More Beats! Fail Case Analysis, Data Augmentation Strategies and Fine Tuning Beat Prediction Models

Milo Beuzeval, Navid Hallajian, Arhan Vohra

***Abstract*—We reproduce results in the state of the art *Beat This!* beat tracker, providing an in-depth evaluation of weaknesses in the model on the existing test set as well as biases induced from the limited training set. We find that the model struggles to track beats and downbeats for music with non-4/4 time signatures or internal tempo variations.**

Given weaknesses in the results, we implement data augmentation strategies based on beat interval removal and segmented time-stretching to emulate different time signatures and tempo variations. We fine-tune the transformer model with these augmentations and report our results.

I. Introduction

Beat tracking is the task of predicting the temporal locations of the "pulse" of a song, the points where a listener might tap their foot. Downbeat detection identifies the first beat of each bar. Automatic beat detection has been approached from many angles, from signal processing techniques [1] to supervised [2] and self-supervised [3] machine learning models. Post-processing supervised techniques with Dynamic Bayesian Networks (DBN) has been the dominant approach [4]–[6]. However recently, the *Beat This!* model proposed a DBN-less approach using a transformer based architecture and bespoke loss function [7] achieving state of the art performance in F-measure, albeit on a limited GTZAN test set [8]. Deep learning approaches like this require a large amount of annotated data for good general performance. Beat and downbeat annotations require lots of time and expertise making them difficult and expensive to produce. Recent efforts have tried to address this by exploring new data augmentation techniques to increase the size of training sets [9], [10] to improve generalised performance. An additional downside of DL models, and particularly *Beat This!* since it is composed of 6 stacked transformer blocks, is that they are computationally expensive to train from scratch. Thus, cheaper fine-tuning approaches are a good way of making use of public model weights and improving their performance for specific types of music [11].

Inspired by the challenge posed by the *Beat This!* authors to enhance their model, along with the aforementioned approaches to data augmentation and fine-tuning in the beat and downbeat prediction task, we explore the following areas in this work:

1) Analysis of the training data used by *Beat This!* to understand potential biases.
2) Fail case analysis of the *Beat This!* results to find areas for improvement.
3) Testing the time-signature augmentation approach from *Skip That Beat* [9] on the transformer model.
4) Formulating and implementing a novel time stretching augmentation to capture tempo variation characteristics identified during our analysis.
5) Fine-tuning the transformer model with the augmented data and discuss our findings.

II. Methodology

A. Time Signature Analysis

We analyse the time signatures of pieces across the train and test sets used in *Beat This!* to establish the distributions, and to try to distinguish whether it has an effect on performance and potential areas of improvement.

1) Training data: The *Beat This!* training set comprises data from 18 sources commonly used in MIR tasks, primarily covering Western popular music, classical, and jazz. It also includes artist-specific datasets, such as the Beatles dataset, highlighting the scarcity of annotated data in this task. Additionally, drum heavy datasets of HJDB [12] and Candombe [13] are used. For comprehensive details, we refer the reader to the dataset section of the *Beat This!* paper [7] or its repository[1]. The datasets provide beat and, in most cases, downbeat annotations, enabling us to infer time signatures.

We find that the distribution of time signatures in the dataset is highly skewed towards 4/4 time signature (almost 70%) see Figure 1. While 4/4 is dominant in Western music, this imbalance introduces bias, which poses a challenge for a model aiming for "generality across a diverse music range". Despite this, there are only a limited number of datasets available with beat and downbeat annotations, and in light of their stated goal, the authors would very likely make use of a wider range of time signatures if they were available. Hence, to address this imbalance, we would like to explore the data augmentation approaches proposed [9] that can supplement under-represented time signatures such as 2/4 that have far less representation in the training set (4.50%). We aim to fine-tune the model with these augmentations and evaluate their impact on specific time signatures as well as overall performance.

2) Test data: GTZAN is used as the test set in the Beat This! paper. Similarly to the training set, it consists primarily of 4/4 pieces (93%) see Figure 2. We reproduce the performance of the provided final checkpoint of the *Beat This!* model on the GTZAN test set (993 tracks, since one is not annotated and 6 exclude downbeat annotations) by following the reproduction instructions provided in the GitHub repository for their project. We record the performance metrics from mir-eval [14] for each track in a modified script [2].

