# HIGH-RESOLUTION VIOLIN TRANSCRIPTION USING WEAK LABELS

**Nazif Can Tamer**[♭] **Yigitcan Özer**[♯] **Meinard Müller**[♯] **Xavier Serra**[♭]
[♭] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[♯] International Audio Laboratories Erlangen, Germany
nazifcan.tamer@upf.edu, yigitcan.oezer@audiolabs-erlangen.de,
meinard.mueller@audiolabs-erlangen.de, xavier.serra@upf.edu

## ABSTRACT

A descriptive transcription of a violin performance requires detecting not only the notes but also the fine-grained pitch variations, such as vibrato. Most existing deep learning methods for music transcription do not capture these variations and often need frame-level annotations, which are scarce for the violin. In this paper, we propose a novel method for high-resolution violin transcription that can leverage piece-level weak labels for training. Our conformer-based model works on the raw audio waveform and transcribes violin notes and their corresponding pitch deviations with 5.8 ms frame resolution and 10-cent frequency resolution. We demonstrate that our method (1) outperforms generic systems in the proxy tasks of violin transcription and pitch estimation, and (2) can automatically generate new training labels by aligning its feature representations with unseen scores. We share our model along with 34 hours of score-aligned solo violin performance dataset, notably including the 24 Paganini Caprices.

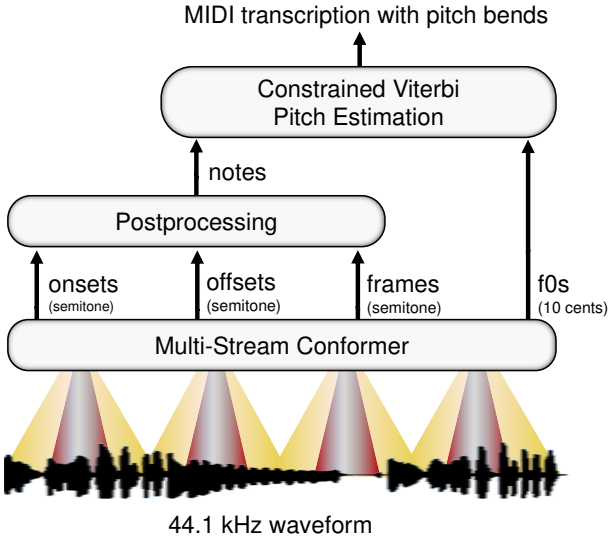


**Figure 1**: Our method transcribes violin recordings sampled with 44.1 kHz waveform into MIDI with a 5.8 ms time- and 10-cent frequency-resolution pitch bends.

## 1. INTRODUCTION

Automatic music transcription (AMT) is a core task in Music Information Retrieval that aims to convert a musical performance into some form of symbolic notation. While general-purpose AMT systems have recently seen substantial progress with deep learning [1–5], instrument-specific systems usually perform better, e.g., for piano [6–9], vocals [10, 11], guitar [12–14], and drums [15–17]. Despite the prominence of the violin in Western classical music and other traditions, a specialized high-precision violin transcription system that applies the recent advances in deep learning does not exist. In this paper, we aim to transcribe violin performances into a descriptive music notation [18]. As opposed to a prescriptive transcription, whose aim would be to produce an easily understandable score from which a musician can perform according to stylistic conventions of Western classical music writing, a descriptive transcription has an analytical purpose, aiming at notating high-precision pitch modulations along the notes.

Most typical AMT systems employ audio-to-MIDI transcription where each note event is represented with semitone resolution in the 12-tone equal temperament (12-TET). However, cognitive studies show that even the Western classical violinists heavily deviate from the 12-TET in favor of Pythagorean tuning and just intonation [19, 20]. Furthermore, the violin also plays a central role in many other traditions that do not employ the Western 12-TET [21]. Considering playing styles such as the vibrato and glissando that involve pitch modulations, a higher frequency resolution than the conventional 12-TET is required for violin transcription. An important step towards transcription outside the 12-TET was introduced by Bittner et al. [1] with an instrument-agnostic AMT system, which employs MIDI pitch bends to represent performances with 33-cent frequency resolution. However, adapting their approach to violin transcription remains to be a challenge since 33-cent frequency resolution is still too high compared to a violinist's intonation precision [20].

A further main challenge in violin transcription is the lack of frame-level annotated training data. To cope with the absence of frame-level annotations, Weiß and

Peeters [22] employ sequence-level targets and a variant of the Connectionist Temporal Classification (CTC) loss for multipitch estimation. However, this strategy is sensitive to the segment duration (stable until segment lengths of 60 seconds) and, therefore, still requires some form of weak alignment. While some works explore data augmentation for frame-level supervised models through additional unlabeled [4] or pseudo-labeled [5] data, recent AMT methods are mostly trained using frame-level annotations [1, 3, 8, 9]. In some cases, obtaining such annotations is feasible through electronic music instruments, e.g., Disklavier. For example, the MAESTRO [23] dataset, with 200 hours of virtuoso piano performances and respective note labels captured with 3 ms frame resolution, enabled significant improvements for piano transcription.

