WEAKLY SUPERVISED MULTI-PITCH ESTIMATION USING CROSS-VERSION ALIGNMENT

Michael Krause, Sebastian Strahl, Meinard Müller
International Audio Laboratories Erlangen, Germany
{michael.krause,sebastian.strahl,meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Multi-pitch estimation (MPE), the task of detecting active pitches within a polyphonic music recording, has garnered significant research interest in recent years. Most state-of-the-art approaches for MPE are based on deep networks trained using pitch annotations as targets. The success of current methods is therefore limited by the difficulty of obtaining large amounts of accurate annotations. In this paper, we propose a novel technique for learning MPE without any pitch annotations at all. Our approach exploits multiple recorded versions of a musical piece as surrogate targets. Given one version of a piece as input, we train a network to minimize the distance between its output and time–frequency representations of other versions of that piece. Since all versions are based on the same musical score, we hypothesize that the learned output corresponds to pitch estimates. To further ensure that this hypothesis holds, we incorporate domain knowledge about overtones and noise levels into the network. Overall, our method replaces strong pitch annotations with weaker and easier-to-obtain cross-version targets. In our experiments, we show that our proposed approach yields viable multi-pitch estimates and outperforms two baselines.

1. INTRODUCTION

Music transcription, i. e., converting music audio recordings into score representations, is a fundamental task in music information retrieval (MIR). As a subtask of transcription, one may estimate the pitches active at different points in time throughout a recording of polyphonic music, yielding a piano roll representation (without considering instrumentation, note values, or other score-based information). This goal is commonly referred to as multi-pitch estimation (MPE). Recent years have seen significant advances in MPE systems, mainly due to the use of deep learning models [1–6]. These models are typically trained with large amounts of aligned pitch annotations as targets, see also Figure 1a. Creating such annotations may involve an enormous effort. In particular, manually annotating pitch activity in every frame of an audio recording would be prohibitively time consuming. Many datasets are thus annotated using semi-automatic methods like score–audio synchronization (e. g., [7]), which introduces annotation errors. Because of this, systems that can learn pitch estimation without large amounts of pitch annotations are highly desirable.

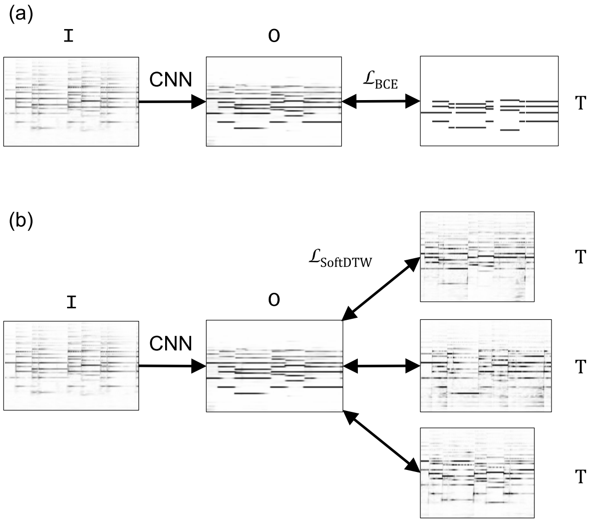

Figure 1: Systems for multi-pitch estimation are typically trained using pitch annotations (a), which are cumbersome to create. In this work, we propose to use different versions of a piece as surrogate targets (b), which are much easier to obtain. In both scenarios, a network input (I) is passed through convolutional layers, producing an output (O), which is compared to one or several targets (T) using some loss function ($\mathcal{L}$).

In this paper, we propose a novel approach for learning MPE without pitch annotations. As our key idea, we use different versions (i. e., recorded performances) of a musical piece as surrogate targets. To this end, we leverage cross-version music datasets, which contain several versions per piece. Such datasets are especially common for Western classical music, where the same compositions are regularly performed by different musicians. Each version exhibits unique timing, artistic expression, and varying acoustic conditions. All versions, however, are based on the same musical score and thus contain the same combinations of pitches. We therefore hypothesize that a deep network may produce pitch estimates by learning the commonalities between different versions of a piece.

