Transformer-Based Rhythm Quantization of Performance MIDI Using Beat Annotations

Maximilian Wachter
Klangio GmbH
Karlsruhe, Germany

Sebastian Murgul
Klangio GmbH
Karlsruhe, Germany

Michael Heizmann
Institute of Industrial Information Technology
Karlsruhe Institute of Technology
Karlsruhe, Germany

***Abstract*—Rhythm transcription is a key subtask of notation-level Automatic Music Transcription (AMT). While deep learning models have been extensively used for detecting the metrical grid in audio and MIDI performances, beat-based rhythm quantization remains largely unexplored. In this work, we introduce a novel deep learning approach for quantizing MIDI performances using a priori beat information. Our method leverages the transformer architecture to effectively process synchronized score and performance data for training a quantization model. Key components of our approach include dataset preparation, a beat-based pre-quantization method to align performance and score times within a unified framework, and a MIDI tokenizer tailored for this task. We adapt a transformer model based on the T5 architecture to meet the specific requirements of rhythm quantization. The model is evaluated using a set of score-level metrics designed for objective assessment of quantization performance. Through systematic evaluation, we optimize both data representation and model architecture. Additionally, we apply performance and score augmentations, such as transposition, note deletion, and performance-side time jitter, to enhance the model's robustness. Finally, a qualitative analysis compares our model's quantization performance against state-of-the-art probabilistic and deep-learning models on various example pieces. Our model achieves an onset F1-score of 97.3 % and a note value accuracy of 83.3 % on the ASAP dataset. It generalizes well across time signatures, including those not seen during training, and produces readable score output. Fine-tuning on instrument-specific datasets further improves performance by capturing characteristic rhythmic and melodic patterns. This work contributes a robust and flexible framework for beat-based MIDI quantization using transformer models.**

I. Introduction

Automatic Music Transcription (AMT) is one task of specific interest in the field of Music Information Retrieval with the goal of retrieving a symbolic music representation from an input audio signal [1]. In most research available the representation retrieved is in a MIDI format with unquantized note onset and offset times. In order to retrieve a notation-level transcription that can be displayed as sheet music, the note timing has to be quantized to an underlying beat grid. This beat grid can be derived from the result of a beat tracking approach or adopted directly in case a piece is performed to a metronome. Since existing probabilistic and deep-learning-based quantization models infer beat information from transcribed rhythms, they are not capable of leveraging metronomic data, even if it is available. Our model on the other hand utilizes this beat grid as a priori information. Furthermore, while it would be possible to directly calculate the quantized note durations from the beat grid, the resulting durations would still reflect all human performance inaccuracies. Therefore, the result would only be barely readable and not practical in most applications.

In this work, we propose a new transformer-based approach that is able to accurately quantize performance MIDIs to scores based on a priori beat information. We show that this approach offers flexibility as well as control over the outputs by simply encoding time signature changes in the beat counter. By fine-tuning the provided model on instrument-specific datasets we are capable of further optimizing the results by modeling common rhythmic and melodic patterns that are used for each instrument.

II. Related Work

While beat detection has been extensively studied, the field of rhythm quantization remains comparatively underexplored. This section distinguishes between research efforts focused on beat tracking and those addressing rhythm quantization. Beat tracking aims to detect the temporal positions of beats within a musical performance. In 2011, Böck et al. introduced a frame-level beat classification method using bidirectional Recurrent Neural Networks (RNNs) and autocorrelation smoothing [2]. This method was later extended with a Dynamic Bayesian Network (DBN) for modeling meter and bar structure [3]. In 2019, Davies et al. improved this by replacing RNNs with a dilated Temporal Convolutional Network (TCN) [4]. Zhao et al. proposed Beat Transformer in 2022, incorporating attention mechanisms and demixed spectrograms [5]. Foscarin et al. introduced Beat This! in 2024, a transformer model robust to style and tempo changes, eliminating the need for DBN postprocessing [6]. Most recently, in 2025, Murgul et al. reframed beat tracking in performance MIDI as a transformer-based sequence translation task [7].

