Training Deep Pitch-Class Representations With a Multi-Label CTC Loss

Christof Weiss, Geoffroy Peeters

▶ **To cite this version:**

Christof Weiss, Geoffroy Peeters. Training Deep Pitch-Class Representations With a Multi-Label CTC Loss. International Society for Music Information Retrieval Conference (ISMIR), Nov 2021, Virtual Event, France. hal-03349734

HAL Id: hal-03349734
https://hal.science/hal-03349734

Submitted on 20 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TRAINING DEEP PITCH-CLASS REPRESENTATIONS WITH A MULTI-LABEL CTC LOSS

Christof Weiß, Geoffroy Peeters
LTCI, Télécom Paris, Institut Polytechnique de Paris, France

ABSTRACT

Despite the success of end-to-end approaches, chroma (or pitch-class) features remain a useful mid-level representation of music audio recordings due to their direct interpretability. Since traditional chroma variants obtained with signal processing suffer from timbral artifacts such as overtones or vibrato, they do not directly reflect the pitch classes notated in the score. For this reason, training a chroma representation using deep learning ("deep chroma") has become an interesting strategy. Existing approaches involve the use of supervised learning with strongly aligned labels for which, however, only few datasets are available. Recently, the Connectionist Temporal Classification (CTC) loss, initially proposed for speech, has been adopted to learn monophonic (single-label) pitch-class features using weakly aligned labels based on corresponding score–audio segment pairs. To exploit this strategy for the polyphonic case, we propose the use of a multi-label variant of this CTC loss, the MCTC, and formalize this loss for the pitch-class scenario. Our experiments demonstrate that the weakly aligned approach achieves almost equivalent pitch-class estimates than training with strongly aligned annotations. We then study the sensitivity of our approach to segment duration and mismatch. Finally, we compare the learned features with other pitch-class representations and demonstrate their use for chord and local key recognition on classical music datasets.

1. INTRODUCTION AND RELATED WORK

The Pitch Class Profile (PCP) or chroma is one of the most frequently used audio feature in Music Information Retrieval (MIR). Chroma features are typical for MIR for several reasons: First, they were developed specifically for music [1] as opposed to other features which were inherited from speech processing (such as MFCCs). Second, despite the success of end-to-end-systems, PCP or chroma features are still used today due to their semantic mid-level nature, being musically interpretable as energy distribution over the twelve chromatic pitch classes in an audio signal (see Figure 1). Because of this, PCPs directly relate to musical harmony, therefore being used for chord and key estimation or audio matching (cover song retrieval) tasks.

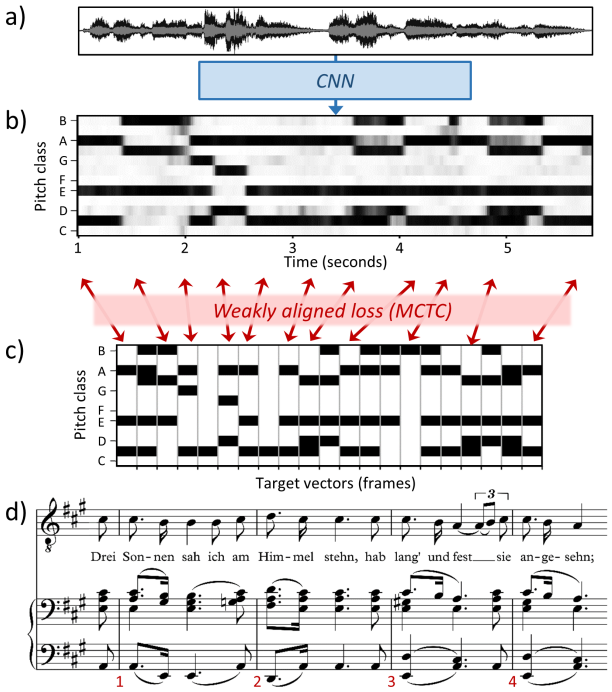

Figure 1. Training a CNN with weakly aligned targets (schematic). Song No. 23 from Schubert's *Winterreise* sung by R. Trekel. (a) Waveform. (b) Pitch-class estimates. (c) Non-aligned Targets derived from the score. (d) Score.

Chroma features based on signal processing. Early approaches [2, 3] to chroma are based on signal processing and map a time–frequency representation such as the Short Time Fourier Transform (STFT) [2] or the Constant-Q-Transform (CQT) [3] to the twelve pitch classes. However, due to timbral characteristics such as overtones (which correspond to different pitch classes), transient note onsets, or vibrato, these chroma features do not directly reflect the pitch classes notated in the score, thus limiting their interpretability. This motivated sophisticated while still hand-crafted features, which aim at reducing the influence of timbre [4–6], at making this influence equal for all instruments [7], or at equalizing loudness variation and transient components [8]. To study the effect of these improvements for chord recognition, Cho et al. [9] present an in-depth comparison, concluding that suitable chroma features largely redeem the benefits of complex chord models.