Attribution: M. Krause, S. Strahl, and M. Müller, "Weakly Supervised Multi-Pitch Estimation Using Cross-Version Alignment", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

In our approach, we train a deep network that takes a time–frequency representation of one version as input, and whose output minimizes a certain distance to time–frequency representations of other versions. This core idea is illustrated in Figure 1b. Since versions vary in length and the timing of pitch events may be different, we require a distance measure that temporally aligns the network output to the representations of other versions. To do so within a deep learning setting, we use a differentiable variant of dynamic time warping called SoftDTW [8]. Apart from the fundamental frequencies of pitches played, all recorded versions of a piece contain overtone structures and ambient noise. To increase the validity of our hypothesis and to encourage the network to capture nothing but pitches, we incorporate knowledge about overtones and noise using additional fixed processing blocks.

Overall, our proposed approach replaces the need for strong pitch annotations (which are frame-wise, binary, and difficult-to-obtain) with weaker cross-version targets (not temporally aligned, real-valued, and easy-to-obtain).

In summary, we make the following contributions: We propose a novel approach for weakly supervised MPE that does not require pitch annotations, based on the hypothesis that pitch estimation can be learned from multiple versions. We further propose to incorporate extra layers for simulating overtones and noise levels to ensure that our hypothesis holds. Finally, as a proof of concept, we show qualitatively and quantitatively that our approach can be used for MPE and outperforms two baselines. To aid reproducibility, we release code and trained models for our approach. [1]

The remainder of this paper is structured as follows: In Section 2, we discuss related work on pitch estimation. In Section 3, we describe our proposed approach. Section 4 covers the experimental setup, while Section 5 contains our results. Section 6 concludes the paper with an overview of possible directions for future work.

2. RELATED WORK ON MULTI-PITCH ESTIMATION

The majority of work on MPE and music transcription in general has focused on supervised training schemes, where a dataset of music recordings with aligned pitch annotations is given. Most recent papers utilize deep learning models that are trained with pitch targets using standard cross-entropy loss functions [1–5]. Often, these works focus on piano music, where annotations can be obtained using MIDI recording technology built into certain types of pianos [9]. We refer to [6] for an overview of music transcription research.

Some works have explored pitch estimation from data without aligned pitch annotations. Weiß and Peeters [10] proposed to utilize weakly aligned annotations, where there may be temporal deviations between recorded performance and annotations. This scenario is also explored in [11]. However, in both cases, pitch annotations are required for the entire training dataset. Gfeller et al. [12] introduced a self-supervised approach for pitch estimation, where a network learns to predict the relative differences between pitch-shifted, monophonic recordings. Their approach requires only a small amount of data with pitch annotations, but does not deal with polyphonic scenarios. Berg-Kirkpatrick et al. [13] describe a system for MPE on piano recordings that does not use pitch annotations. Their approach solves an optimization problem, with constraints motivated by the sound production process in pianos. In contrast, the method we propose in this paper utilizes several versions of a musical piece and could be used for recordings with arbitrary instruments.

3. PROPOSED METHOD

We now describe our proposed approach for learning MPE using cross-version alignment. Here, we assume that we have multiple corresponding recorded versions for each musical piece in the training set. Let us denote the set of all corresponding versions for one piece by $\mathcal{V} = \{V_1, V_2, \dots\}$. Furthermore, given a version $V \in \mathcal{V}$, we write $\text{InputRep}(V)$ for the audio representation of V that our network takes as input. [1]

Given an input $\mathtt{I} = \text{InputRep}(V)$, we formulate MPE as the problem of producing a binary piano roll $\tilde{\mathtt{M}} \in \{0,1\}^{B \times N}$ that matches the pitch annotations $\mathtt{A} \in \{0,1\}^{B \times N}$ for that input. Here, B denotes the number of pitch bins, while N is the number of time frames in the input. In the supervised case, deep networks for MPE produce a real-valued output $\mathtt{O} \in [0,1]^{B \times N}$ that is optimized using the binary cross-entropy loss $\mathcal{L}_{\text{BCE}}$ with $\mathtt{T} = \mathtt{A}$ as targets (where the loss is averaged over all time–pitch bins). The final pitch predictions $\tilde{\mathtt{M}}$ are obtained from $\mathtt{O}$ by applying a threshold τ. This threshold is often set to a fixed value (e. g., $\tau = 0.4$ in [7]) or optimized on a validation dataset [14]. This supervised approach to MPE, which crucially relies on the aligned pitch annotations $\mathtt{A}$, is illustrated in Figure 1a. In the following, we will refer to it with the shorthand Sup.