Rhythm quantization involves aligning performed note onsets to a metrical grid to obtain a symbolic music representation. Early methods relied on rule-based and probabilistic techniques, while more recent approaches leverage machine learning models. Cambouropoulos et al. proposed a system for joint beat detection and rhythm quantization in 2000 [8]. Their approach clustered inter-onset

intervals for beat detection, followed by assigning note onsets to the closest points on a metrical grid and assigning note values based on inter-onset intervals. Cemgil et al. introduced a quantization framework in the same year, utilizing Bayesian probabilistic modeling, incorporating a performance model to formalize simple quantization strategies alongside a prior model to account for rhythmic complexity [9]. In 2002, Takeda et al. proposed the first method utilizing Hidden Markov Models (HMMs) for the rhythm transcription task [10]. They employed the Viterbi algorithm to estimate note values by combining a stochastic model of timing deviations and a grammatical model of plausible note sequences. Hamanaka et al. proposed a method in 2003 for estimating intended onset times from fixed-tempo jam sessions by training an HMM with human performance data using the Baum-Welch algorithm [11]. Temperley's 2007 book 'Music and Probability' extended Bayesian probabilistic approaches to infer complete metrical grids rather than score positions relative to a bar [12]. Cogliati et al. presented an HMM-based system in 2016 for joint estimation of meter, harmony, and stream separation, combined with a distance-based quantization algorithm [13]. Foscarin et al. introduced a parse-based system in 2019 employing weighted context-free grammars (WCFGs) for joint rhythm quantization and music score production [14]. Shibata et al. proposed a piano transcription system in 2021 that incorporated HMMs and Markov Random Fields (MRFs) for rhythm quantization, leveraging non-local musical statistics to infer global parameters [15]. Liu et al. proposed a Convolutional-Recurrent Neural Network (CRNN)-based system in 2022 for MIDI-to-score conversion, incorporating onset-based beat detection and rhythm quantization [16]. Kim et al. developed a transformer and Convolutional Neural Network (CNN)-based guitar transcription model in 2023 that produced note-level transcriptions from spectrograms using beat information [17]. Beyer et al. introduced a performance MIDI-to-score conversion approach in 2024 based on the Roformer architecture. Their encoder-decoder model directly generated MusicXML tokens while implicitly performing beat estimation and rhythm quantization on MIDI token sequences [18].

Although recent transformer-based models have advanced MIDI-to-score conversion, most rely on end-to-end architectures with implicit beat estimation. This approach prevents the use of external beat information, such as metronome data or manually annotated beats. Incorporating explicit beat inputs into quantization systems improves both flexibility and interpretability, filling an important gap in current research.

III. Methodology

A. Task Definition

Our model aims to convert an unquantized performance MIDI sequence X_n, represented in the time domain, into a score-like sequence Y_n with musical timing information, guided by beat and downbeat annotations X_{b} and X_{db}. Unlike most state-of-the-art quantization models, such as [18], which operate solely on time-domain input, our approach explicitly incorporates a priori beat information, including beat estimations or metronomic data. In particular, when a performance is aligned to a metronome, this information can remove ambiguity about the underlying metrical grid, leading to more accurate quantization. The model assumes a one-to-one correspondence between performance and score notes, ensuring that no additional notes are added or removed during quantization. This criterion further ensures correspondence between measures in scores and performances, which is necessary due to the at times poor alignment between them.

The input sequence X_n is defined as

$$X_n = \{(p_i, o_i, d_i)\}_{i=1}^{N_{\text{perf}}} \tag{1}$$

with the individual notes in the sequence being represented by the MIDI pitch p_i, onset o_i, and duration d_i in seconds. Notes in the target sequence Y_n are represented using musical onsets mo_i and musical note values mnv_i described by

$$Y_n = \{(p_i, mo_i, mnv_i, n_{\text{measure}})\}_{i=1}^{N_{\text{perf}}} \quad . \tag{2}$$

To limit the range of possible values, musical onsets hereby denote a note's position within its respective measure instead of the absolute position within the entire piece. Thus, a note's position within a score is defined by its measure number n_{measure} and its musical onset time.

Beat annotations X_{b} and downbeat annotations X_{db} are given in the form of

$$X_{\text{b}} = \{(t_j)\}_{j=1}^{N_{\text{beat}}} \tag{3}$$

and

$$X_{\text{db}} = \{(t_k)\}_{k=1}^{N_{\text{downbeat}}} \tag{4}$$

where $X_{\text{db}} \subseteq X_{\text{b}}$.