In case electronic music instruments are unavailable, a common approach for obtaining automatic frame-level annotations is employing audio-to-score alignment (ASA), which found application in score following [24, 25]. ASA itself is not a technology developed for creating training datasets for AMT systems, and it has been reported that inaccurately aligned datasets may even worsen the result [2]. The intertwined nature of ASA and transcription can also be viewed from another aspect. For example, Kwon et al. [26] showed that frame and onset features of an AMT system work as robust feature representations for ASA. To our knowledge, the only deep-learning-based transcription system that integrates ASA into AMT is the recent work by Maman and Bermano [2], which utilizes ASA with chroma representations obtained from AMT frames.

As the main contribution of this paper, we propose a novel AMT system specifically tailored for descriptive violin transcription [1] regarding two crucial aspects: 1) We represent pitch deviations such as vibrato, glissando, or intonation choice by incorporating fine-grained pitch representations into the transcription. While borrowing our note postprocessing system and the MIDI pitch bend representations from Bittner et al. [1], we build a conformer-based model that works on the raw audio waveform and further improves the pitch bend estimation through note-constrained Viterbi pitch tracking. 2) We acquire frame-level annotations for violin transcription by considering simultaneous transcription and alignment in a joint framework, similar to the work by Maman and Bermano [2]. Following the findings from the music synchronization literature, we also incorporate activation-function-based features in the alignment [27, 28].

In order to benchmark our descriptive violin transcription method, we consider the proxy tasks of transcription and pitch estimation and compare our model with general-purpose baselines. As a side contribution, we also release a 34-hour dataset of solo violin recordings, with automatically aligned MIDIs and note-constrained multi-f0 tracks obtained using our descriptive violin transcription system.

The remainder of this paper is organized as follows: in Section 2, we introduce our MUlti-Stream Conformer (MUSC) model for AMT that processes an audio waveform into four musical representations. In Section 3, we describe our strategy for learning without frame-level annotations. In Section 4, we introduce how we simultaneously annotate a novel violin transcription dataset while training our model. In Section 5, we compare our descriptive violin transcription model against general-purpose transcription and pitch estimation baselines. Finally, we conclude in Section 6 with prospects on future work.

[1] https://github.com/MTG/violin-transcription/

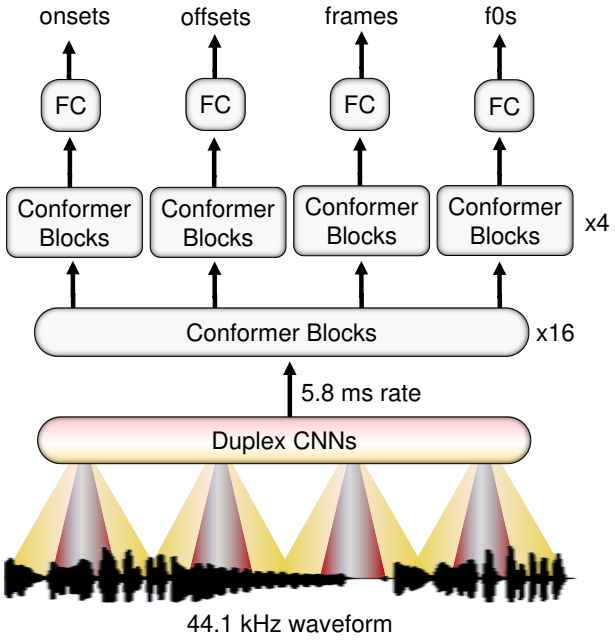


**Figure 2**: The Multi-Stream Conformer architecture converts raw audio sampled with 44.1 kHz into four feature representations with a frame rate of 5.8 ms.

## 2. MULTI-STREAM CONFORMER

We propose a MUlti-Stream Conformer (MUSC) that processes the raw audio waveform into four streams that estimate onset, offset, semitone-level pitch frames (denoted as frames as in the AMT literature), and high-resolution f0 frames as shown in Figure 2. The raw audio waveform sampled with 44.1 kHz is converted into 256-dimensional features with a hop size of 5.8 ms through duplex CNNs. Then, these features pass through the Conformer blocks to estimate the four representations. The resulting representations can be either used for MIDI transcription with pitch bends as in Figure 1, or for frame-level dataset annotation for training (see Section 3).