Pitch-class representations based on deep learning. Recently, deep learning of pitch-class features from data has become a promising direction. Yet, prior works on "deep chroma" have a concrete application task in mind

© C. Weiß and G. Peeters. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C. Weiß and G. Peeters, "Training Deep Pitch-Class Representations With a Multi-Label CTC Loss", in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, Online, 2021.

and do not directly evaluate the obtained pitch-class representations: Humphrey et al. [10] trained a Convolutional Neural Network (CNN) to estimate a Tonnetz representation and use this representation to estimate chords. Later works [11,12] extend this to end-to-end (CQT-based) chord recognition. In a similar fashion, more recent approaches [13, 14] train pitch-class extractors using annotations derived from chord labels. While this led to promising results for chord recognition and other tasks [15], Korzeniowski et al. [13] showed that the learned representations strongly focus on chord-like structures and do not actually represent the pitch classes notated in the score, thus limiting their interpretability and generalization capability. As an alternative, Wu et al. [16] estimate pitch classes by training with audio and annotations synthesized from MIDI files (which are available in large quantity). The trained features are used for chord recognition with good results, improved in a follow-up work [17]. While this strategy is interesting , systems trained on synthetic data show limited generalization to real audio.

Training with aligned scores. To overcome this problem, large amounts of real audio recordings with pitch-class annotations are needed. Annotation can be done with MIDI-fied instruments [18], which led to several transcription datasets such as MAPS [19], SMD [20], or MAESTRO [21]—all limited to the piano. For other instruments, there are only few pitch-annotated datasets such as Bach10 [22], TRIOS [23], or PHENICX-Anechoic [24] (all $\leq$10 pieces), which often involve multi-track recordings to simplify annotation. As an alternative, symbolic scores can be used to semi-automatically generate pitch-class annotations. While such scores are considered only "weak labels" for popular music [25], the correspondence between score and audio is clearly higher for professional recordings of classical music. For exploiting such score–audio pairs (as done for the MusicNet dataset [26]), automated music synchronization technology such as [27] is required in order to generate a so-called *strong alignment*. While this training strategy leads to effective pitch-class representations [28], the necessary synchronization consitutes a costly and challenging pre-processing step.

Motivation of our work. To simplify and improve this procedure, synchronisation between audio and labels can be done either within the *network* using attention models [29] or transformers [30], or within the *loss computation* using the Connectionist Temporal Classification (CTC) loss [31] for sequence-to-sequence training. CTC was successfully applied by Zalkow et al. [32] to train a monophonic deep-chroma representation from weakly aligned data, which they use for cross-modal retrieval. Yet, since CTC applies to single-label outputs, only monophonic pitch-class representations can be trained this way.

Proposal and paper organization. To overcome this limitation, we propose to use a multi-label variant of CTC (denoted MCTC), recently introduced for handwritten text and optical music recognition [33]. Based on our previous work on MCTC for multi-pitch estimation [34], we apply this loss to train a polyphonic pitch-class representations from score–audio pairs of general correspondence (*weak alignment*) without the need for pre-computing *strong alignments*. Using MCTC, we train a network to detect the framewise activity of pitch classes as indicated by the score (*multi-pitch-class estimation*, see Figure 1).

Our main contributions are as follows: First, we reformalize the MCTC loss to be applicable for PCP. Second, we use this loss to train a musically-motivated CNN inspired by [28] for extracting pitch-class representations. Using several public datasets, we perform experiments to analyze their efficacy and robustness against input modifications. Third, we propose a set of performance measures to directly evaluate the PCP quality without a side-task. Fourth, we demonstrate the potential of our MCTC-based features for visualization and for two downstream tasks: local key and chord estimation. We compare our features to several baselines including features trained with strongly aligned scores. All results indicate that MCTC is a promising tool for training efficient pitch-class representations with weakly aligned score–audio pairs.

2. MCTC LOSS FOR PITCH CLASS ESTIMATION

In the following, we describe and formalize the MCTC loss for training deep pitch-class features, closely following the descriptions in [33, 34] for comparability.

CTC. We consider a Neural Network (NN) which maps an input sequence $\boldsymbol{x}:=(\boldsymbol{x}^1,\ldots,\boldsymbol{x}^U)$ to an output sequence $\boldsymbol{y}:=(\boldsymbol{y}^1,\ldots,\boldsymbol{y}^T)$ with U possibly larger than T due to additional context frames. The CTC loss, initially proposed for speech recognition [31], allows to map the output sequence $\boldsymbol{y}$ to a target sequence (or *label*) l of length $S \ll T$, $l:=(l^1,\ldots,l^S)$. l consists of *characters* $l^s \in L$ where L is an *alphabet*. A path π is a possible alignment between the two sequences $\boldsymbol{y}$ and l. To compute the probability of l given $\boldsymbol{x}$, $p(l|\boldsymbol{x})$, we need to consider all possible (valid) paths π between $\boldsymbol{y}$ and l. CTC requires an extra character $blank$ (or "–"), which stands for either no symbol being active or a repetition of the previously active symbol. This results in an extended alphabet $L' = L \cup \{blank\}$. We then define a mapping function $\mathcal{B}: {L'}^T \rightarrow L^{S\leq T}$, which transforms a path $\pi = (\pi^1 \ldots \pi^t \ldots \pi^T) \in {L'}^T$ to a label $l = (l^1,\ldots,l^S) \in L^S$ by removing repeated and then blank symbols. [1] Given a target sequence l, the inverse of $\mathcal{B}$ or *pre-image* $\mathcal{B}^{-1}(l)$ provides the set of all valid paths π that collapse to l. In practice, $\boldsymbol{y}^t$ is the output of a NN at time t with a softmax activation giving the likelihood of each character $k \in L'$ at time t. The probability of a given path π is the product of the relevant probabilities over time: $\prod_{t=1}^{T} y^t_{\pi^t}$. The probability of observing the label l is the sum over all its valid paths $\pi \in \mathcal{B}^{-1}(l)$:

$$p(l|\boldsymbol{x}) = \sum_{\pi \in \mathcal{B}^{-1}(l)} \prod_{t=1}^{T} y^t_{\pi^t} \tag{1}$$

CTC allows to compute this efficiently using dynamic programming and was used for MIR tasks such as lyrics alignment [35] or monophonic pitch-class representations [32].

[1] For example, if $T=5$ and $L=\{\mathrm{a},\ldots,\mathrm{z}\}$, $\mathcal{B}(\mathrm{a}-\mathrm{ab}-) = \mathcal{B}(\mathrm{aa}-\mathrm{ab}) = \mathcal{B}(-\mathrm{a}-\mathrm{ab}) = \mathrm{aab}$.

Multi-label CTC (MCTC). The CTC loss is a useful tool for single-label problems. However, polyphonic pitch-class estimation is a multi-label problem, where an input $\boldsymbol{x}$ needs to be mapped to several (non-mutually exclusive) target labels y_k. An extension of the CTC loss to the multi-label case has been proposed by Wigington et al. [33] for handwritten text recognition (letters with accents) and applied to multi-pitch estimation in our previous work [34].

In the case of PCP, any combination of pitch-classes is allowed to be simultaneously active (see Figure 1). Modeling all combinations as individual symbols would be possible in theory but leads to a large network output ($2^{12} = 4096$), thereby not accounting for the high inter-dependency of similar combinations, and is therefore not adequate for the problem. Following [34], we thus consider the 12 pitch classes $q_i \in \{C, C\sharp \ldots, B\}$ as different *categories* C_i, $i \in \{1, \ldots, 12\}$, each of which comprises the same set of *components*: 0 (absence of this pitch-class), 1 (presence) together with the $blank$ symbol. This leads to the alphabet $\{blank, 0, 1\}$. A *character* k is then the union (a tuple) of components from different categories, i. e. in our case, a multi-hot target vector denoting pitch-class activities $k \in \{0, 1\}^{12}$ denoting the co-occurrence of several pitch classes. The label l is a *sequence of characters* with vocabulary L (all binary pitch-class vectors). It can be decomposed into component-level label sequences l_i with vocabulary L_i. In the same spirit, at the path-level, we can define a character-level path π with vocabulary L' and a component-level path π_i with vocabulary L'_i. We now describe the realization of these sets for different multi-label variants of CTC proposed in [33].

Separable CTC Loss (SCTC). Assuming that pitch-classes occur independently of each other (which is of course a wrong assumption since e. g., the pitch classes of a chord are tied together), there is a trivial approach in which each category C_i is considered by an individual CTC loss. The individual losses are then multiplied:

$$p(l|\boldsymbol{x}) = \prod_{i=1}^{12} \sum_{\pi_i \in \mathcal{B}^{-1}(l_i)} \prod_{t=1}^{T} y^t_{i,\pi^t_i}. \tag{2}$$

For applying SCTC to pitch classes, $C_i = L'_i = \{blank, 0, 1\}$ and $L_i = \{0, 1\}$, resulting in $|C| = 12$ distinct categories. The input to the loss, which is the output of the network, is a tensor $y^t_{i\in\{1\ldots12\},k_i\in\{blank,0,1\}} \in [0, 1]^{12\times3}$ with softmax activation along the second dimension. It represents the probability of observing $blank$, not observing pitch-class q_i, or observing q_i at time t. Treating each pitch class as an independent sequence of components $\in \{blank, 0, 1\}$ makes their alignment difficult (no explicit modelling of pitch-class co-occurence). We thus do not expect SCTC to work well in accordance with [33, 34].