Our proposed approach, illustrated in Figure 1b, also takes an input representation $\mathtt{I} = \text{InputRep}(V)$ for some version $V \in \mathcal{V}$. As before, our network yields a real-valued output $\mathtt{O} \in [0,1]^{B \times N}$. However, rather than using pitch annotations $\mathtt{A}$, we utilize a surrogate target $\mathtt{T} = \text{TargetRep}(V')$ based on another version $V' \in \mathcal{V} \setminus \{V\}$. We choose a time–frequency representation $\text{TargetRep}(V') \in [0,1]^{B \times N'}$ as target that is normalized in the range $[0,1]$ and has the same number of bins B as $\mathtt{O}$, but a potentially different number of time frames N', due to the temporal differences between versions. [1] As explained in the introduction, $\mathtt{T}$ contains the same combinations of pitches as $\mathtt{I}$. [2] Intuitively, if $\mathtt{O}$ is close to the target representations of all versions $\mathcal{V} \setminus \{V\}$, we hypothesize that $\mathtt{O}$ must correspond to pitch estimates for $\mathtt{I}$. We

[1] https://www.audiolabs-erlangen.de/resources/MIR/2023-ISMIR-WeaklySupervisedMPE

[1] Details of InputRep and TargetRep are provided in Section 4.

[2] Here, we assume that there are no structural differences between versions, i. e., performers do not deviate from the score. Versions performed in different keys can be handled through pitch shifting, see Section 4.

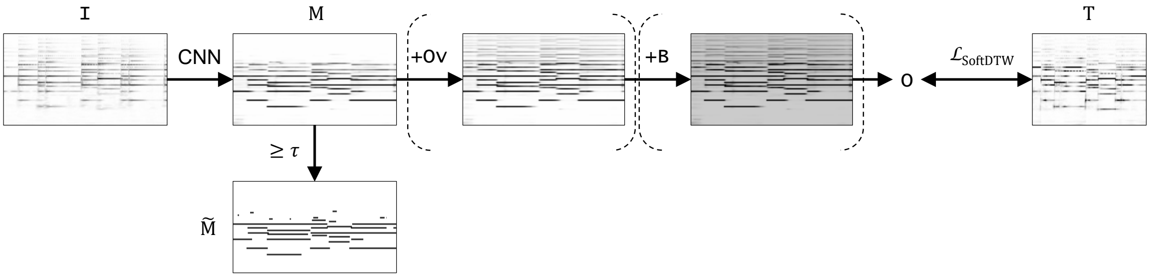

Figure 2: Detailed overview of the proposed cross-version alignment (`CVA`) method, see also Figure 1b. Before applying the alignment loss ($\mathcal{L}_{\mathrm{SoftDTW}}$), the intermediate output of the network (`M`) is optionally extended using a simple overtone model (`+Ov`) and a bias value (`+B`) to address background noise. The final output `O` can thus arise from different configurations (e. g., `O` = `M`, `O` = `M+Ov`, . . .). Importantly, the MPE output of the system ($\tilde{\mathtt{M}}$) is computed based on the intermediate representation `M`, rather than the output `O`.

refer to our proposed approach with the shorthand `CVA` (for "cross-version alignment").

Note that we cannot directly apply a loss on time–pitch bins here (as in the supervised case), since `O` and `T` are not temporally aligned. For this reason, we use the differentiable alignment loss $\mathcal{L}_{\mathrm{SoftDTW}}$ in our approach, see Section 3.1. Furthermore, our hypothesis may fail to apply, since recorded versions of a piece contain overtone structures and background noise in addition to the pitches played. We thus extend our approach to account for these properties of music recordings in Section 3.2.

3.1 Differentiable Alignment

In order to perform temporal alignment between `O` and a target representation `T` in a differentiable fashion, we use the SoftDTW loss [8]. SoftDTW is a differentiable approximation of the classical dynamic time warping algorithm that is often used to align music sequences [15]. SoftDTW has originally been introduced for one-dimensional time series but has also been adopted for computer vision tasks like action recognition in video recordings [16,17]. Within MIR, SoftDTW has previously been used in the context of music synchronization [18] and MPE [11]. In [11], the authors showed that SoftDTW can be used to replace strongly aligned (i. e., frame-wise) pitch annotations with weakly aligned pitches without a major impact on MPE performance. Nevertheless, their approach requires pitch annotations for training.