While it is logical to train the model using ground truth beat data, it is reasonable to assume that, once a model has gained a deep understanding of rhythmic structure, it is able to produce a meaningful quantization even with faulty beat data. To obtain a tokenizable representation of X_n, the MIDI note time information is fused with X_{b} and X_{db}. Therefore, we interpolate X_{b} to twelve equidistant sub-beats (referred to as ticks $tick_l$) per beat, obtaining a 32$^{\text{nd}}$-note triplet grid X_{ticks} given as

$$X_{ticks} = \{(tick_l)\}_{l=1}^{12 \cdot N_{\text{beat}}}. \tag{5}$$

This resolution is chosen as it allows representation of straight and triplet-based note values of a 16$^{\text{th}}$-note triplet and above. The continuous onset and duration times are then quantized to 32$^{\text{nd}}$-note triplets using the Euclidean distance given by

$$\arg\min_{l} \left(|tick_l - o_i|\right) \quad . \tag{6}$$

To enable independent training on individual measures, onset values are reset to zero at the beginning of each measure. While this step already converts performance note times into

TABLE I
SUMMARY OF TOKEN TYPES USED IN THE MODEL'S VOCABULARY, INCLUDING PITCH, ONSET, NOTE VALUE, AND STRUCTURAL INDICATORS, WITH CORRESPONDING VALUE RANGES AND TOTAL COUNTS.

Token Type	No. of Values	Value Range
Pitch	88	$[21, 108]$
Onset	48	$[0, 47]$
Note value	48	$[1, 48]$
New measure	1	$\{0\}$

musical note values, the results are not suitable for human-readable scores due to timing inconsistencies in expressive performances, which often produce irregular or complex note durations. The score sequences are based on the MusicXML format [19], where onset and note values are expressed in quarter note units. To align with this representation, onsets and durations are scaled by a factor of 12. With both X_n and Y_n expressed in musical note values, input and target sequences can be encoded using a unified tokenization scheme.

B. Tokenization Scheme

The tokenization scheme is designed to efficiently encode the note sequences derived from both performance and score data. It is loosely based on the approach introduced in [20]. In our model, each note in the input sequence X_n and the target sequence Y_n is represented using three distinct tokens: one each for pitch, onset, and note value. Measure numbers are not encoded explicitly. Instead, the start of a new measure is indicated by a dedicated 'new measure' token, eliminating the need for a wide range of measure number tokens. For input sequences, a new measure token is inserted when a note's onset exceeds the downbeat of the following measure. In the target sequences, new measure tokens are inferred directly from the score. The vocabulary consists of 88 pitch tokens (covering the full range of piano MIDI pitches), 48 onset tokens (corresponding to 32^{nd}-note triplet subdivisions in a 4/4 measure), and 48 note value tokens. As a result, the current model supports quantization of measures up to the length of a whole note, with note values capped at that duration. The full set of token types and their ranges is detailed in Table I.

C. Network Architecture

Our model architecture is based on the T5 transformer by Raffel et al. [21], but we omit pre-training since we use a custom token vocabulary described in Section III-B. The configuration is significantly smaller than *t5-small*. Specifically, the model uses a key/value dimensionality of $d_{\text{kv}} = 64$, a feed-forward layer size of $d_{\text{ff}} = 1024$, and a vocabulary size of 187, which includes the tokens from Section III-B along with end-of-sequence (EOS) and padding (PAD) tokens. During optimization, we reduced the number of layers from six to two, the number of attention heads from eight to four, and the embedding size from 512 to 128. This configuration yielded the best empirical results. The improved performance of the smaller model can be attributed to the structured nature of onset and note value data, where closely related values frequently co-occur, and to the compact vocabulary size, which renders larger embedding spaces unnecessary. As a result, our model is more computationally efficient than other state-of-the-art approaches. Furthermore, by processing short segments of M measures sequentially, computational cost grows linearly with input length rather than exponentially, enabling scalable and parallelizable inference.