MCTC Loss Without Epsilon (MCTC:NE). For correctly modelling the joint occurrence of pitch classes, we introduce the MCTC loss in its simplest form, the "no epsilon" variant MCTC:NE (details in the following). In this case, we have an individual $blank_i$ symbol for each category so that $C_i = L'_i = \{blank_i, 0, 1\}$. Then, the set of all possible *characters* is $L' = L'_1 \times L'_2 \times \ldots \times L'_{12}$. The overall $blank$ character is the combination of the blank components: $blank_{\mathrm{MCTC}} = (blank_1, \ldots, blank_{12})$. We compute the probability y^t_k of a character k at time t as the product of all its component probabilities $y^t_k = \prod_{i=1}^{12} y^t_{i,k_i}$:

$$p(l|\boldsymbol{x}) = \sum_{\pi \in \mathcal{B}^{-1}(l)} \prod_{t=1}^{T} \prod_{i=1}^{12} y^t_{i,\pi^t_i}. \tag{3}$$

In practice, we do this only for the characters k within the training batch. The input to the loss is a $\boldsymbol{y}^t \in [0, 1]^{12\times3}$ with softmax activation over the second dimension.

MCTC Loss With Epsilon (MCTC:WE). In the previous variant, the network has to simultaneously predict $blank$ for all components individually in order to predict the $blank_{\mathrm{MCTC}}$ character. As proposed in [33], there is a more elegant way of dealing with repetitions of the complete character (pitch-class vector): using an extra category. We therefore define the new category $C_1 = \{blank, not\, blank\}$. The remaining categories $C_2 \ldots C_{13}$ correspond to the 12 pitch classes, as before. To ignore these categories when computing the $blank_{\mathrm{MCTC}}$ probability, we introduce for them an additional ε symbol. This leads to $L'_i = \{0, 1, \varepsilon\}$ for $C_2 \ldots C_{13}$. Please note that ε does not correspond to a network output but is only defined for mathematical convenience. In this scenario, the $blank_{\mathrm{MCTC}}$ *character* is defined as $blank_{\mathrm{MCTC}} = (blank, \varepsilon, \ldots, \varepsilon)$ and all other characters are of the form $k = (not\, blank, 0/1, \ldots, 0/1)$. Using this variant, it is also possible to explicitly model silence using the character $k = (not\, blank, 0, \ldots, 0)$. We therefore compute

$$y^t_k = \begin{cases} y^t_{1,blank} & k = blank_{\mathrm{MCTC}} \\ y^t_{1,not\,blank} \cdot \prod_{i=2}^{13} y^t_{i,k_i} & \text{otherwise.} \end{cases} \tag{4}$$

In this variant, $blank_{\mathrm{MCTC}}$ can be used to repeat the whole character. Its probability is computed ignoring the other categories' probabilities (ε). Here, the input to the loss is a tensor $\boldsymbol{y}^t \in [0, 1]^{13\times2}$ with softmax activation along the second dimension, [2] where the first category corresponds to $blank$ and the other 12 categories to the pitch classes.

Table 1. CNN architecture. Depending on the loss used, we choose $Q \in \{1, 2, 3\}$ and $P \in \{0, 1\}$ appropriately.

Layer	Kernel size	Output shape	# Parameters
Layer norm.		$(T{+}74, 216, 6)$	2592
Conv2D, MaxPool	15×15	$(T{+}74, 216, 20)$	27020
Conv2D, MaxPool	3×3	$(T{+}74, 72, 20)$	3620
Conv2D	75×1	$(T, 72, 10)$	15010
Conv2D	1×1	$(T, 72, 1)$	11
Conv2D	1×61	$(T, 12{+}P, Q)$	$Q(62{+}73{\cdot}P)$
Total			48253 $+Q(62{+}73{\cdot}P)$

3. DEEP-LEARNING METHOD

To investigate the benefit of MCTC for training, we do neither use complex architectures such as CRNNs [36] or U-Nets [37] nor data augmentation strategies [38] but instead

[2] A $73{\times}2$ tensor with sigmoid activation is equivalent; softmax allows for using the numerically stable *logsoftmax* implementation of Pytorch.

Table 2. Overview of the datasets used in this work.

ID	Name	Instrum.	Annot. Strategy	hh:mm
Mae	MAESTRO [21] v3.0.0	Piano	MIDI piano	198:39
B10	Bach10 [22]	Violin, wind	Multitrack	0:06
Tri	TRIOS [23]	Chamber m.	Multitrack	0:03
PhA	PHENICX-Anechoic [24]	Orchestra	Multitrack	0:10
MuN	MusicNet [26]	Chamber m.	Aligned scores	34:08
SWD	Schubert Winterreise [41]	Piano, voice	Aligned scores	10:50

use a simple 5-layer CNN (Table 1). Inspired by [32, 39], we use a Harmonic Constant-Q Transform (HCQT) as input representations, with five harmonics and one subharmonic (six input channels). The audio sample rate is 22050 Hz, the CQT has a hopsize of 512 samples (roughly 43.07 Hz), and three bins per semitone over six octaves ($3 \cdot 6 \cdot 12 = 216$ bins). We use the librosa implementation of the CQT, which includes tuning estimation. [3] The input is a HQCT tensor $\boldsymbol{x}$ of shape $(T+74, 216, 6)$ (with 74 context frames, $U = T+74$) processed with log compression ($\gamma = 10$) and layer normalization [40].