We compare the `mir-eval` performance metrics across different time signatures in the GTZAN test set for the final checkpoint. Figure 5 shows the F-measure beat scores, where 4/4 is generally the best-predicted time signature. However, numerous outliers indicate that 4/4 alone is not a reliable predictor of strong model performance. Due to the limited representation of other time signatures in the test set, assessing the model's true performance on them is challenging. This is evident in the wide range of results for 3/4, which has only 54 instances. Notably, 2/4—represented by just seven instances—has the lowest median F-measure score. Given this, we focus on fine-tuning and augmentation strategies to improve the model's performance on 2/4.

B. Tempo Variation Analysis

1) Test Data: In examining the test data with the lowest beat-tracking performance from the GTZAN dataset, we found that many of them exhibit either (1) flexible (or rubato) tempo characteristics, or (2) tempo changes between sections. To quantitatively verify these observations, we used the standard deviation in inter-beat interval (IBI) from the annotations as a measure of variation in tempo across each song. Figure 6 shows a plot of the mean F-measure for tracks in each genre, plotted against mean tempo variation. We can see that test performance was significantly lower for genres with higher tempo variance on average.

The authors of *Beat This!* indicated that tempo rubato annotations were removed from the training data for unspecified reasons, but we observed that such songs exist in the test data nonetheless, experiencing poor performance overall. For example, `gtzan_rock_00040` exhibits near-constant changes in tempo, leading to the predicted beat "lagging" behind the actual beat during sections where the performer speeds up. In `gtzan_classical_00051`, which contains a tempo change halfway through the clip, we can see in Figure 4 that the model struggles to interpolate implicit beat positions between clear instrumental onsets during the slower, dynamically soft section.

2) Training Data: On average, the training data used by *Beat This!* exhibited more tempo variation than the GTZAN test set (mean IBI SD $= 0.039$ for training, vs. 0.016 for GTZAN). However, this variability was concentrated in a few specific datasets. For example, the ASAP dataset, composed of expressive classical piano performances, showed high tempo flexibility (mean IBI SD ≥ 0.175), while GuitarSet, which contains rhythmically steady guitar comping, had near-zero variation in beat spacing. To further investigate this relationship, we plot the mean F-measure score per dataset against the mean tempo variation (IBI SD) (see Fig 7) and observed a clear negative correlation: datasets with greater tempo variability see systematically lower beat-tracking performance. This suggests that exposure to variable tempo is critical for learning robust temporal representations, and that the model's performance drop in tempo-flexible music is not merely an artifact of noise or annotation quality, but a systemic gap in training coverage.

C. Data Augmentation

Our data augmentation strategies were designed in direct response to the performance limitations observed in our data and fail case analysis, particularly regarding time signature misclassification and tempo instability. We implemented two augmentation methods: a time-signature perturbation approach based on *Skip That Beat* [9], and a novel time-stretching approach aimed at improving model robustness to local tempo deviations.

1) Beat Interval Removal (Skip That Beat): Morais et al. highlight how most beat and downbeat tracking datasets are dominated by 4/4 meter, which restricts model generalisation to styles such as samba (2/4) or waltz (3/4). They propose generating synthetic 2/4 and 3/4 versions of existing 4/4 tracks by removing selected beat intervals while preserving musical continuity.

We apply the Skip That Beat procedure to the Candombe dataset, which originally consists only of 4/4 tracks. This yields modified versions in 2/4 and 3/4. Since Candombe is rhythmically dense and features regular percussion patterns, we hypothesised that it would respond well to this form of beat deletion. These augmented tracks are used to fine-tune the model with the goal of improving downbeat tracking in 2/4 test material such as BRID [9]. Since the effect of incorrect time-signature prediction is incorrect downbeat detection, we focus on downbeat metrics. We then assess the performance of the fine-tuned model on the GTZAN test set seeing how it performs on a majority 4/4 dataset.

2) Tempo Variation via Segmented Time-Stretching: In our fail case analysis, we found that the model often struggled to track beats in rubato passages or songs with mid-track tempo changes—particularly in genres like blues and classical. To address this, we designed a custom tempo augmentation procedure that introduces localised tempo shifts by selectively time-stretching one part of a track while leaving the rest unaltered.

We applied this method to the GuitarSet dataset, which contains metrically stable, low-variation guitar performances. This made it an ideal base for injecting artificial tempo dynamics. The process is repeated several times per track, creating multiple augmented versions with different stretch points and factors. An outline of this method is given in Algorithm 1. As guitar-oriented blues tracks had some of the lowest performance outcomes in the test set, we thought that fine-tuning on augmented versions of the GuitarSet training data could help us make targeted improvements to these tracks.