In our case, we crucially rely on the ability of SoftDTW to align real-valued sequences such as time–frequency representations of audio. In contrast, a commonly used alternative loss function called connectionist temporal classification (CTC) can only handle discrete target sequences. To compute $\mathcal{L}_{\mathrm{SoftDTW}}$, one needs to choose a local cost function (for comparing individual frames of the time–frequency representations) and set a temperature hyperparameter called γ (which determines the approximation quality of SoftDTW). Here, we use the cosine distance for comparing frames, which exhibited high training stability in our experiments. We further set $\gamma = 0.1$, which corresponds to a good approximation of DTW.

As a drawback, the time and space complexity of SoftDTW is quadratic in the lengths of the input sequences. We thus train on short input excerpts (see Section 4).

3.2 Overtone and Noise Model

Aside from differentiable alignments, our proposed approach utilizes fixed processing layers that simulate overtone structures and background noise. In this way, our method follows the analysis-by-synthesis paradigm [19], where one estimates parameters from an audio recording (pitches, in our case) by re-synthesizing the input. Choi and Cho [20] utilized this idea for unsupervised drum transcription. Their network consists of a transcription stage and a fixed sample-based drum synthesizer. The transcription network is trained by minimizing a reconstruction loss on the synthesizer output. In recent years, such systems have become more popular due to the release of the differentiable digital signal processing (DDSP) library [21], which has been used, e. g., in the context of unsupervised monophonic pitch estimation [22]. In contrast to these works, our proposed approach utilizes cross-version data.

A full overview of our `CVA` approach is given in Figure 2. We explicitly add overtones (denoted by `+Ov`) and background noise (`+B`) to an intermediate output `M` of our network via dedicated layers. In this way, the network may learn a sparser and more piano roll-like representation `M`, since overtones and noise are added afterwards. Crucially, the final MPE results $\tilde{\mathtt{M}}$ are obtained from `M`, before overtones and noise are applied. The output `O`, used for alignment with the cross-version targets, depends on the model configuration used. For example, `O` = `M+Ov+B` if all modules are used, `O` = `M+Ov` if only overtones are added, etc. In the basic system without extensions, `O` = `M`.

Here, we opt for very simple overtone and noise models that serve to indicate the potential of our core idea. We estimate the relative amplitudes of different harmonics from a small internal dataset of single-note piano recordings. The resulting estimates, used for our overtone model, are illustrated in Figure 3. We keep these values fixed for all subsequent experiments. To apply this fixed overtone model within our network in a differentiable fashion, we sum up pitch-shifted versions of `M`. For each harmonic h, we shift `M` along the vertical axis by a number of semitones corre-

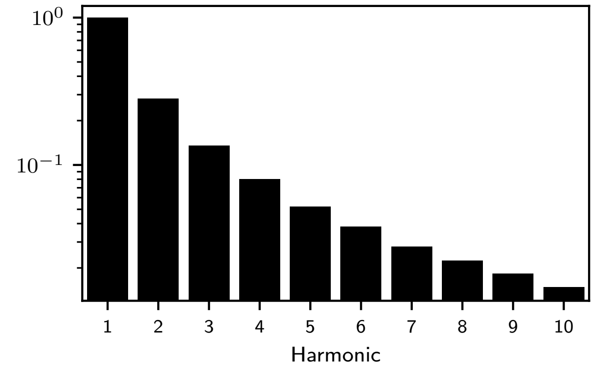

Figure 3: Amplitudes for the overtone model (+Ov) employed in our proposed approach.

sponding to h (e. g., 12 semitones for $h = 2$). We then weight the shifted representation with the amplitude estimated for h (see Figure 3). The final output is obtained by summing the resulting representations for all h.[3] To address the overall noise level in the target T, we add a fixed bias term of $\delta = 0.2$ after applying the overtone model. As a result of this additional processing, we may obtain outputs larger than 1. We therefore clip all values outside the interval $[0, 1]$ (corresponding to the value range of the target representations T) to get the final output O.

4. EXPERIMENTAL SETUP

4.1 Model, Representations, and Training

In this work, we focus on demonstrating the potential of our cross-version approach compared to traditional, fully supervised training for MPE. Thus, we do not propose complicated network architectures that require extensive tuning. Instead, we use a relatively small convolutional neural network for extracting the representation M from I. For InputRep and TargetRep, we use time–frequency representations based on the constant-Q transform (CQT), which provides a frequency axis corresponding to semitones. Note that we cannot train on entire (several minutes long) recordings in a single step. Instead, our training batches contain short input excerpts and we use state-of-the-art music synchronization techniques [23] to find the corresponding sections in other versions.