To simulate the weakly aligned target label sequences $l := (l^1, \ldots, l^S)$, we use strongly aligned target vectors $(\boldsymbol{y}^1, \ldots, \boldsymbol{y}^T)$ and remove repeated vectors (see Figure 1c).

Following [28], we use a musically motivated architecture where the first layer performs pre-filtering using small rectangular kernels, followed by binning to the 72 pitches and temporal reduction (removing context frames). Next, we merge the channels with 1×1 convolutions. The final convolution reduces the 72 pitches to 12 pitch classes using a kernel of length 61. The exact output size depends on the loss used and is parameterized by P and Q: For our baseline (strongly aligned targets), we use the same architecture with binary cross entropy (BCE), then $P = 0$, $Q = 1$ and sigmoid activation for the output (see Table 1). For SCTC and MCTC:NE, we use $P = 0$, $Q = 3$ and softmax activation over the last dimension. For MCTC:WE, we need a further output dimension for the $blank_{\mathrm{MCTC}}$, thus using $P = 1$ and $Q = 2$. Resulting from this, our network has roughly 48k parameters, slightly varying according to the loss used. We use LeakyReLU activations, max pooling, stochastic gradient descent with momentum (as in [38]), and learning rate scheduling. For strongly aligned training, we use mini-batches of size 25 and length T=1 or U=75. For MCTC, we use only one example $(\boldsymbol{x}, l)$ per batch but of a considerably higher length (default $T = 500$ frames). [4]

4. DATASETS

For our experiments, we consider several datasets (Table 2) representing the annotation strategies introduced in Section 1. As a MIDI-piano dataset, we consider a subset (1/6) of MAESTRO (Mae). Moreover, we include the multi-pitch datasets Bach10 (B10), TRIOS (Tri), and PHENICX-Anechoic (PhA), whose pitch annotations are reduced to the pitch-class level. Furthermore, we use two datasets based on aligned scores: MusicNet (MuN), which comprises pitch annotations for 330 chamber music recordings, and the Schubert Winterreise Dataset (SWD), which comprises several performances, scores, and annotations of Franz Schubert's song cycle *Winterreise*. For SWD, we use MIDI files and measure annotations together with a synchronization algorithm [27] to generate pitch-class annotations. [5] For MCTC-based training, we do not need these strongly aligned pitch-class annotations but reduce the pre-aligned targets to a sequence of non-repeating vectors (see Figure 1c). For baseline experiments and for evaluation, we use the strongly aligned targets.

[3] https://librosa.org/.

[4] Code: https://github.com/christofw/pitchclass_mctc/.

Table 3. Comparison of MCTC variants, trained on the dataset SWD in a performance (or version) split.

Model/Loss	P	R	F	CS	AP
All-Zero	0	0	0	0.486	0.211
CQT-Chroma	0.512	0.681	0.579	0.701	0.594
CNN – SCTC	**0.850**	0.048	0.090	0.520	0.416
CNN – MCTC:NE	0.747	0.775	0.758	0.802	0.798
CNN – MCTC:WE	0.762	**0.853**	0.802	0.830	0.851
CNN – Strong alignment	**0.850**	0.790	**0.818**	**0.860**	**0.886**

5. EXPERIMENTS

In the following, we present several experiments to assess the effectiveness of MCTC for pitch class estimation.

5.1 Evaluation Measures

All models output frame-wise probabilities $\hat{y}_k^t$ (frame rate 43.07 Hz) for the activity of the twelve pitch classes. To directly measure the similarity between targets $\boldsymbol{y}^t$ and predictions $\hat{\boldsymbol{y}}^t$ without a threshold (continuous-valued), we took inspiration from other music transcription tasks by using their Cosine Similarity (CS) and their Average Precision (AP) [6] as in [38]. We also binarize $\hat{\boldsymbol{y}}^t$ using a threshold of 0.5 (motivated by the sigmoid/softmax outputs) to compute precision (P), recall (R), and F-measure (F).

5.2 Comparing MCTC Variants

First, we run a number of experiments to compare the different variants of MCTC (Table 3). To test generalization to new acoustic conditions, we consider a "version split" of the SWD [42,43] with seven performances used for training and validation and two (HU33, SC06) for testing. To assess the effectiveness of MCTC, we consider several baselines:

All-zero: Since the majority of pitch classes are inactive more often than active, we compute our evaluation measures for an all-zero output. We obtain a cosine similarity of CS=0.486 and an average precision of AP=0.211.

CQT-Chroma: Next, we evaluate a CQT-based chroma as implemented in librosa (1 bin per semitone), followed by max-normalization and thresholding. This already leads to CS=0.701 and AP=0.594 as well as F=0.579.

CNN – Strong alignment: This is our central baseline, relying on the supervised training with pre-aligned annota-

[5] https://zenodo.org/record/5139893/.

[6] Average precision corresponds to the area under the precision–recall curve—a concept similar to Receiver-Operator-Characteristics (ROC).