### 2.1 Duplex CNNs

We borrow the basic CNN structure from the first two layers of the CREPE [29] pitch estimator, except for zero padding. We remove the zero padding in the convolutional layers so that the duplex CNNs can access to the information at the borders of the window with varying receptive fields. With the raw audio in 44.1 kHz as the input, the duplex CNNs independently summarize the waveform into 128-dimensional frames with a hop length of 5.8 ms.

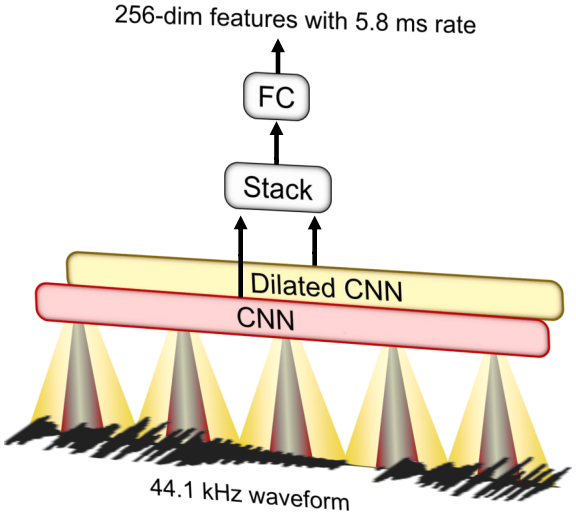


**Figure 3**: A closer look at the Duplex CNNs.

The standard CNN (shown in red in Figure 3) analyzes the frame with the CREPE configuration, resulting in a receptive field of 26 ms. The dilated CNN (depicted in yellow within Figure 3) incorporates double the number of dilations and strides per layer, ultimately leading to a receptive field of 118 ms. Thanks to the dilations and strides, the sampling rate for the dilated CNN is subsequently reduced to 22.05 kHz, and 11 kHz. Thus, it effectively analyzes a smoother version of the signal. The 128-dimensional outputs of the individual CNNs are then stacked into a 256-dimensional representation and pass through a simple fully-connected layer before the main Conformer stream.

### 2.2 Conformer Blocks

Due to the direct analogy between music transcription and speech recognition, we adopt the Conformer [30], a state-of-the-art automatic speech recognition (ASR) model, as the base block of MUSC. We directly employ conformer blocks from the Conformer encoder (M version) as described by Gulati et al. [30], i.e., with four attention heads, a depthwise convolution size of 32, and an encoder dimension of 256. For the main stream, we repeat the conformer blocks 16 times as in Conformer (M). Then, we employ separate conformer blocks for each of the onset, offset, frame, and f0 streams with four conformer blocks per representation. The total number of conformer blocks we utilize in the multi-stream conformer architecture is 32.

### 2.3 Feature Representations

Our method is based on transforming weak labels into frame-level features that are used both as training targets and alignment features. The feature representations encompass the violin pitch range from $F\sharp 3$ to $E8$, i.e., 58 bins for the onsets, offsets, and note frames, which work on semitone resolution, and 580 bins for the f0s, which work on 10-cent resolution. More precisely, we use a fixed sequence duration of three seconds and convert the audio waveform into $512 \times 58$ dimensional onset, offset, and (note) frames, and $512 \times 580$ dimensional f0 frames.

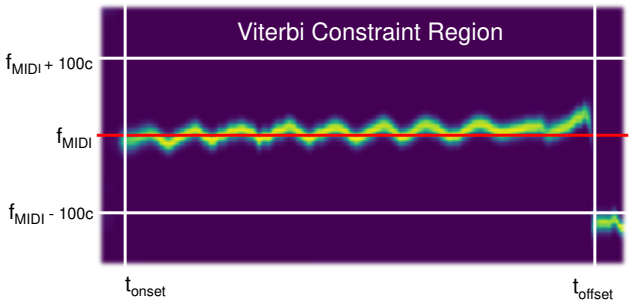


**Figure 4**: Constraint region for the Viterbi pitch tracking.

We train the model to predict strong onset, offset, and frame labels that are generated from iterative score alignments. We employ Gaussian label smoothing for onset, offset, and f0 features. For the onsets and offsets, we smooth the feature representations with a standard deviation of 4 ms. Following Kim et al. [29], we also blur the f0 features with a 12-cent standard deviation.

Note that the high-precision f0 features are not included in the score, hence cannot be inferred from the alignment. For the f0 features, we train the model to predict pseudo-labels generated by the TAPE model [31] in the first iteration. Then, we use our model's predictions as pseudo f0 labels. The polyphonic multipitch information are also encoded in the f0 representations. We employ constrained Viterbi pitch estimation (see Section 2.5) for generating pseudo-f0 labels for the polyphonic segments.