D. Fine-tuning

We adapt the training script from *Beat This!* to allow loading the parameters defining the network and its weights from a given checkpoint [4]. We also modify it to only train on a subset of data when specified. In our experiments, we only fine-tuned on one augmented dataset at a time. This means that beyond the pre-trained checkpoint, the model only learns to fit the augmented data from a single dataset. Both fine-tunings run for 100 epochs. In both cases the full network is being trained

without any layer freezing. Ideally, we would have trained with the full training data in addition to our augmented data when training the full network, but we had time and computational constraints.

III. Results

A. Fail case Analysis Results

In our fail case analysis on the GTZAN test set, we make a number of key findings:

1) Poor performance on tracks with expressive tempo (rubato) or tempo changes
2) Genres that feature fewer tracks with percussive instrumentation or explicit expressions of the beat (such as classical and blues) had lower F-measures on average.
3) Pieces with 4/4 time signature are the best predicted (see Figure 5), but limited examples of other time signatures, such as 2/4, make it hard to assess true performance.

TABLE I
F-measures for different time signature(s) (TS)

TS	F-measure (Beat)	F-measure (Downbeat)	Track Count
2/4	0.6665	0.5966	7
3/4	0.7916	0.6839	54
4/4	0.8937	0.7848	930
5/4	0.9919	0.3379	2

B. Fine Tuning Results

1) Beat Interval Removal Results: We compare the performance of *Beat This!* before and after fine-tuning on a time signature-augmented Candombe dataset, evaluating both on the BRID dataset.

TABLE II
Comparison of Beat This! and Candombe fine-tuned model performance on BRID dataset

Metric	Original	Fine-tuned	Improvement
F-measure_beat	0.9575	0.9432	-1.43%
F-measure_downbeat	0.5155	0.5290	1.35%
Cemgil_downbeat	0.6268	0.6609	3.41%
CMLt_downbeat	0.1325	0.2688	13.64%
AMLt_downbeat	0.5458	0.6827	13.69%

Beat This! excels at beat detection but struggles with downbeats, as shown by its downbeat F-measure being half that of beats. This suggests that while potential downbeats are correctly identified as beats, nearly half are not recognized as downbeats. The CMLt score is a continuity metric assessing the ratio of the longest continuously correct segment to the length of the track [15]. It being this low means we don't tend to get successive correct downbeat detections. The combination of these scores shows the issue of assuming the tracks are in 4/4 and thus missing half of the downbeats, which we confirm by listening to the misclassifications.

The fine-tuned model shows a 13.64% increase in downbeat CMLt. From analysing the performance on specific tracks, we see that in cases of better downbeat detection, it's because the correct 2/4 time signature is being determined, explaining the increased CMLt score. The trend in analysing tracks with decreasing performance is that increasing portions of the track are misclassified at the same incorrect time signature as in the original model. The network also classifies a larger range of events as downbeats, often making multiple downbeat detections in a row in portions of the track like drum fills where every beat is prominent. Another common error is a sound being played at double the tempo of the beat and the model locking onto that as the tempo.

TABLE III
Comparison of Beat This! and Candombe fine-tuned model performance on GTZAN

Metric	Original	Fine-tuned	Improvement
F-measure_beat	0.8921	0.8877	-0.44%
F-measure_downbeat	0.7870	0.7275	-5.95%
Cemgil_downbeat	0.7390	0.6978	-4.12%
CMLt_downbeat	0.6776	0.5015	-17.61%
AMLt_downbeat	0.7990	0.7143	-8.47%

Improvements in CMLt do not carry over to the GTZAN test and we see a decrease of -17.61%. Decreased continuity suggests time signature issues. From listening to the errors, we see that the most common error is incorrectly assuming a 2/4 time signature. This demonstrates the limitation of our fine-tuning approach. Training on an augmented dataset with no 4/4 music means losing some of the knowledge the model had about 4/4 classification, favouring the time signatures it has seen. We believe that training alongside the full training set would remedy this.

In cases where the model performs better, it does so as a side effect of predicting a time signature of 2/4. As an example, when analysing `gtzan_reggae_00064` both the original and finetuned models predict half of the actual tempo. But since the new model predicts a time signature of 2/4, it predicts twice the number of downbeats as the original model that correctly predicted 4/4, 'accidentally' making the downbeats correct.