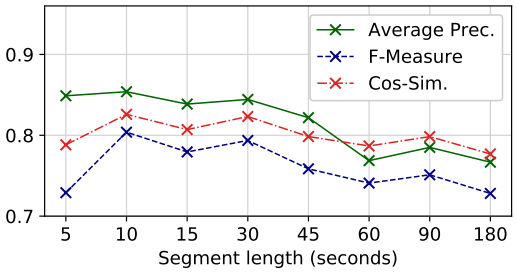

Figure 2. Pitch-class estimation results for different durations of MCTC training segments.

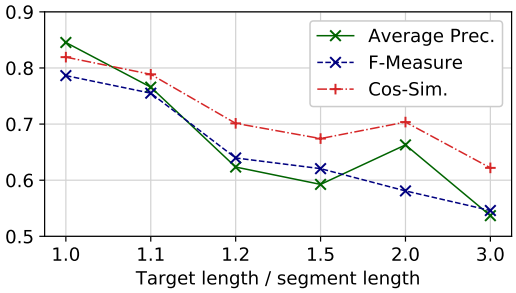

Figure 3. Estimation results for weakly corresponding target segments (until three times the input length).

tions and BCE loss. For this approach, we obtain F=0.818, CS=0.860 and AP=0.886, which are promising results.

Now, we train our CNN described in Section 3 with the different MCTC losses, feeding segments of length $T = 500$ frames (roughly 12 sec) plus context to the network, together with the unaligned pitch activity vectors of the segment as targets l. Similar to [33, 34], SCTC leads to poor performance—in our case with a very low recall and CS=0.520, only slightly above the all-zero baseline, which means that the network mostly predicts zero. For MCTC:NE, the results are better with CS=0.802 and AP=0.798. As in [33, 34], the MCTC:WE variant with an explicit $blank_{\mathrm{MCTC}}$ produces the best results with F=0.802, CS=0.830, and AP=0.851. Though all results with MCTC:WE are below the strongly aligned baseline, the gap between the two approaches is small. We thus consider the MCTC:WE as a promising tool, which only requires weakly aligned data for training (and allows to scale up data more easily). For the following experiments, we only use the MCTC:WE variant (from now on: MCTC). Training time (per epoch) is longer for MCTC (by a factor of roughly 20) compared to strongly aligned training while convergence was faster with MCTC.

5.3 Sensitivity of MCTC-based Training

To investigate the behavior of MCTC-based training, we conduct two further systematic experiments.

Sensitivity to segment duration. First, we test the influence of the input segment length T (previously 500 frames or 12 sec). Since boundaries of input and target segments are musically corresponding, we expect shorter segments to result in a simpler alignment task for the loss, but longer segments to give more freedom for alignment. The results in Figure 2 confirms this assumption—a segment length of 10 sec is beneficial compared to 5 sec, and

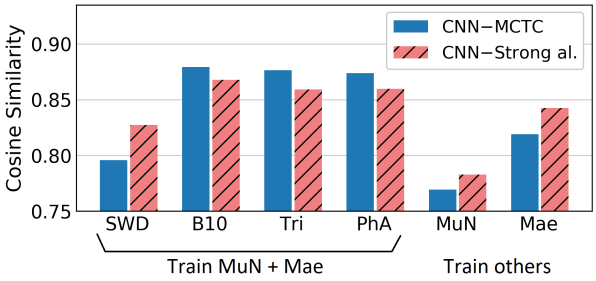

Figure 4. Results for the cross-dataset experiment.

the training behavior is quite stable until 30 sec length. For longer segments, the scores slowly drop. However, even a segment of 3 minutes length leads to a meaningful model, still outperforming the CQT baseline in Table 3. This encouraging result suggests that long score–audio segments of general correspondence can be used with MCTC, i. e. for classical music, short pieces or sections of long ones.

Sensitivity to segment mismatch. Next, we investigate the sensitivity of training when the boundaries of input $\boldsymbol{x}$ and target $\boldsymbol{l}$ segment are not perfectly corresponding (Figure 3). To this end, we use a target segment that corresponds to a longer input segment while the actual input segment is kept at a constant length of $T = 500$ frames. For a factor of 1.1 (target length 550 frames), performance only slightly decreases. In absolute time, this means that the target has one second more "pitch class information" than the network's input. This scenario can be handled successfully by MCTC, which means that "even more weakly" aligned pairs are possible. For longer targets, performance drops—though training does not break down until a target context of three times the input segment length. We conclude that MCTC allows for quite some imprecision in the correspondence of the segment boundaries (up to one second).

5.4 Cross-Datasets Evaluation

Next, we test our MCTC training procedure on all datasets described in Section 4, covering various instrumentations and annotation strategies. Figure 4 shows the corresponding results. For the first four datasets (`SWD`, `B10`, `Tri`, `PhA`), we train on `MuN` and `Mae`. For `SWD`, these cross-dataset results are slightly worse than the results reported for cross-validation in Table 3. Evaluating on `MuN` (training on all others) leads to slighly worse CS; evaluating on `Mae` works better. For the larger datasets (`SWD`, `MuN`, `Mae`), strongly aligned training is superior to MCTC-based training. For the smaller transcription datasets (`B10`, `Tri`, `PhA`), the MCTC-based strategy obtains slightly better results. However, all differences are small. This is encouraging since with MCTC, larger training datasets can be easily achieved so that using larger networks is promising. In preliminary experiments with a larger CNN (more channels, 600k parameters), we already observed improved results.

