: beyond human-computer interaction*. John Wiley & Sons, Inc., 6th edition. Cited on p. 54.

Royal, M. S. (1995). *The Perception Of Rhythm And Tempo Modulation In Music*. Ph.d. thesis, The University of Western Ontario, London, Ontario, Canada. Cited on p. 20.

Sachs, C. (1953). *Rhythm and Tempo: A Study in Music History*. W. W. Norton and Co. Cited on p. 2.

Salamon, J. (2019). What's Broken in Music Informatics Research? Three Uncomfortable Statements. In *36th International Conference on Machine Learning (ICML), Workshop on Machine Learning for Music Discovery*. Cited on p. 4.

Santos, C. d. O., Resende, T. S., and Keays, P. M. (2009). *Batuque Book: Maracatu Baque Virado e Baque Solto.* Author's edition. Cited on p. 133.

Schechter, J. M., Sheehy, D. E., and Smith, R. R. (1985). Latin America. *Ethnomusicology*, 29(2):317. Cited on pp. 146 and 152.

Schedl, M. and Flexer, A. (2012). Putting the user in the center of music information retrieval. *Proceedings of the 13th International Society for Music Information Retrieval Conference, (ISMIR)*, pages 385–390. Cited on pp. 2 and 45.

Schedl, M., Flexer, A., and Urbano, J. (2013). The neglected user in music information retrieval research. *Journal of Intelligent Information Systems*, 41(3):523–539. Cited on p. 45.

Schedl, M. and Knees, P. (2013). Personalization in multimodal music retrieval. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7836 LNCS:58–71. Cited on pp. 2 and 45.

Scheirer, E. (1991). Pulse tracking with a pitch tracker. In *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, volume 89. IEEE. Cited on p. 28.

Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601. Cited on p. 29.

Schloss, W. (1985). *On the Automatic Transcription of Percussive Music - From Acoustic Signal to High-Level Analysis*. Master thesis, Stanford. Cited on p. 30.

Schlüter, J. and Böck, S. (2014). Improved musical onset detection with Convolutional Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983. IEEE. Cited on pp. 25 and 134.

Schreiber, H. and Müller, M. (2018). A single-step approach to musical tempo estimation using a convolutional neural network. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pages 98–105. Cited on p. 96.

Schreiber, H. and Müller, M. (2019). Musical tempo and key estimation using convolutional neural networks with directional filters. In *Sound and Music Computing Conference (SMC)*, pages 47–54. Cited on p. 25.

Seppanen, J. (2001). Tatum grid analysis of musical signals. In *Proceedings of the IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pages 131–134. IEEE. Cited on p. 29.

Serra, X. (2011). A Multicultural Approach in Music Information Research. *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 151–156. Cited on pp. 5 and 37.

Serra, X., Magas, M., Gómez, E., Herrera, P., Jorda, S., Flexer, A., Schlüter, J., Widmer, G., Gouyon, F., Peeters, G., Vinet, H., Benetos, E., Chudy, M., Dixon, S., Paytuvi, O., Flexer, A., Gómez, E., Gouyon, F., Herrera, P., Jorda, S., Paytuvi, O., Peeters, G., Schlüter, J., Vinet, H., and Widmer, G. (2013). *Roadmap for Music Information ReSearch*. The MIReS Consortium. Cited on pp. 2, 39, and 45.

Sethares, W. A. (2007). *Rhythm and Transforms*. Springer Science & Business Media. Cited on p. 25.

Settles, B. (2009). Active Learning Survey. Technical report, Department of Computer Sciences, University of Wisconsin-Madison. Cited on p. 47.

Settles, B. and Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, page 1070. Association for Computational Linguistics. Cited on p. 47.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control.* The MIT Press. Cited on p. 46.

Smith, L. M. and Honing, H. (2008). Time–frequency representation of musical rhythm by continuous wavelets. *Journal of Mathematics and Music*, 2(2):81–97. Cited on p. 15.

Srinivasamurthy, A., Holzapfel, A., and Serra, X. (2014). In Search of Automatic Rhythm Analysis Methods for Turkish and Indian Art Music. *Journal of New Music Research*, 43(1):94–114. Cited on p. 35.

Srinivasamurthy, A., Holzapfel, A., and Serra, X. (2017). Informed automatic meter analysis of music recordings. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 679–685. Cited on pp. 5 and 37.

Stark, A. M. and Plumbley, M. D. (2011). Performance Following: Real-Time Prediction of Musical Sequences Without a Score. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):190–199. Cited on pp. 3 and 25.

Stobart, H. and Cross, I. (2000). The Andean anacrusis? Rhythmic structure and perception in Easter songs of Northern Potosí, Bolivia. *British Journal of Ethnomusicology*, 9(2):63–92. Cited on p. 35.

Stowell, D., Robertson, A., Bryan-Kinns, N., and Plumbley, M. D. (2009). Evaluation of live human-computer music-making: Quantitative and qualitative approaches. *International Journal of Human Computer Studies*, 67(11):960–975. Cited on p. 63.

Sturm, B. L. (2013). Classification accuracy is not enough: On the evaluation of music genre recognition systems. *Journal of Intelligent Information Systems*, 41(3):371–406. Cited on pp. 5 and 53.