### 2.4 Note postprocessing

In the original Conformer paper [30], which is designed for ASR, the output of the encoder is proceeded by a decoder that uses an external language model to generate the word sequence. A natural adoption of this strategy to our scenario would require onsets, offsets, and frames to be fed into a language model that is specialized in the violin repertoire. However, employing a decoder is not viable since violin repertoire remains a low-resource language, and training decoders with such limited data is prone to overfitting. Instead, we experiment with postprocessing techniques from open-source AMT libraries and adopt the one [2] from Bittner et al. [1]. We leave improving the post-processing stage as an open question for further studies.

### 2.5 Constrained Viterbi Pitch Estimation

Previous studies have shown that score information [32] and the continuity principle of pitch perception [33] can be used for refining the f0 estimation. We apply continuity constraints within note sections to detect the pitch bends with higher accuracy. First, we define the constraint region on the f0 matrix from the note onset, offset, and 200 cents around the note frequency as shown in Figure 4. We calculate the Viterbi path within the note boundaries by utilizing the constraint region as observation probabilities and f0 transition probability matrix $\mathbf{S} \in \mathbb{R}^{21\times 21}$ covering the

[2] https://github.com/spotify/basic-pitch/blob/main/basic_pitch/note_creation.py

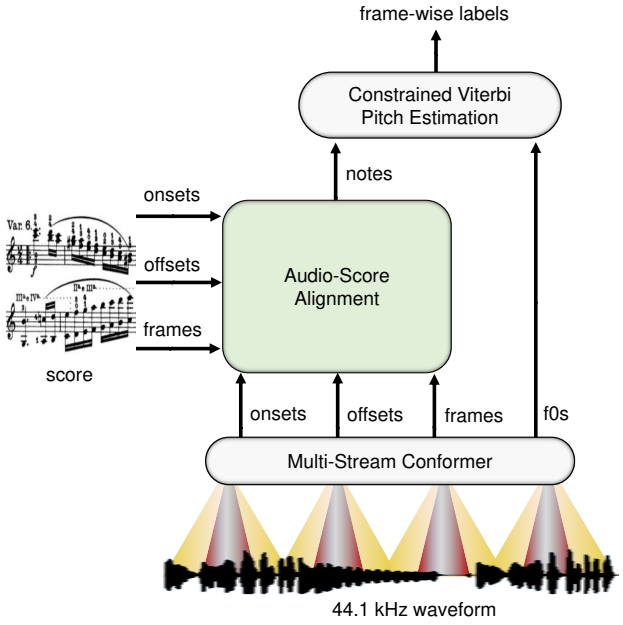


**Figure 5**: The proposed high-resolution violin transcription model only requires piece-level labels for learning as it can generate frame-wise labels using its own onset, offset, and frame feature representations.

200 cents around the note frequency. For each consecutive time instant, $\mathbf{S}$ allows smooth transitions with a Gaussian standard deviation of 25 cents, i.e., 2.5 f0 states:

$$s_{ij} = \frac{\exp\left(-\frac{1}{2}\left(\frac{j-i}{(25/10)}\right)^2\right)}{(25/10)\sqrt{2\pi}},$$

for $i, j \in [1 : 21]$, where $s_{ij}$ denotes the state transition probabilities in the 10-cent resolution f0 matrix.

Since Viterbi algorithm has a complexity of $O(n^2)$, applying the pitch tracking within the constrained region also improves the runtime speed compared to Viterbi without note constraints. Moreover, applying Viterbi within note constraints allow detecting multiple f0s.

After per-note Viterbi paths are calculated, the frame-wise pitch predictions are obtained through the regional weighted averaging method from Kim et al. [29] to deter the f0 estimates through further interpolations.

## 3. LEARNING FROM WEAK LABELS

Our proposed method enables learning from weak labels, which involve pairs of violin recordings and their publicly-available scores. The learning procedure consists of four phases. First, we create initial audio-score alignments using music synchronization techniques. Second, we use the aligned audio-score pair for the first round of training. Third, we recompute the alignment using the estimated features. Fourth and finally, we finetune our model using the finer features learned by the model.

To create the initial audio-score alignments, we use dynamic time warping (DTW), which is a well-known technique for music synchronization [34–36]. Conventional methods for music synchronization typically use DTW and chroma features as the input representation [32, 37], whereas the integration of additional activation functions, e.g., onsets, beats, downbeats, has proven to enhance the synchronization accuracy [27, 28]. Since we deal with violin transcription in this paper, we follow the alignment method in [28], which deals with a similar scenario, i.e., audio-to-audio synchronization of string quartets. Inspired by their combined synchronization approach, we first incorporate beat, downbeat, and onset activation functions alongside chroma features to generate the initial audio-score alignments. The inclusion of activation functions results in a grid-like structure in the DTW cost matrix, which guides the alignment through activation cues that point to note onsets or other musical events. At the same time, chroma features account for the harmonic and melodic information.