5.5 Application: Visualization

As said, we aim at training a transcription-like representations capturing the pitch classes as indicated by the score. To illustrate this, we provide a visual example without

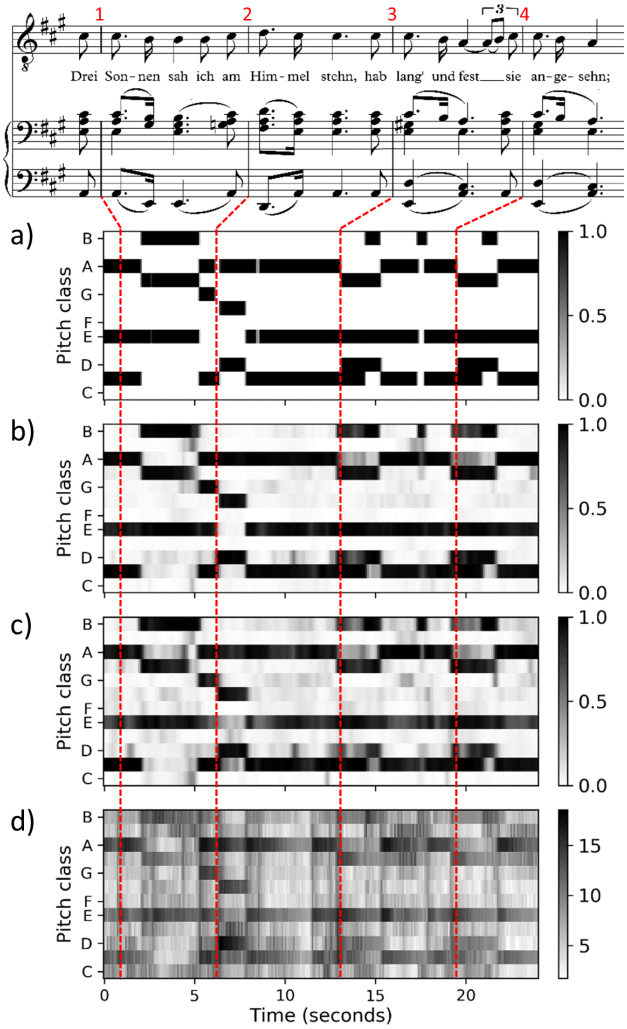

Figure 5. Example pitch-class features for an excerpt of Schubert's *Winterreise* (Song No. 23 sung by R. Trekel). (a) Pitch-class annotations from aligned score. (b) CNN's pitch-class predictions trained with strongly aligned targets and (c) trained with MCTC. (d) CQT chroma features.

thresholding the outputs (Figure 5). Both strongly and weakly (MCTC) aligned training (on other tracks of SWD) lead to visualizations (Figure 5b+c) close to the score-based one (a), with the strongly aligned representation (cosine similarity 0.948 with the score) marginally "cleaner" than the MCTC one (CS=0.930). In contrast, the CQT chroma (Figure 5d) is less clear (CS=0.714) and exhibits the typical artifacts: First, a singer vibrato (e. g., for $C\sharp$ at 10 sec); second, overtones (e. g., B as overtone of E at 9 sec), and third, transient piano onsets (e. g., at 8 sec). All of these artifacts are suppressed by the trained CNNs.

5.6 Application: Chord and Local Key Estimation

Finally, we test the usefulness of our learned features for two harmony analysis tasks: chord recognition (for the 24 major and minor chords) [9] and local key estimation [42] using the respective annotations of SWD [41]. Having a score-like and interpretable feature at hand, we opt for a traditional system based on simple templates (thus allowing for defining chord or key templates only through music theory knowledge) and a Hidden Markov Model (HMM)

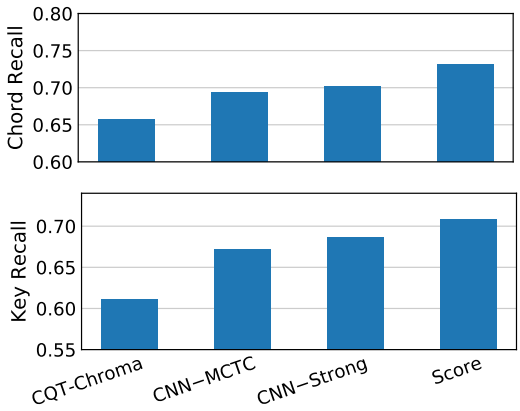

Figure 6. Chord (upper) and local key (lower) estimation results on SWD using different pitch-class features.

for context-sensitive smoothing (with uniform, diagonal-enhanced transition matrix, see [9]). This system does not require any pretraining. For both tasks, we compute the HMMs emission probabilities using the cosine similarity between PCP and templates. We downsample all pitch-class features to roughly 10 Hz.