TABLE IV
Comparison of Beat and Downbeat Tracking Performance on GTZAN and GuitarSet-Timeshift

Metric	Original	Fine-tuned	Improvement
F-measure_beat	0.8921	0.7926	-9.96%
Cemgil_beat	0.8217	0.7215	-10.02%
CMLt_beat	0.7982	0.6032	-19.50%
AMLt_beat	0.9026	0.6887	-21.39%
F-measure_downbeat	0.7870	0.6543	-13.28%
Cemgil_downbeat	0.7390	0.6226	-11.64%
CMLt_downbeat	0.6776	0.4594	-21.82%
AMLt_downbeat	0.7990	0.6073	-19.17%

2) Tempo Variation via Segmented Time Stretching: The performance of *Beat This!*, fine-tuned with time-stretching augmentation on GuitarSet, was evaluated on GTZAN. Results show an overall decline in beat and downbeat metrics, though some tracks in specific genres improved 10. Notably, blues—initially among the weakest genres—saw the smallest performance drop, suggesting some potential for augmentation

in genres such as blues which feature expressive-timing. We suspect the specifics of our implementation hindered better results and plan to refine parameters in future work, such as the extent and frequency of time-stretching, to reflect the characteristics of real compositions with sectional tempo changes.

IV. Conclusion and Future Work

In this work, we analysed failure cases in transformer-based beat tracking and explored two targeted data augmentation strategies to address these shortcomings. Our aim was to improve model performance on under-represented time-signatures and tempo-varying music without re-training from scratch.

Beat interval removal augmentation provided reasonable improvement for the time-signatures seen in the augmented set by making the network better predict continuous downbeats in 2/4 time signatures. This finding validates the possibility of fine-tuning pre-trained transformer based models to a chosen time-signature or style of music. Particularly useful for a network such as Beat This! which is computationally expensive to train and a task where annotated data is difficult to obtain. Performance on the original test set dropped due to the dominance of 4/4 in its tracks, so in future work we suggest exploring the effect of training on the full training set in addition to the augmented data and freezing earlier layers during fine-tuning.

Our time-stretching augmentation provided a novel approach to producing training data for beat tracking tasks with internal tempo variations. However, we were unable to improve the transformer model's performance by fine-tuning on this data, perhaps due to the extreme nature of the augmentations applied. Constraining the time-stretching parameters and re-training the model with this data may provide better results on the evaluation set overall.

References

[1] P. Grosche and M. Müller, "A mid-level representation for capturing dominant tempo and pulse information in music recordings." in *ISMIR*, 2009, pp. 189–194.

[2] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *Proc. Int. Conf. Digital Audio Effects*, 2011, pp. 135–139.

[3] D. Desblancs, V. Lostanlen, and R. Hennequin, "Zero-note samba: Self-supervised beat tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2922–2934, 2023.

[4] N. Whiteley, A. T. Cemgil, and S. J. Godsill, "Bayesian Modelling of Temporal Structure in Musical Audio." in *ISMIR*, 2006, pp. 29–34. [Online]. Available: https://archives.ismir.net/ismir2006/paper/000044.pdf

[5] F. Krebs, S. Böck, and G. Widmer, "An efficient state-space model for joint tempo and meter tracking." in *ISMIR*, 2015, pp. 72–78.

[6] S. Böck and M. E. P. Davies, "DECONSTRUCT, ANALYSE, RECONSTRUCT: HOW TO IMPROVE TEMPO, BEAT, AND DOWNBEAT ESTIMATION," 2020.

[7] F. Foscarin, J. Schlüter, and G. Widmer, "Beat this! Accurate beat tracking without DBN postprocessing," Jul. 2024, arXiv:2407.21658 [cs]. [Online]. Available: http://arxiv.org/abs/2407.21658

[8] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, Jul. 2002. [Online]. Available: https://ieeexplore.ieee.org/document/1021072/

[9] G. Morais, B. McFee, and M. Fuentes, "Skip That Beat: Augmenting Meter Tracking Models for Underrepresented Time Signatures," Feb. 2025, arXiv:2502.12972 [cs]. [Online]. Available: http://arxiv.org/abs/2502.12972

[10] C.-Y. Chiu, J. Ching, W.-Y. Hsiao, Y.-H. Chen, A. W.-Y. Su, and Y.-H. Yang, "Source Separation-based Data Augmentation for Improved Joint Beat and Downbeat Tracking," Jun. 2021, arXiv:2106.08703 [cs]. [Online]. Available: http://arxiv.org/abs/2106.08703

[11] L. S. Maia, M. Rocamora, L. W. P. Biscainho, and M. Fuentes, "Adapting Meter Tracking Models to Latin American Music," Apr. 2023, arXiv:2304.07186 [cs]. [Online]. Available: http://arxiv.org/abs/2304.07186

[12] J. A. Hockman, M. E. P. Davies, and I. Fujinaga, "ONE IN THE JUNGLE: DOWNBEAT DETECTION IN HARDCORE, JUNGLE, AND DRUM AND BASS."