Following the setting in [28], we use a sample rate of 22.05 kHz and a feature rate of 50 Hz to create the alignments. As this feature rate (20 ms) is coarser than the model's frame resolution (5.8 ms), we apply linear interpolation to create labels. Note that we cannot evaluate the synchronization accuracy of the training data since we do not have any annotations for these. Using these target labels obtained from the initial alignment, which can possibly be inaccurate, we train our model for one epoch in the first training phase.

Following the first training phase, we obtain the four learned representations, onset, offset, semitone-level frames, and high-resolution f0 frames for each audio-score pair. To acquire finer and more accurate labels, we run a novel synchronization stage. We recompute the alignment with the refined features, estimated semitone-level frame representations, and the activation with the stacked onset and offset features (see Section 2.3). Note that the feature rate we use in the alignment is the same as the MUSC features (hop size of 5.8 ms). Using the labels obtained from synchronization, we finetune our model using early stopping.

Our iterative training strategy resembles the approach by Maman and Bermano [2]. Their approach starts with training the transcription model with synthetic data and then creating the initial alignments with the features estimated by this model and involves three training iterations: first on synthetic data and two more iterations to finetune the model on the target dataset. In contrast, we start from a robust ASA and complete the training process in two iterations.

## 4. DATASET AND TRAINING

In this section, we describe our dataset that we use for the training and our training procedure. The weakly-labeled dataset consists of 120 scores and 34 hours of solo violin performances. We also provide automatic score alignments and frame-level pitch bends that are generated by our joint data curation and training process.

| | #s | #p | #r | dur |
|---|---|---|---|---|
| Paganini, Op. 1 | 24 | 10 | 235 | 13:00 |
| Wohlfahrt, Op. 45 | 60 | 6 | 506 | 11:36 |
| Kayser, Op. 20 | 36 | 8 | 280 | 09:48 |
| **Total** | **120** | **22** | **1021** | **34:24** |

**Table 1**: Dataset statistics. #s: number of scores, #p: number of distinct players, #r: number of recordings, dur: total recording duration in hh:mm.

### 4.1 Dataset Statistics

Our dataset comprises public scores of 96 etudes which are included in the Violin Etudes dataset [38], i.e., Wohlfahrt Op. 45, and Kayser Op. 20. We also extend these scores with additional 24 etudes/caprices by Paganini Op. 1. In contrast to the Violin Etudes dataset, which only includes monophonic recordings, the recordings in our dataset include a mix of monophonic and polyphonic etudes. We collect multiple versions of these etudes from YouTube and automatically match and align them using the method described in Section 3. For the Paganini Op. 1 score, we noticed that performers do not always follow the repeat signs. To ensure better alignments, we automatically expand each repetition pattern individually and select the one that best matches the recording based on the alignment distance. As the most extreme case, we found four different repetition patterns for the Paganini Op. 1 No. 23, which we label as Op01-23, Op01-23-a, Op01-23-b, and Op01-23-c in the dataset, respectively.

The dataset we provide includes original YouTube links, annotated start and end timestamps, and aligned MIDI files containing multi-pitch bends. These resources can be utilized to generate expressive performances featuring vibrato. Moreover, for each etude and caprice, we provide at least five performances, which can be utilized for audio-to-audio synchronization and comparative studies. Table 1 summarizes the dataset statistics.

### 4.2 Training Details

Using Adam optimizer and a learning rate of $1\mathrm{e}{-3}$, we train the model to minimize the binary cross entropy (BCE) loss for the onset, offset, frame, and f0s:

$$\mathcal{L} = \mathcal{L}_{\text{onset}} + \mathcal{L}_{\text{offset}} + \mathcal{L}_{\text{frame}} + \frac{\mathcal{L}_{\text{f0}}}{10}.$$

In addition to Gaussian label smoothing as described in Section 2.3, we weight positive onset and offsets with 9 to balance the sparse matrices. Furthermore, we also observe that weighting the $\mathcal{L}_{\text{f0}}$ by $1/10$ helps in increasing the stability of the training.

Since our dataset includes several versions per piece, we do not employ further data augmentations. We train the model using a batch size of 16 and a fixed sequence duration of three seconds (512 frames). We employ $(80-20)$ train–validation splits and consider each sample with the etude no $\equiv 3 \pmod 5$ for the validation set.