D. Training and Inference

The T5 model is trained using cross-entropy loss and the Adafactor optimizer [22]. Although prior work such as [20] recommends a fixed learning rate of 0.001, we adopt an adaptive learning rate, which led to faster and more stable convergence in our experiments. Each input sequence consists of M measures, where M is a tunable parameter (see Section IV-D). The sequences are non-overlapping, as empirical results showed improved performance with shorter inputs, indicating limited long-range context dependency. Notes are tokenized into a single one-dimensional sequence, with each note represented by an ordered triplet of pitch, onset, and note value tokens. This requires the model to learn the correct token structure to generate processable outputs. Training was performed for up to 100 epochs with a batch size of eight. An early stopping criterion was applied, terminating training if validation loss did not improve for 20 consecutive epochs. In practice, convergence typically occurred before epoch 60. We used a dropout rate of 0.1 for regularization during training. During inference, we apply beam search decoding with a beam width of five.

IV. EXPERIMENTS

A. Datasets

Effective rhythm transcription requires learning the relationship between expressive timing variations in human performances and the corresponding notated musical timing. Consequently, a suitable dataset must include ground truth scores, human-performed MIDI recordings, and precise beat and downbeat annotations. These criteria exclude the widely used A-MAPS dataset [23], as its performances are generated from tempo-modified quantized MIDI files and do not reflect the full spectrum of human timing deviations [24]. This leaves only a small selection of datasets.

Firstly, we use the ASAP dataset [24], which, to the best of our knowledge, most effectively meets these requirements. It contains $1,067$ performance MIDI files spanning 236 classical piano pieces, with many pieces having multiple performances. Each performance is paired with a MusicXML score and includes annotations for beats, downbeats, time signatures, and musical keys.

B. Dataset Preparation

Figure 1 illustrates the data preparation pipeline for our model. We train on measures in the time signatures 4/4, 3/4, and 2/4, which are separated into distinct datasets to facilitate comparison. Other time signatures are omitted. Aside from this separation, the model is time signature-agnostic, as the relevant time signature is provided as a priori information and

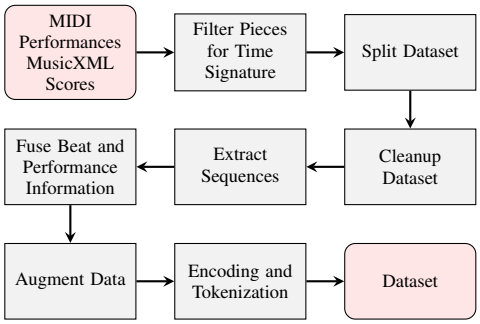

Fig. 1. Overview of the data preprocessing steps, from raw MIDI and MusicXML files to finalized token sequences. Including filtering, alignment, augmentation, and tokenization stages for training-ready inputs.

applied during preprocessing. This results in 85 pieces with 4/4 measures, 53 with 3/4, and 50 with 2/4.

We randomly select 10 % of the pieces for the test and validation sets, using the remaining 90 % for training. Since ties in MusicXML are represented as two distinct notes, we resolve them by matching each tie end to its corresponding tie start, merging their durations and removing the redundant note. Training examples are extracted as sequences of M measures. To simplify input, hand separation is removed so that each sequence consists of a single stream of individual notes. Measures whose actual duration does not match the annotated time signature are excluded. As our model assumes a one-to-one correspondence between performance and score notes, we include only those measures where the number of notes in both sequences aligns. To increase the number of eligible measures, we shift the search interval for note matching 50 ms earlier than the annotated downbeats. This accounts for notes played slightly ahead of the measure boundary, which typically belong to the following measure. The 50 ms threshold was chosen empirically to maximize training coverage. After filtering and alignment, approximately $40,000$ measures remain available for training.

We similarly adapt the model to guitar data using the Leduc dataset [25], which contains 239 jazz guitar performances with high-quality transcriptions by François Leduc in GuitarPro[1] format. These scores are converted to MIDI and aligned with the original audio using the method described by Riley et al. [26] which we extended to also align beats and downbeats. We then apply the same preprocessing pipeline used for the piano data, treating the aligned MIDI transcriptions as performance input.

C. Metrics

We use two sets of metrics for evaluation: one for model optimization and one for comparison with other methods. Since the model is trained under the assumption of a one-to-one correspondence between performance and score notes, we ensure that the length of the generated sequence matches the unquantized input at inference time.