Concretely, we use the network architecture, input representation, and training setup from [10] (we refer to their paper for details). Their network consists of five convolutional layers with musically motivated kernel shapes and roughly 50 000 learnable parameters. The network takes a magnitude harmonic CQT (HCQT [24]) of an audio excerpt as InputRep, containing $N = 500$ frames computed with a hop size of 512 from waveforms at 22 050 Hz (i. e., an excerpt of 11.6 seconds length). The network produces outputs M of the same length, with a pitch axis containing $B = 72$ bins (corresponding to the semitones from C1 to B6). The final layer of the network contains a sigmoid activation, such that all values in M are restricted to the interval $[0, 1]$. For TargetRep, we use magnitude CQTs where the center frequencies of different bins correspond to the same $B = 72$ semitones. Column-wise max-normalization is applied on T, such that the target values are also in $[0, 1]$.

We train our network by minimizing the SoftDTW loss over all training excerpts until the validation loss has stopped improving for 12 epochs. In each training step, we compute the loss on a batch of 16 inputs. Each input excerpt is based on some version $V \in \mathcal{V}$ and aligned to the corresponding excerpt in one randomly selected target version $V' \in \mathcal{V} \setminus \{V\}$. We use the Adam optimizer with a learning rate of 0.001, which is reduced whenever the validation loss has not improved for three epochs. Finally, we employ an efficient CUDA implementation of the SoftDTW recursions by Maghumi et al. [25].[4]

4.2 Dataset and Split

To train our cross-version approach, we require a dataset containing multiple versions per piece. For testing, we additionally require aligned pitch annotations for the recordings. We opt for using the Schubert Winterreise Dataset (SWD, [26]) for training, which contains nine versions of the 24 songs in the cycle "Winterreise" composed by Franz Schubert (in total, roughly 11 h of audio). Each song constitutes one unique musical piece. The recordings consist of a tenor or baritone singer accompanied by piano. There are no structural differences between versions. Thus, all recordings for a piece contain the same combinations of pitches up to transposition (a global pitch shift), since some musicians chose to perform some songs in different keys. When training our CVA approach, we ensure that input and target version are in the same key by appropriately shifting the target CQT representation according to the key annotations given in the dataset.

We train and evaluate our model using a challenging split where the train and test sets contain both different versions and different songs. We choose songs 1–13 for training, 14–16 for validation, and 17–24 for testing. Furthermore, versions HU33 and SC06 are used for testing, while the remaining seven versions are used for training and validation. Such a split is also referred to as a "neither split", since neither the same versions nor songs appear during training and testing [27]. This split avoids over-optimistic evaluation due to confounders such as the "album effect" [28].

4.3 Baselines

Aside from the supervised baseline Sup, which is trained using strong pitch annotations, we compare our proposed CVA approach to two additional baselines. With these, we aim to evaluate our hypothesis that cross-version targets are useful for learning MPE-like representations (see

[3] Equivalently, the overtone model can be understood as a frame-wise convolution in pitch direction, with a kernel based on the amplitudes in Figure 3.

[4] Note that, within one batch, the targets T may have different lengths. In order to benefit from parallelization across the batch dimension, we therefore rescale the targets T to a common length $N' = 500$ (a trick referred to as W4 in [11]). This did not affect results negatively in early experiments. Note that rescaling is not equivalent to temporally aligning inputs and targets.

Scenario	**CS**	**AP**	$\tau = 0.4$ **F**	 **Acc.**	$\tau = \tau^*$ **F**	 **Acc.**
CQT	0.585	0.410	0.443	0.287	0.450	0.292
AE	0.588	0.500	0.336	0.203	0.511	0.345
CVA	0.632	0.589	0.585	0.416	0.592	0.423
CVA+Ov	0.664	0.639	0.553	0.384	0.623	0.455
CVA+B	0.633	0.563	0.560	0.392	0.592	0.424
CVA+Ov+B	0.682	0.646	0.625	0.458	0.627	0.460
Sup	0.748	0.753	0.700	0.543	0.703	0.546

Table 1: Results for multi-pitch estimation on the Schubert Winterreise Dataset for the baselines and different configurations of our proposed approach.