After training for one epoch on the dataset obtained with initial alignments and pseudo f0 labels, we realign the dataset with the model's onset, offset, and frame features and apply constrained Viterbi tracking for the f0 labels. Using the new labels estimated by the model, we train the model further, applying early stopping.

## 5. EXPERIMENTS

While we aim at the task of descriptive violin transcription with high-resolution pitch bends, there is no previous work on which we can directly compare with. Therefore, we compare our model with general-purpose baselines for the closely-related proxy tasks of transcription and pitch estimation. We provide our experimental results on the violin tracks of two manually-annotated and corrected datasets, i.e., URMP [39] and Bach10 [40].

### 5.1 Test Datasets

The URMP dataset [39] is a multimodal dataset that includes 44 performances in various chamber ensemble settings. The dataset was annotated with the help of the Tony melody transcription software [41], which utilizes the pYIN [33] algorithm for the initial f0 estimates and applies a hidden Markov model for note quantization. The note onsets, offsets, and f0s are then manually corrected. For our evaluation, we use all the violin tracks from the URMP dataset. We note that one of our transcription baselines, the MT3 [3] model, was trained using this dataset. Since we employ our tests in the entirety of the violin tracks, the tests include the training samples of the MT3.

Our second test dataset, Bach10 [40], comprises 10 four-part chorales played by a violin, clarinet, tenor saxophone, and bassoon quartet. The ground-truth f0 annotations in the dataset were estimated first using the YIN [42] algorithm and then corrected manually. The dataset also includes note annotations derived from the beat times that are manually-annotated by musicians. However, the manual correction for offset times is not included in the dataset. For our evaluation, we use all the violin tracks from the Bach10 dataset. We note that the Bach10 dataset was included in the training of one of our baselines in pitch estimation, i.e., CREPE [29].

### 5.2 Evaluation Metrics

As a proxy to descriptive violin transcription, we evaluate our method's transcription and pitch estimation performance separately using the common `mir_eval` metrics, and compare with general-purpose baselines. For the transcription, we provide our results with Precision P, Recall R, F1-score F1, and F1-score without offset $\mathrm{F1_{no}}$ using the default thresholds. Namely, for P, R, and F1, a note is considered correct its pitch is within 50 cents, the onset is within 50 ms and the offset is within 20% of the note's duration. We also include an additional measure, $\mathrm{F1_{no}}$, where a note is considered correct if the onset is within 50 ms without considering the offset. For the pitch estimation experiments, we used the Raw Pitch Accuracy (RPA) metric with two thresholds: the standard RPA50 metric, which considers the estimate accurate if it is within 50

| | URMP | | | | Bach10 | | | |
|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | $F1_{no}$ | P | R | F1 | $F1_{no}$ |
| **MUSC** | **86.5** | 83.1 | **84.6** | **93.0** | **65.0** | **64.8** | **64.8** | **77.0** |
| **MT3** | 79.1 | **87.1** | 82.2 | 88.9 | 54.2 | 51.5 | 52.7 | 62.0 |
| **BP** | 58.8 | 67.9 | 62.8 | 83.3 | 33.6 | 43.2 | 37.6 | 57.5 |

**Table 2**: Violin transcription results (%) comparing MUSC with two general-purpose AMT methods. Tests are conducted on all violin stems from the datasets. Bach10 represents the fair evaluation in a dataset unseen to all models. URMP was involved in the training dataset of the MT3, whereas it is unseen to both BP and MUSC.

| | URMP | | | | Bach10 | | | |
|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | $F1_{no}$ | P | R | F1 | $F1_{no}$ |
| **Iter1** | 84.6 | 82.5 | 83.6 | 92.9 | 63.1 | 63.5 | 63.2 | 75.3 |
| **Iter2** | **86.5** | **83.1** | **84.6** | **93.0** | **65.0** | **64.8** | **64.8** | **77.0** |

**Table 3**: Violin transcription results (%) before (Iter1) and after (Iter2) fine-tuning the proposed MUSC model with the iterative alignment.

cents, and the RPA10 metric, which has a more strict 10-cent threshold.

### 5.3 Results

We compare MUSC with two recent general-purpose AMT baselines: Our first baseline is the Basic Pitch [1] (*BP*), which is a lightweight model for instrument-agnostic AMT. The postprocessing method of BP is optimized for $F1_{no}$, and MUSC also shares the same postprocessing script with their default parameters. The second baseline we consider for transcription is the MT3 [3], which is a multi-instrument transcription model that predicts instrument labels alongside transcription. Since we only test on violin recordings, we combine their output without the instrument labels for fair evaluation.