[1]https://www.guitar-pro.com

For optimization, we use onset F1-score, note value accuracy, and note value mean squared error (MSE). A true positive in onset F1 is defined as a note with exactly matching pitch and onset in both the quantized output and the ground truth. This metric relies on ground truth beat annotations, as any deviation would misalign onsets; therefore, it cannot be used to compare models that do not leverage beat information and are therefore not alignable. Note value metrics are computed only for notes with correctly predicted onsets, as correct note values at incorrect positions are not musically meaningful. This introduces a bias: errors in onset prediction often imply errors in note duration. Note value MSE, expressed in quarter note units, is averaged over all test examples. As note value accuracy is strongly tied to onset precision, it serves as the primary metric for model tuning.

For broader evaluation and comparison, we adopt the edit-distance-based metrics introduced in [27], commonly referred to as MUSTER scores [18]. The onset-time error rate ϵ_{onset} is calculated based on the number of shift and scaling operations needed to align the estimated score with the ground truth [28]. The offset-time error rate ϵ_{offset} reflects mismatches in note offsets relative to their onsets. To account for additional, missing, or incorrect notes, the output and reference sequences are first aligned, also yielding a pitch error rate ϵ_p, an extra note rate ϵ_{extra}, and a missing note rate ϵ_{miss}. However, since our model assumes a one-to-one correspondence between input and output notes, these three metrics are not applicable and are therefore excluded from evaluation.

D. Model Optimization

Our optimization strategy begins with training a baseline model using only 4/4 measures from the ASAP dataset, focusing on tuning model parameters and evaluating data augmentation methods. Once optimized, the model is extended to handle additional time signatures and, finally, adapted to guitar performances from the Leduc dataset.

1) Optimizing Training Sequences: We begin by optimizing the number of measures per input sequence (M) and the ordering of notes within each sequence. As shown in Table II, using two-measure sequences yields slightly better performance than other configurations. Apparently, the two-measure version provides sequences that are short enough to learn meaningful representations in the intermediate layers but still provides enough contextual information between the measures.

Further improvements are achieved by synchronizing the note order between input and target sequences, leveraging the one-to-one correspondence assumption. Specifically, we reorder the target sequence to match the onset-sorted order of the input sequence, ensuring a more consistent mapping between corresponding notes. As Table II indicates, this synchronization results in a notable increase of approximately 2 % in onset F1-score compared to unsynchronized sequences.

2) Data Augmentation: To further enhance model performance, we apply three data augmentation strategies during training:

TABLE II
EVALUATION RESULTS ON THE ASAP DATASET FOR MODELS TRAINED WITH ONE TO FOUR MEASURES PER INPUT SEQUENCE, WITH AND WITHOUT SYNCHRONIZED NOTE ORDERING. METRICS INCLUDE ONSET F1-SCORE, NOTE VALUE ACCURACY (NV ACC.), AND NOTE VALUE MEAN SQUARED ERROR (NV MSE).

Number of Measures	Synchronized	Onset F1	NV Acc.	NV MSE
One	No	93.4 %	80.9 %	0.25
One	Yes	95.8 %	81.8 %	**0.21**
Two	No	94.0 %	82.5 %	0.23
Two	Yes	**96.0 %**	**82.6 %**	**0.21**
Three	Yes	95.9 %	80.4 %	0.27
Four	Yes	94.8 %	81.2 %	0.22

- **Transposition:** input and target sequences are randomly transposed by a fixed number of semitones. The transpose value is drawn from a uniform distribution so that the available MIDI pitch range for piano is not exceeded.
- **Deletion:** During training 20 % of notes are randomly selected and deleted from the input and label sequences with a 50 % probability.
- **Note Value Noise:** Since the deviation between performance and score is quite high for note durations, we add a noise term to performance note durations, following a normal distribution with a standard deviation of 5 % of the note duration.

The effects of these augmentations on onset F1-score and note value accuracy are shown in Figure 2.