Section 3). For the CQT baseline, we take the target representations of our test recordings (which are normalized to have values in the range $[0, 1]$) and obtain multi-pitch estimates by directly thresholding these magnitude CQTs with τ. This learning-free baseline was previously proposed in [10] and, like CVA, does not require pitch annotations. Furthermore, we consider a second baseline that is very similar to CVA but does not utilize cross-version targets. Therefore, for each input excerpt, we choose the same version $V \in \mathcal{V}$ for both I and T. Thus, the network needs to effectively recreate its input, similar to an autoencoder. We refer to this baseline with the shorthand AE. Intuitively, we expect CQT and AE to yield similar results. However, AE allows us to verify that any improvements observed for CVA stem from the cross-version targets and not from the model architecture or training setup. Note that Sup and AE use the same network architecture as CVA.

4.4 Evaluation Metrics

We evaluate the multi-pitch estimates of our proposed approach and all baselines using standard metrics on the test set. For this, we utilize the strongly aligned pitch annotations provided in the test data. As metrics, we use the cosine similarity (CS) between predictions and annotations, averaged over all frames and files in the test set. Furthermore, we compare the average precision (AP, computed as the area under the precision-recall curve), F-measure (F), and the accuracy (Acc.) metric introduced in [29]. For these measures, we average over all pitches. Note that F and Acc. are evaluated on $\tilde{\mathtt{M}}$ and thus depend on the threshold τ, while CS and AP are threshold-free evaluation metrics that directly compare M and A.

5. RESULTS

The main results of our study are summarized in Table 1. Rows correspond to different baselines or configurations of our proposed approach. We write +Ov when adding overtones and +B when including the bias term to account for background noise. Our model including all proposed modules is thus referred to as CVA+Ov+B. Columns contain the evaluation metrics. For the thresholding-based metrics F and Acc., we provide both results based on a fixed threshold ($\tau = 0.4$) and a threshold chosen to optimize F on the validation set ($\tau = \tau^*$).

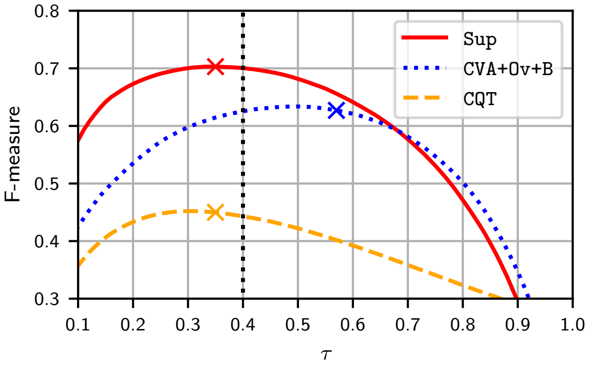

Figure 4: F-measures on the test set for different MPE approaches, depending on the choice of threshold τ. Markers show the optimal threshold τ^* as determined on the validation set.

Our proposed approach CVA outperforms the two baselines CQT and AE across all metrics, demonstrating the effectiveness of using different versions of a piece to capture pitches in M. For example, CS $= 0.632$ for CVA compared to CS $= 0.588$ for AE, and AP $= 0.589$ for CVA compared to AP $= 0.410$ for CQT. Furthermore, our proposed overtone and noise models are effective. By adding overtones (CVA+Ov), we can further increase AP from 0.589 to 0.639. Adding a fixed bias term (CVA+B) does not yield improvements by itself. However, by combining both modules (CVA+Ov+B), we achieve the best results for our approach, further increasing AP to 0.646 and CS to 0.682.

Despite these encouraging results, there remains a gap between the best results for our proposed approach and those for the supervised baseline Sup. We emphasize again that—unlike CVA—Sup requires strong pitch annotations for training.

5.1 Impact of Threshold τ

When using the standard value of $\tau = 0.4$ for thresholding M, our CVA approach also outperforms both baselines in terms of F-measure and accuracy (e. g., F $= 0.553$ for CVA+Ov compared to 0.443 for CQT).

A fixed threshold may be sub-optimal, especially for methods that are not explicitly trained for MPE. When evaluating using the optimized threshold τ^*, we observe increased results for all approaches. CVA and its extensions continue to outperform the two baselines. The F-measure for CVA+Ov, for example, further increases to F $= 0.623$. For that model, the optimal threshold as determined on the validation set is $\tau^* = 0.28$. In this case, our method requires at least a few pitch annotations to determine τ^* and is no longer relying solely on the cross-version targets.