Table 2 summarizes the results for the transcription experiments. At a first glance, the proposed violin-specific model MUSC outperforms MT3 and BP on both datasets, indicating that it is a more effective method for violin transcription. Even though the training set of MT3 included the test samples in the URMP dataset, MUSC yields the best F1-score value among the three AMT systems. Furthermore, the performance gap between MUSC and MT3 is greater for the Bach10, which was not included the training set of any method. The results indicate that the all the models yield rather poor scores on the Bach10 dataset when evaluated using the conventional P, R, and F1 metrics. Since the offsets in the Bach10 dataset are not manually-corrected, the $F1_{no}$ scores can be viewed as a better indicator of the transcription performance for this dataset.

We also compare our model's transcription performance before and after fine-tuning with alignments generated using its own feature representations. The Table 3 shows that some of the improvements in our model's transcription performance can be attributed to the iterative training strategy.

For the pitch estimation experiments, we compare MUSC with four well-known pitch estimators: the pre-trained CREPE model [29] from its official repository [3], pYIN [33], and YIN [42] from librosa [4], and SWIPE [43] from the libf0 library [5]. We use the same $F\sharp 3$ (min) to $E8$ (max) frequency range for a fair evaluation.

| | URMP | | Bach10 | |
|---|---|---|---|---|
| | RPA50 | RPA10 | RPA50 | RPA10 |
| **MUSC** | 98.3 | 89.0 | 98.3 | 86.9 |
| **vMUSC** | **98.6** | **89.4** | 98.4 | 87.0 |
| **CREPE** | 96.4 | 87.2 | **98.6** | **88.1** |
| **vCREPE** | 97.3 | 88.4 | **98.6** | **88.1** |
| **YIN** | 95.3 | 88.4 | 97.1 | 81.7 |
| **pYIN** | 97.2 | 88.6 | 97.4 | 80.3 |
| **SWIPE** | 97.2 | 89.3 | 97.7 | 84.3 |

**Table 4**: Violin Raw Pitch Accuracy (RPA, %) results. Note that the training set of CREPE involved the Bach10 dataset. vMUSC and vCREPE contain an additional Viterbi decoding stage.

Table 4 summarizes the pitch estimation results. First, all the pitch estimators achieve high accuracies on both datasets. For the URMP dataset which is unseen to all the models, vMUSC (MUSC with Viterbi decoding) outperforms the common state-of-the-art pitch estimators in terms of RPA50 and RPA10. For the Bach10 dataset, which is included in the training samples of the pre-trained CREPE model, the CREPE expectedly yields the best RPA values. Note that even though our model was not trained with these test samples from Bach10, MUSC remains to be competitive (e.g., 98.4% versus 98.6% RPA50 in Bach10).

## 6. CONCLUSION

In this paper, we introduced MUSC, an AMT system tailored for violin transcription through high-precision pitch bend estimation, and the capability of learning from piece-wise weak labels. We showed that, by only utilizing 120 scores, we were able to obtain state-of-the-art transcription and pitch estimation results for the violin. We also shared our descriptive violin transcription dataset to the MIR community. In the future, we will focus on improving the note postprocessing and alignment stages of the MUSC in order to specialize better for the string repertoire, and use it as a large-scale dataset curation tool for strings music, ethnomusicology, and music education research. We believe that the descriptive music transcription capabilities of the MUSC will accelerate the research in music education, ethnomusicology, and expressive performance generation.

## 7. ACKNOWLEDGEMENTS

This research is funded by the project Musical AI - PID2019-111403GB-I00/AEI/10.13039/501100011033 funded by the Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU) and the Agencia Estatal de Investigación (AEI), and by the German Research Foundation (DFG MU 2686/10-2).

[3] https://github.com/marl/crepe
[4] https://librosa.org/
[5] https://github.com/groupmm/libf0

## 8. REFERENCES


[1] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, 2022.

[2] B. Maman and A. H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *Proceedings of the International Conference on Machine Learning (ICML)*, July 2022, pp. 14 918–14 934.

[3] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, "MT3: multi-task multitrack music transcription," *Computing Research Repository (CoRR)*, vol. abs/2111.03017, 2021. [Online]. Available: https://arxiv.org/abs/2111.03017

[4] K. W. Cheuk, D. Herremans, and L. Su, "Reconvat: A semi-supervised automatic music transcription framework for low-resource real-world data," in *Proceedings of the ACM Multimedia Conference*, H. T. Shen, Y. Zhuang, J. R. Smith, Y. Yang, P. Cesar, F. Metze, and B. Prabhakaran, Eds., Virtual Event, China, 2021, pp. 3918–3926.

[5] I. Simon, J. Gardner, C. Hawthorne, E. Manilow, and J. Engel, "Scaling polyphonic transcription with mixtures of monophonic transcriptions," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 749–756.

[6] S. Ewert and M. B. Sandler, "Piano transcription in the studio using an extensible alternating directions framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1983–1997, 2016.