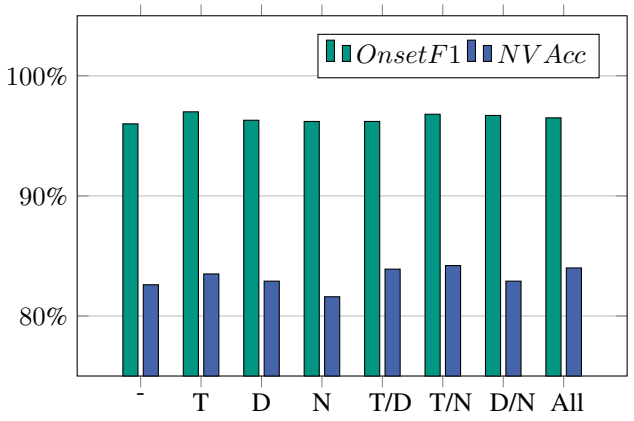

Fig. 2. Comparison of Onset F1-score and Note Value Accuracy in percent for various combinations of augmentation methods: Transposition (T), Deletion (D), and Note Value Noise (N). Combining methods yields the best overall performance.

Notably, all augmentation methods combined improve the onset F1-score. However, applying note value noise alone reduces note value accuracy. When combined with other augmentations, especially transposition, it yields substantial improvements. This may be due to delayed early stopping, allowing the model more time to generalize from the augmented data. Interestingly, transposition improves quantization performance even though it does not modify rhythm. This suggests it strengthens the model's understanding of pitch-related structure and exposes it to underrepresented pitch ranges. Based on these findings, subsequent models are trained using a combination of transposition and note value noise, as this yields the best balance between onset and note value accuracy.

TABLE III
QUANTIZATION RESULTS ON THE ASAP DATASET FOR MODELS TRAINED ON VARIOUS COMBINATIONS OF 2/4, 3/4, AND 4/4 TIME SIGNATURES AND TESTED ON EACH TIME SIGNATURE INDIVIDUALLY. INCLUDING MULTIPLE TIME SIGNATURES GENERALLY IMPROVES PERFORMANCE AND GENERALIZATION.

Tested On	Trained On	Onset F1	NV Acc.	NV MSE
4/4	4/4	96.8 %	84.2 %	0.19
	3/4, 4/4	**96.9 %**	**85.2 %**	**0.17**
	2/4, 3/4, 4/4	96.8 %	84.0 %	0.20
3/4	3/4	97.4 %	76.8 %	0.47
	4/4	95.9 %	68.5 %	0.54
	3/4, 4/4	97.9 %	**80.4 %**	0.41
	2/4, 3/4, 4/4	**98.1 %**	79.0 %	**0.40**
2/4	2/4	96.0 %	80.8 %	0.12
	4/4	96.7 %	84.2 %	0.13
	3/4, 4/4	97.0 %	84.5 %	0.13
	2/4, 3/4, 4/4	**98.0 %**	**86.9 %**	**0.10**

3) Extending to Different Time Signatures: Since our model implicitly handles time signatures, no explicit time signature token is required. Instead, support for different time signatures is achieved through the preprocessing procedure described in Section III-B, where note timing is given relative to the measure start but not tied to a specific time signature. As a result, the model can generalize to time signatures it was not explicitly trained on. Table III presents evaluation results for models trained on various combinations of time signatures. Each model is tested on a separate set containing a single time signature to assess how including data from other meters influences performance.

Incorporating multiple time signatures, and therefore a more diverse dataset, generally improves quantization performance. This effect is likely due to increased data variety, which enhances the model's ability to generalize. Rhythmic patterns learned from one time signature often transfer successfully to others, as shown by the strong performance of the 4/4-trained model on both 2/4 and 3/4 sequences. These findings support the use of a single model across all time signatures rather than training separate models for each case. An exception, as shown in Table III, is a slight decrease in note value accuracy for 4/4 when 2/4 data is included in training. This may be explained by the reduced occurrence of longer note values in shorter measures. Nonetheless, the overall advantages of a combined model, including improved generalization and simplified deployment, make it the preferred approach.

4) Training on Guitar Data: Finally, we extend the model to guitar data using the Leduc dataset [25]. To evaluate instrument-specific generalization, we train three versions of the model: one using only guitar data, one using only piano data, and one trained on a combined dataset. Each model is tested on both guitar and piano data to assess whether rhythmic performance characteristics differ by instrument and whether cross-instrument generalization is feasible. All models

are trained on measures in 2/4, 3/4, and 4/4 time signatures, consistent with the findings in Section IV-D3. The evaluation results are presented in Figure IV.

TABLE IV
ONSET F1-SCORE AND NOTE VALUE ACCURACY FOR MODELS TRAINED ON GUITAR (LEDUC), PIANO (ASAP), AND COMBINED DATASETS, EVALUATED ON BOTH TEST SETS. INSTRUMENT-SPECIFIC MODELS OUTPERFORM CROSS-DOMAIN MODELS, ESPECIALLY IN NOTE VALUE ACCURACY.