Figure 4 further demonstrates the impact of the parameter τ. F-measures (vertical axis) are shown for different MPE approaches (colored lines), depending on the choice of τ (horizontal axis). Markers indicate τ^*. As shown in this figure, a poor choice of τ may strongly affect test results. Moreover, τ^* as found using the validation set may not always give the highest scores on the test set. For in-

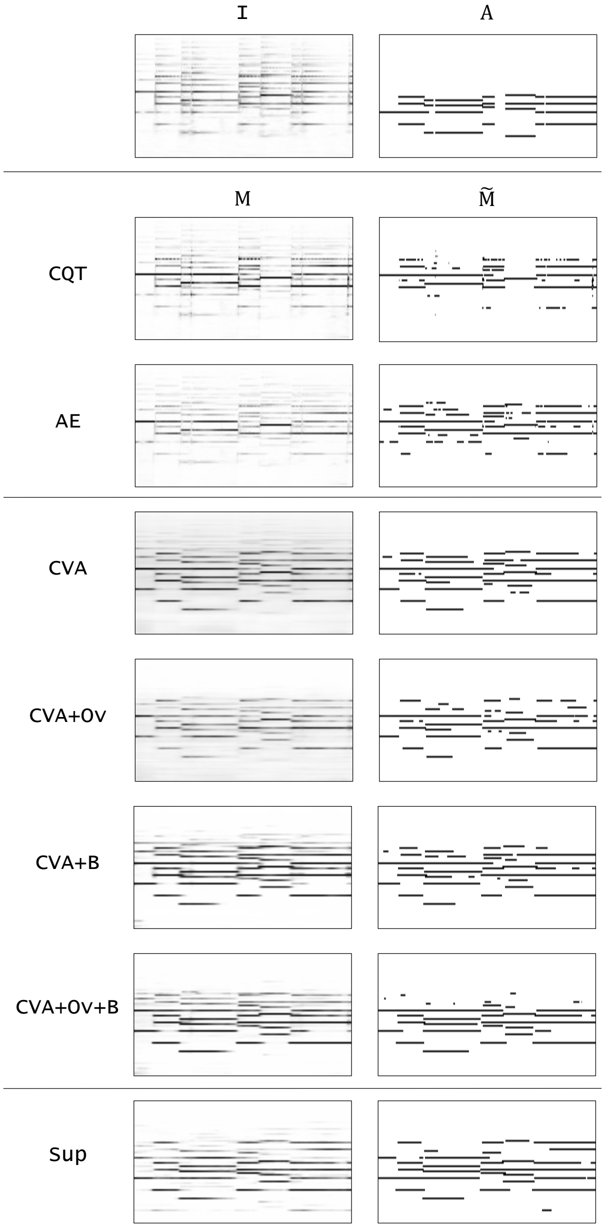

Figure 5: Qualitative results on a test excerpt from SWD.

stance, a choice of $\tau = 0.5$ would yield an even higher F-measure of 0.634 for CVA+Ov+B.

5.2 Training Stability

The metrics reported in Table 1 are computed from a single training run per method. When repeating the experiment, results may deviate slightly due to random network initialization, dataset shuffling, or dropout. For CVA+Ov+B, we repeat the experiment five times and find low standard deviation σ in results ($\sigma(\text{CS}) = 0.004$, $\sigma(\text{AP}) = 0.009$, $\sigma(\text{F}) = 0.008$, and $\sigma(\text{Acc.}) = 0.009$ for $\tau = \tau^*$).

5.3 Qualitative Results

To complement the quantitative evaluation, we also provide qualitative results on an exemplary excerpt in Figure 5. The first row shows an input excerpt (I) and corresponding pitch annotations (A), while the remaining rows show multi-pitch estimates before (M) and after thresholding (M̃, computed using $\tau = \tau^*$).

For CQT and AE, the resulting M correspond to the input representation and thus lead to poor multi-pitch estimates.

When training our approach without overtones or noise model (CVA), the output representation M emphasizes the fundamental frequencies of many of the actual pitches being played. However, M also contains a lot of energy from overtone structures and background noise. As a consequence, the resulting M̃ contains many spurious pitch predictions, especially for higher pitches.

With +Ov and +B, we see a reduced impact of overtones or background noise in M, respectively. In both cases, many erroneous predictions remain after thresholding. By including both modules (CVA+Ov+B), we obtain a promising representation that bears visual resemblance to the results for Sup. We also observe fewer spurious activations in M̃ compared to the basic CVA. Overall, the proposed extensions are effective in encouraging the model to produce MPE predictions in M.