[13] M. Rocamora, L. Jure, B. Marenco, M. Fuentes, F. Lanzaro, and A. Gómez, "An audio-visual database of candombe of performances for computational musicological studies," in *Memorias del II Congreso Internacional de Ciencia y Tecnología Musical (CICTeM 2015)*, Buenos Aires, Argentina, Sep. 2015, pp. 17–24.

[14] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. Ellis, "MIR_eval: A Transparent Implementation of Common MIR Metrics," 2014. [Online]. Available: https://www.semanticscholar.org/paper/MIR_EVAL%3A-A-Transparent-Implementation-of-Common-Raffel-McFee/6d37fbd2fccaef3ecdf75d34a7aee18ab9519a6f

[15] "How do we evaluate? 2014; Tempo, Beat and Downbeat Estimation — tempobeatdownbeat.github.io," https://tempobeatdownbeat.github.io/tutorial/ch2_basics/evaluate.html, [Accessed 21-03-2025].

Appendix

A. Code and Data

- Analysis Notebooks, Augmentation and Results
- Beat This! Fork
- Beat This! Annotations Fork

B. Analysis Results

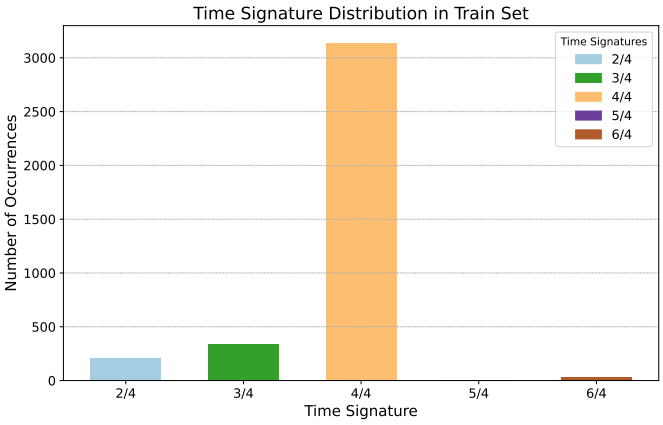

Fig. 1. Time Signature Distribution by Dataset

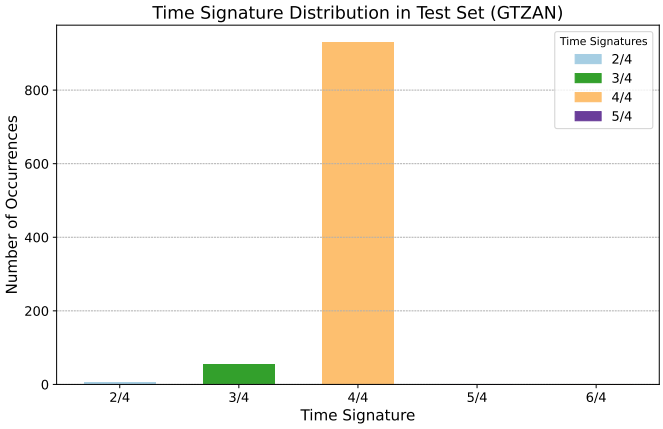

Fig. 2. Time Signature Distribution by Dataset

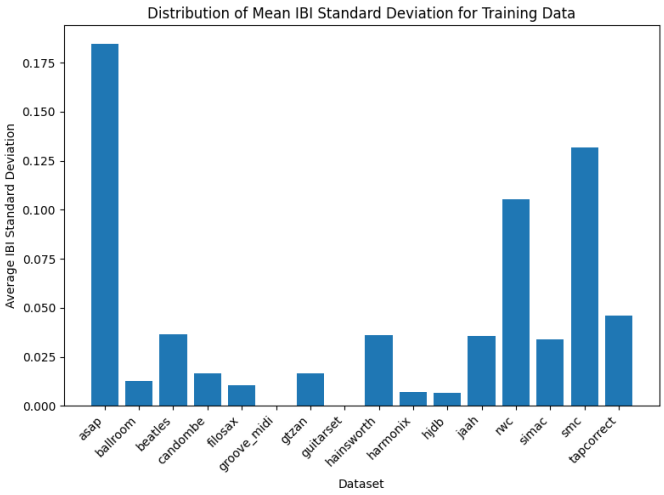

Fig. 3. Distribution of Mean IBI SD by Dataset in Training Data (from Beat This!)