Tested on	Trained on	Onset F1	Note value acc.
Leduc	Leduc	92.1 %	90.2 %
	ASAP	87.2 %	71.3 %
	Leduc + ASAP	90.3 %	86.9 %
ASAP	Leduc	90.5 %	69.4 %
	ASAP	97.3 %	83.3 %
	Leduc + ASAP	97.2 %	81.1 %

Although our model operates solely on symbolic data, the results clearly indicate that models trained exclusively on data from a specific instrument outperform those trained on other instruments. While the combined model reliably yields improvements compared to the evaluation results for models trained without the evaluated instrument, it still underperforms compared to the evaluation results of the models trained exclusively on the evaluated instrument. Differences in the quantization performance become especially apparent for note value accuracy, which corresponds with the assumption that the characteristics of rhythmic interpretation of note durations vary from instrument to instrument. Consequently, training separate models for each instrument, when feasible, appears to be more effective than relying on a single, universal model.

E. Comparative Experiments

For comparative evaluation, we train our model using the ASAP splits defined in the ACPAS dataset [31], which combines predefined train/test splits from A-MAPS [23] and ASAP [24]. This setup ensures direct comparability with prior work, including [16] and [18]. We use the MUSTER score [28] as our primary comparison metric by using the publicly available implementation found on Github[2]. Table V shows a comparison of our evaluation results with the results achieved by the deep learning-based model by Beyer et al. [18] as well as other models referenced in the same publication. These include the commercial products MuseScore [29] and Finale [30] as well as state-of-the-art probabilistic and deep learning-based approaches [15], [16].

[2]https://github.com/amtevaluation/amtevaluation.github.io

TABLE V
COMPARISON OF ONSET-TIME (ϵ_{onset}) AND OFFSET-TIME (ϵ_{offset}) ERROR RATES USING THE MUSTER METRIC. THE PROPOSED MODEL OUTPERFORMS ALL BASELINES IN ONSET QUANTIZATION AND RANKS SECOND IN OFFSET ACCURACY.

Method	ϵ_{onset}	ϵ_{offset}
Neural Beat Tracking [16]	68.28	54.11
End-to-End PM2S [18]	15.55	**23.84**
HMMs + Heuristics (J-Pop) [15]	25.02	29.21
HMMs + Heuristics (classical) [15]	22.58	29.84
MuseScore [29]	47.90	49.44
Finale [30]	31.85	45.34
Our Model	**12.30**	28.30

In order to compute the metric, we quantize entire performances by segmenting the performances in the test set into sequences of two measures, quantizing the sequences, and concatenating them back to a quantized MusicXML score. The comparative experiments pose an additional challenge, since the test set partly consists of time signatures that our model was not trained on, like 6/8 or even 12/16. We transcribe these by adapting the preprocessing for these time signatures to the beat count used in the annotations. For instance, since 6/8 measures are counted in two, we interpolate to 18 ticks per beat in this case. To produce valid and readable scores, we apply post-processing steps including chord merging, tie reconstruction for notes crossing measure boundaries, and a simple voice separation algorithm to avoid overlapping notes within a voice.

Our model achieves the best performance in terms of ϵ_{onset} and is second only to [18] in ϵ_{offset}. Notably, our model is not trained to recognize 32$^{\text{nd}}$ notes or irregular time signatures, which are present in the test set. Despite this, the results demonstrate that leveraging beat annotations enables our model to match or surpass state-of-the-art quantization approaches.

V. CONCLUSION

In this work, we presented the first transformer-based model for beat-based rhythm quantization of MIDI performances. To enable this, we introduced a simple yet effective preprocessing method that fuses beat and performance information into a unified tokenized representation for both input and target sequences. We adapted this preprocessing method to different time signatures and proved that the resulting model is capable of quantizing unseen time signatures once adapted to the preprocessing framework. We defined a confusion-based metric for evaluating beat-based quantization derived from musical onset and note value and optimized our model using different sequence structures and augmentations. While initially training on piano performances from the ASAP dataset, we adapted the model to the domain of guitar data using the Leduc dataset and showed that instrument-specific rhythm quantization models show better performance which is likely due to the differences in rhythmic interpretation between instruments.