6. CONCLUSION

In this paper, we presented a novel approach for MPE that does not require pitch annotations for training. Instead, our method utilizes multiple versions of the same musical piece as surrogate targets. We train a network that takes a time–frequency representation of one version as input and minimizes an alignment-based distance to time–frequency representations of other versions. We hypothesized that this would result in outputs corresponding to pitch estimates. We further incorporate knowledge about overtones and noise levels into our system to support this hypothesis and improve results. In our experiments, we showed that our approach outperforms two baselines and that our proposed extensions to the model are effective. Overall, our work demonstrates the use of weak cross-version targets to replace strong pitch annotations.

This paper serves as a proof of concept for our core idea, which could be extended in future work. First, better results may be obtained by utilizing larger model architectures and bigger training datasets than in the present study. Here, we also abstained from excessive model and hyperparameter tweaking. In the future, larger and more extensively tuned models may close the gap between fully supervised approaches and the proposed cross-version training. Second, one may extend our approach to align one input excerpt to multiple versions simultaneously within the same training step (rather than choosing one target version at a time). This may further regularize the model output. Finally, future work may explore more elaborate synthesis models that could replace the simplistic overtone and noise models used here. For example, one may incorporate knowledge about the sound production processes of different instruments into the network [13]. In this context, results might also be improved by estimating the synthesis parameters (e. g., amplitudes of the overtone model) from the input recording, rather than using fixed processing steps.

Acknowledgments: This work was supported by the German Research Foundation (DFG MU 2686/7-2, MU 2686/11-2). The authors are with the International Audio Laboratories Erlangen, a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. The authors gratefully acknowledge the compute resources and support provided by the Erlangen Regional Computing Center (RRZE).

7. REFERENCES

[1] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 475–481.

[2] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.

[3] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference, (ISMIR)*, Paris, France, 2018, pp. 50–57.

[4] K. W. Cheuk, Y. Luo, E. Benetos, and D. Herremans, "Revisiting the onsets and frames model with additive attention," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 2021.

[5] 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.

[6] 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, 2019.

[7] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[8] M. Cuturi and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, 2017, pp. 894–903.

[9] 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.

[10] 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.

[11] M. Krause, C. Weiß, and M. Müller, "Soft dynamic time warping for multi-pitch estimation and beyond," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.

[12] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirovic, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1118–1128, 2020.

[13] T. Berg-Kirkpatrick, J. Andreas, and D. Klein, "Unsupervised transcription of piano music," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, Montréal, Canada, 2014, pp. 1538–1546.

[14] Y. Wu, B. Chen, and L. Su, "Polyphonic music transcription with semantic segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 166–170.

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

[16] I. Hadji, K. G. Derpanis, and A. D. Jepson, "Representation learning via global temporal alignment and cycle-consistency," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual, 2021, pp. 11 068–11 077.

[17] C. Chang, D. Huang, Y. Sui, L. Fei-Fei, and J. C. Niebles, "D3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 3546–3555.

[18] R. Agrawal, D. Wolff, and S. Dixon, "A convolutional-attentional neural framework for structure-aware performance-score synchronization," *IEEE Signal Processing Letters*, vol. 29, pp. 344–348, 2021.

[19] N. Cleju, M. G. Jafari, and M. D. Plumbley, "Analysis-based sparse reconstruction with synthesis-based solvers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 5401–5404.

[20] 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.

[21] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2020.

[22] J. Engel, R. Swavely, L. H. Hantrakul, A. Roberts, and C. Hawthorne, "Self-supervised pitch detection by inverse audio synthesis," in *International Conference on Machine Learning (ICML), Workshop on Self-Supervision in Audio and Speech*, Vienna, Austria, 2020.

[23] 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.

[24] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 tracking in polyphonic music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 63–70.

[25] M. Maghoumi, E. M. Taranta, and J. LaViola, "DeepNAG: Deep non-adversarial gesture generation," in *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, College Station, Texas, USA, 2021, pp. 213–223.

[26] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. Grohganz, "Schubert Winterreise dataset: A multimodal scenario for music analysis," *ACM Journal on Computing and Cultural Heritage (JOCCH)*, vol. 14, no. 2, pp. 25:1–18, 2021.

[27] C. Weiß, H. Schreiber, and M. Müller, "Local key estimation in music recordings: A case study across songs, versions, and annotators," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2919–2932, 2020.

[28] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 341–344.

[29] G. E. Poliner and D. P. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, 2007.