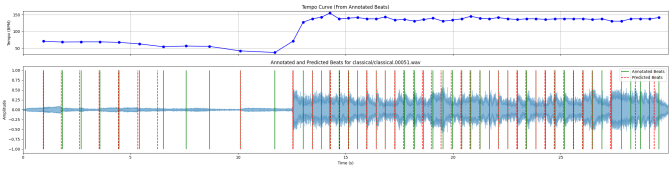

Fig. 4. Plot of Tempo Variation, Beat Annotations, and Predicted Beats with the Pre-trained Model for `gtzan_classical_00051.wav`

C. *Beat This! Final Checkpoint Model Results analysis*

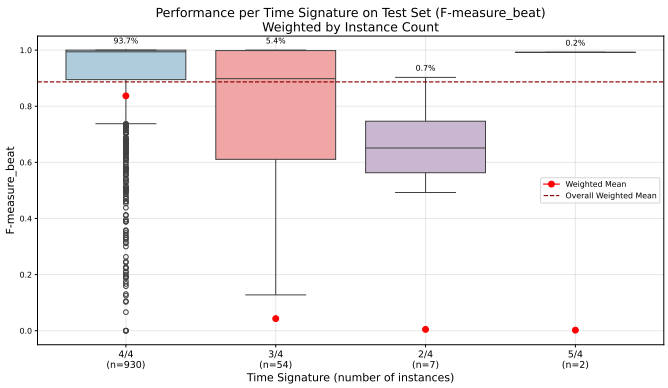

Fig. 5. Weighted F-measure scores on beat detection across time signatures in GTZAN for the *Beat This!* models final checkpoint original model

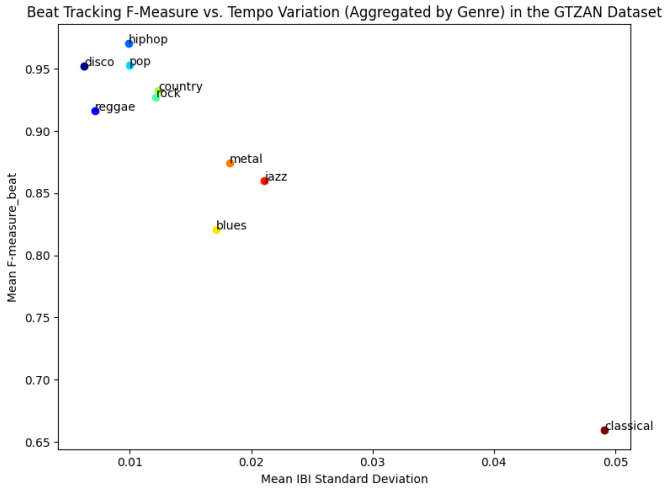

Fig. 6. Mean Test F-Measure vs. IBI SD by Genre in GTZAN (from Beat This!)

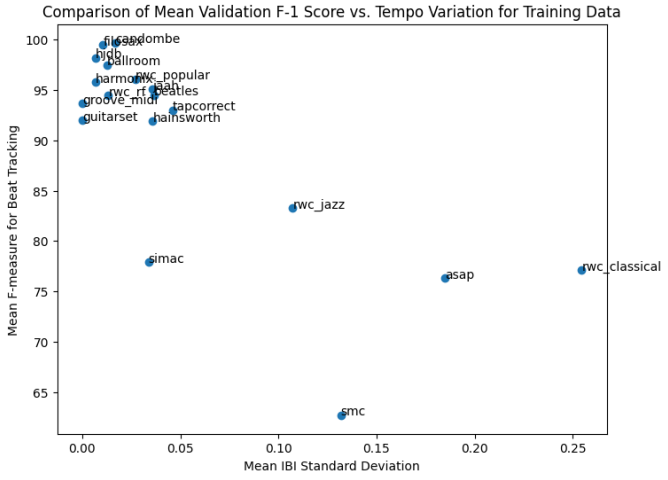

Fig. 7. Mean Validation F-Measure vs. IBI SD by Dataset in Training Data (from Beat This!)