Future work may focus on expanding the model's capabilities by incorporating a broader range of time signatures and note values, extending the dataset, and integrating additional musical context such as voice separation by including voice tokens to individual notes or explicit time signature tokens.

REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic Music Transcription: An Overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2018.

[2] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx)*, 2011.

[3] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[4] M. E. P. Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *27th European Signal Processing Conference (EUSIPCO)*, 2019.

[5] J. Zhao, G. Xia, and Y. Wang, "Beat transformer: Demixed beat and downbeat tracking with dilated self-attention," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[6] F. Foscarin, J. Schlüter, and G. Widmer, "Beat this! accurate beat tracking without dbn postprocessing," in *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[7] S. Murgul and M. Heizmann, "Beat and downbeat tracking in performance midi using an end-to-end transformer architecture," in *Proceedings of the 22nd Sound and Music Computing Conference (SMC)*, 2025.

[8] E. Cambouropoulos, "From midi to traditional musical notation," in *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis*, 2000.

[9] A. T. Cemgil, P. Desain, and B. Kappen, "Rhythm quantization for transcription," *Computer Music Journal*, vol. 24, no. 2, pp. 60–76, 2000.

[10] H. Takeda, N. Saito, T. Otsuki, M. Nakai, H. Shimodaira, and S. Sagayama, "Hidden markov model for automatic transcription of midi signals," in *IEEE Workshop on Multimedia Signal Processing*, 2002.

[11] M. Hamanaka, M. Goto, H. Asoh, and N. Otsu, "A learning-based quantization: Unsupervised estimation of the model parameters," in *Proceedings of the International Computer Music Conference (ICMC)*, 2003.

[12] D. Temperley, *Music and probability*. Mit Press, 2007.

[13] A. Cogliati, D. Temperley, and Z. Duan, "Transcribing human piano performances into music notation." in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[14] F. Foscarin, F. Jacquemard, P. Rigaux, and M. Sakai, "A parse-based framework for coupled rhythm quantization and score structuring," in *Mathematics and Computation in Music*, 2019.

[15] K. Shibata, E. Nakamura, and K. Yoshii, "Non-local musical statistics as guides for audio-to-score piano transcription," *Information Sciences*, vol. 566, pp. 262–280, 2021.

[16] L. Liu, Q. Kong, G. Morfi, E. Benetos *et al.*, "Performance midi-to-score conversion by neural beat tracking," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[17] S. Kim, T. Hayashi, and T. Toda, "Note-level automatic guitar transcription using attention mechanism," in *Proceedings of the 30th European Signal Processing Conference (EUSIPCO)*, 2022.

[18] T. Beyer and A. Dai, "End-to-end piano performance-midi to score conversion with transformers," in *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[19] MakeMusic, Inc. (2025) Musicxml for exchanging digital sheet music. https://www.musicxml.com/. (accessed Feb. 19, 2025).

[20] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[21] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[22] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[23] A. Ycart and E. Benetos, "A-maps: Augmented maps dataset with rhythm and key annotations," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[24] F. Foscarin, A. Mcleod, P. Rigaux, F. Jacquemard, and M. Sakai, "Asap: a dataset of aligned scores and performances for piano transcription," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[25] D. Edwards, X. Riley, and S. Dixon, "The Francois Leduc Dataset," Apr. 2024. [Online]. Available: https://doi.org/10.5281/zenodo.10984521

[26] X. Riley, D. Edwards, and S. Dixon, "High resolution guitar transcription via domain adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1051–1055.

[27] E. Nakamura, E. Benetos, K. Yoshii, and S. Dixon, "Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[28] E. Nakamura, K. Yoshii, and S. Sagayama, "Rhythm transcription of polyphonic piano music based on merged-output hmm for multiple voices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[29] MuseScore, "MuseScore: Free music composition and notation software," https://musescore.org, 2025, (accessed Feb. 20, 2025).

[30] MakeMusic, Inc., "Finale version 27," https://finalemusic.com/, 1988, (accessed Jul. 17, 2024).

[31] L. Liu, V. Morfi, E. Benetos *et al.*, "Acpas: a dataset of aligned classical piano audio and scores for audio-to-score transcription," in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.