[7] R. Kelz and G. Widmer, "Towards interpretable polyphonic transcription with invertible neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, November 2019, pp. 376–383.

[8] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[9] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, "Sequence-to-sequence piano transcription with transformers," pp. 246–253, 2021.

[10] R. Schramm and E. Benetos, "Automatic transcription of a cappella recordings from multiple singers," in *Proceedings of the AES International Conference on Semantic Audio*, Erlangen, Germany, 2017, pp. 108–115.

[11] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii, "Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 161–165.

[12] T.-W. Su, Y.-P. Chen, L. Su, and Y.-H. Yang, "TENT: Technique-embedded note tracking for real-world guitar solo recordings," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 2, no. 1, July 2019.

[13] A. Wiggins and Y. E. Kim, "Guitar tablature estimation with a convolutional neural network." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, Netherlands, November 2019, pp. 284–291.

[14] J. Abeßer and M. Müller, "Jazz bass transcription using a U-net architecture," *Electronics*, vol. 10, no. 6, p. 670, 2021.

[15] M. A. Kaliakatsos-Papakostas, A. Floros, M. N. Vrahatis, and N. Kanellopoulos, "Real-time drums transcription with characteristic bandpass filtering," in *Proceedings of the Audio Mostly: A Conference on Interaction with Sound*, Corfu, Greece, September 2012, pp. 152–159.

[16] C. Southall, R. Stables, and J. Hockman, "Automatic drum transcription using bi-directional recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, August 2016, pp. 591–597.

[17] K. Choi and K. Cho, "Deep unsupervised drum transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 183–191.

[18] C. Seeger, "Prescriptive and descriptive music-writing," *The Musical Quarterly*, vol. 44, no. 2, pp. 184–195, 1958.

[19] P. C. Greene, *Violin performance with reference to tempered, natural, and Pythagorean intonation*. University of Iowa Press, 1937.

[20] J. M. Geringer, "Eight artist-level violinists performing unaccompanied bach: Are there consistent tuning patterns?" *String Research Journal*, vol. 8, no. 1, pp. 51–61, 2018.

[21] G. N. Swift, *The violin as cross cultural vehicle: Ornamentation in South Indian violin and its influence on a style of Western violin improvisation*. Wesleyan University, 1989.

[22] C. Weiß and G. Peeters, "Learning multi-pitch estimation from weakly aligned score-audio pairs using a multi-label CTC loss," in *Proceedings of the IEEE*

*Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2021, pp. 121–125.

[23] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, USA, 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[24] D. Schwarz, N. Orio, and N. Schnell, "Robust polyphonic midi score following with hidden Markov models," in *International Computer Music Conference (ICMC)*, Miami, Florida, USA, 2004.

[25] M. Dorfer, A. Arzt, and G. Widmer, "Towards score following in sheet music images," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 789–795.

[26] T. Kwon, D. Jeong, and J. Nam, "Audio-to-score alignment of piano music using RNN-based automatic music transcription," in *Proceedings of the Sound and Music Computing Conference (SMC)*, Espoo, Finland, 2017, pp. 380–385.

[27] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[28] Y. Özer, M. Istvanek, V. Arifi-Müller, and M. Müller, "Using activation functions for improving measure-level audio synchronization," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 749–756.

[29] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 161–165.

[30] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2020, pp. 5036–5040.

[31] N. C. Tamer, Y. Özer, M. Müller, and X. Serra, "TAPE: An end-to-end timbre-aware pitch estimator," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.

[32] M. Müller, *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*, 2nd ed. Springer Verlag, 2021.

[33] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 659–663.

[34] S. Salvador and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," in *Proceedings of the KDD Workshop on Mining Temporal and Sequential Data*, 2004.

[35] S. Dixon and G. Widmer, "MATCH: A music alignment tool chest," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, London, UK, 2005, pp. 492–497.

[36] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync Toolbox: A Python package for efficient, robust, and accurate music synchronization," *Journal of Open Source Software (JOSS)*, vol. 6, no. 64, pp. 3434:1–4, 2021.

[37] R. B. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors," in *Proceedings of the International Computer Music Conference (ICMC)*, San Francisco, USA, 2003, pp. 27–34.

[38] N. C. Tamer, P. Ramoneda, and X. Serra, "Violin etudes: a comprehensive dataset for f0 estimation and performance analysis," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 517–524.

[39] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 522–535, 2019.

[40] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2121–2133, 2010.

[41] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided melody note transcription using the Tony software: Accuracy and efficiency," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2015.

[42] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music." *Journal of the Acoustical Society of America (JASA)*, vol. 111, no. 4, pp. 1917–1930, 2002.

[43] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.