D. Skip-that-beat fine-tuned model results

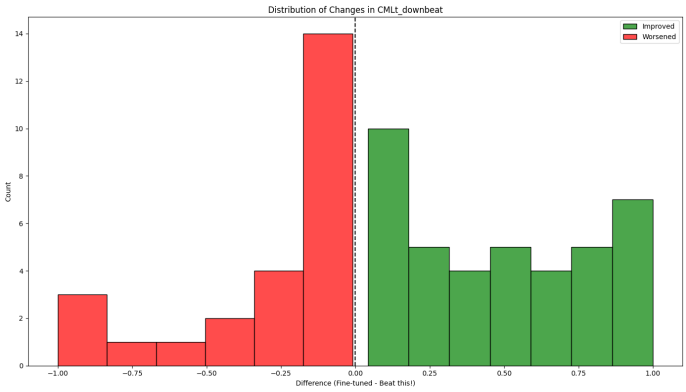

Fig. 8. Change in CMLt downbeat between Candombe time-signature augmentation fine-tuned model and Beat This! on BRID dataset

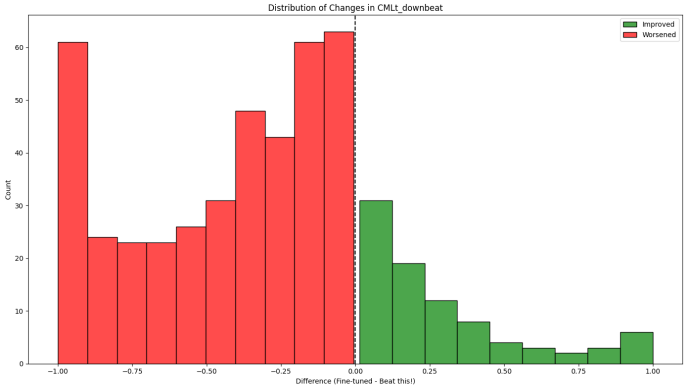

Fig. 9. Change in CMLt downbeat between Candombe time-signature augmentation fine-tuned model and Beat This! on GTZAN dataset

E. Time-stretch fine-tuned model results

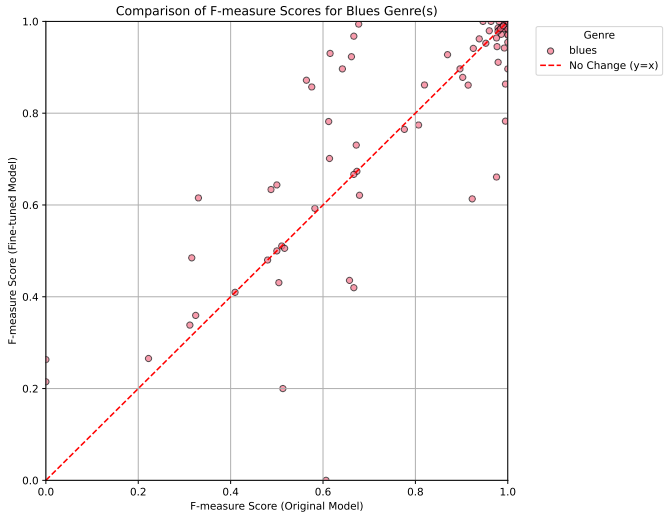

Fig. 10. Comparison of F-measure scores of the original model and the fine tuned model on the time-stretched augmented GuitarSet data

F. Pseudo Code for Time Stretching Augmentation

Algorithm 1 Tempo Variation Augmentation via Time Stretching

Require: Audio a, sample rate sr, beat times $B = \{b_0, b_1, \ldots, b_n\}$, number of augmentations N

Initialize augmented audio $A \leftarrow a$ and beat times $B' \leftarrow B$

for $i = 1$ to N **do**
  Select a random beat index $j \in [0, |B'| - 2]$
  Compute split point $t = \frac{B'_j + B'_{j+1}}{2}$
  Sample a random stretch factor $s \sim \mathcal{U}(0.5, 2.0)$
  Time-stretch audio after t by factor s
  Adjust beat times after t accordingly
end for
return Augmented audio A, updated beat times B'