Taylor, S. A. (2003). Ligeti, Africa and Polyrhythm. *The World of Music*, 45(2):83–94. Cited on p. 158.

Terhardt, E. (1974). Pitch of Pure Tones: Its Relation to Intensity. In Zwicker, E. and Terhardt, E., editors, *Ear and Hearing*, volume 8 of *Communication and Cybernetics*, pages 353–360. Springer Berlin Heidelberg. Cited on p. 39.

Thaut, M. (2013). *Rhythm, Music, and the Brain*. Routledge. Cited on p. 15.

Todd, N. (1994). The Auditory "Primal Sketch": A Multiscale Model of Rhythmic Grouping. *Journal of New Music Research*, 23(1):25–70. Cited on p. 15.

Toussaint, G. (2005). The Geometry of Musical Rhythm. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3742 LNCS, pages 198–212. CRC Press. Cited on p. 35.

Toussaint, G. T. (2019). *The Geometry of Musical Rhythm*. Chapman and Hall/CRC, second edition. Cited on p. 37.

Tzanetakis, G. (2014). Computational ethnomusicology: A music information retrieval perspective. In *Proceedings of the 40th International Computer Music Conference (ICMC)*, pages 69–74. Cited on pp. 38 and 131.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302. Cited on pp. 4, 89, and 106.

Tzanetakis, G., Kapur, A., Schloss, W., and Wright, M. (2007). Computational Ethnomusicology. *Journal of Interdisciplinary Music Studies*, 1(2):1–24. Cited on pp. 4 and 38.

Urbano, J., Schedl, M., and Serra, X. (2013). Evaluation in Music Information Retrieval. *Journal of Intelligent Information Systems*, 41(3):345–369. Cited on pp. 5 and 32.

Valero-Mas, J. J. and Iñesta, J. M. (2017). Interactive user correction of automatically detected onsets: approach and evaluation. *EURASIP Journal on Audio, Speech, and Music Processing*, 2017(1):15. Cited on pp. 50, 82, and 173.

van den Oord, A., Dieleman, S., and Schrauwen, B. (2014). Transfer Learning by Supervised Pre-training for Audio-based Music Classification. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 29–34. Cited on p. 33.

Vande Veire, L. and De Bie, T. (2018). From raw audio to a seamless mix: creating an automated DJ system for Drum and Bass. *EURASIP Journal on Audio, Speech, and Music Processing*, 2018(13). Cited on pp. 3 and 25.

Wang, M. and Hua, X.-S. (2011). Active learning in multimedia annotation and retrieval. *ACM Transactions on Intelligent Systems and Technology*, 2(2):1–21. Cited on p. 48.

Wang, Y., Bryan, N. J., Cartwright, M., Pablo Bello, J., and Salamon, J. (2021). Few-Shot Continual Learning for Audio Classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 321–325. IEEE. Cited on p. 48.

Wang, Y., Salamon, J., Bryan, N. J., and Pablo Bello, J. (2020a). Few-Shot Sound Event Detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 81–85. IEEE. Cited on p. 48.

Wang, Y., Salamon, J., Cartwright, M., Bryan, N. J., and Bello, J. P. (2020b). Few-Shot Drum Transcription in Polyphonic Music. In *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pages 117–124. Cited on p. 48.

Wang, Y., Yao, Q., Kwok, J., and Ni, L. M. (2019). Generalizing from a Few Examples: A Survey on Few-Shot Learning. *arXiv 1904.05046*. Cited on p. 34.

Whiteley, N., Cemgil, A. T., and Godsill, S. (2006). Bayesian modelling of temporal structure in musical audio. *ISMIR 2006 - 7th International Conference on Music Information Retrieval*, (January):29–34. Cited on pp. 29 and 36.

Wright, M., Schloss, W. A., and Tzanetakis, G. (2008). Analyzing Afro-Cuban rhythm using rotation-aware clave template matching with dynamic programming. *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR)*, pages 647–652. Cited on p. 36.

Xu, W. (2019). Toward human-centered AI. *Interactions*, 26(4):42–46. Cited on p. 46.

Yamamoto, K. (2021). Human-in-the-Loop Adaptation for Interactive Musical Beat Tracking. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*. Cited on p. 34.

Yeston, M. (1976). *The Stratification of Musical Rhythm*. Yale University Press. Cited on pp. 14 and 15.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NIPS2014)*, volume 27. Cited on p. 78.

Zapata, J. R., Holzapfel, A., Davies, M. E., Oliveira, J. L. J. L., and Gouyon, F. (2012). Assigning a Confidence Threshold on Automatic Beat Annotation in Large Datasets. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 157–162. Cited on pp. 30 and 60.

Zatorre, R. J., Belin, P., and Penhune, V. B. (2002). Structure and function of auditory cortex: music and speech. *Trends in Cognitive Sciences*, 6(1):37–46. Cited on p. 26.

Zhou, B., Bau, D., Oliva, A., and Torralba, A. (2019). Interpreting Deep Visual Representations via Network Dissection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2131–2145. Cited on p. 138.