Chord recognition. For this task, we set the HMM self-transition probability to $a_{i,i} = 0.1$ (i. e. $a_{i,j\neq i} = 0.9/23$ for all other transitions) and use binary chord templates (1 at the triad's pitch classes, 0 otherwise). The results in Figure 6 (upper plot) are promising for such a simple system: The learned features (CNN–MCTC and CNN–Strong) outperform the CQT-based chroma and show a promising improvement towards score-based pitch classes (which are the targets $\boldsymbol{y}^t$ of our CNN training). Also, MCTC-based results are close to strongly aligned ones.

Local key estimation. For this task, we use log compression and median filtering (filter length 10 seconds) for pre-processing the PCPs, together with a higher self-transition probability $a_{i,i} = 0.5$. The key templates are simply based on music-theory (1 for scale pitch classes, 2 for the tonic triad, 0 otherwise). Again, learned features (CNN-MCTC and CNN-Strong) outperform CQT, now almost closing the gap towards the score-based features, and MCTC-based results are close to strongly aligned ones.

While these results do not reach the state of the art for both tasks (e. g. [14, 42]), they are promising for a purely hand-crafted system. Most remarkably, this strategy allows for an "objective" analysis since the systems' parameters are specified in an explicit, musically motivated way.

6. CONCLUSION

In this paper, we presented a novel strategy for training pitch-class representations with weakly aligned score–audio pairs. To this end, we adapted a multi-label CTC loss, which led to a successful training close to the training with strongly aligned scores. Though being computationally more expensive, MCTC-based feature learning is a very promising direction since weakly aligned annotations for long segments of music can be created with much less effort, thus enabling an easier scalability to larger datasets, which allows for training more complex networks.

7. ACKNOWLEDGEMENTS

C. W. is funded by a research fellowship of the German Research Foundation (DFG WE 6611/1-1). We thank Curtis Wigington for advice on implementation and Meinard Müller and Frank Zalkow for fruitful discussions.

8. REFERENCES

[1] M. A. Bartsch and G. H. Wakefield, "To catch a chorus: Using chroma-based representations for audio thumbnailing," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2001, pp. 15–18.

[2] T. Fujishima, "Realtime chord recognition of musical sound: A system using common lisp music," in *Proc. Int. Computer Music Conf. (ICMC)*, Beijing, China, 1999, pp. 464–467.

[3] G. H. Wakefield, "Mathematical representation of joint time-chroma distributions," in *Proc. SPIE Conf. on Advanced Signal Processing Algorithms, Architecture and Implementations*, Denver, USA, 1999, pp. 637–645.

[4] E. Gómez, "Tonal description of music audio signals," PhD Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[5] K. Lee, "Automatic chord recognition from audio using enhanced pitch class profile," in *Proc. Int. Computer Music Conf. (ICMC)*, New Orleans, USA, 2006, pp. 306–311.

[6] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proc. Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 135–140.

[7] M. Müller, S. Ewert, and S. Kreuzer, "Making chroma features more robust to timbre changes," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[8] M. Müller, F. Kurth, and M. Clausen, "Chroma-based statistical audio features for audio matching," in *Proc. IEEE Workshop on Applications of Signal Processing (WASPAA)*, New Paltz, USA, 2005, pp. 275–278.

[9] T. Cho and J. P. Bello, "On the relative importance of individual components of chord recognition systems," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 477–492, 2014.

[10] E. J. Humphrey, T. Cho, and J. P. Bello, "Learning a robust tonnetz-space transform for automatic chord recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 453–456.

[11] E. J. Humphrey and J. P. Bello, "Rethinking automatic chord recognition with convolutional neural networks," in *Proc. IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, Boca Raton, USA, 2012, pp. 357–362.

[12] ——, "From music audio to chord tablature: Teaching deep convolutional networks to play guitar," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 6974–6978.

[13] F. Korzeniowski and G. Widmer, "Feature learning for chord recognition: The deep chroma extractor," in *Proc. Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, New York City, USA, 2016, pp. 37–43.

[14] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition," in *Proc. Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Suzhou, China, 2017, pp. 188–194.

[15] G. Doras, F. Yesiler, J. Serrà, E. Gómez, and G. Peeters, "Combining musical features for cover detection," in *Proc. Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Montréal, Canada, 2020, pp. 279–286.

[16] Y. Wu and W. Li, "Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF sequence decoding model," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 27, no. 2, pp. 355–366, 2019.

[17] Y. Wu, T. Carsault, and K. Yoshii, "Automatic chord estimation based on a frame-wise convolutional recurrent neural network with non-aligned annotations," in *Proc. European Signal Processing Conf. (EUSIPCO)*, A Coruña, Spain, 2019, pp. 1–5.

[18] L. Su and Y. Yang, "Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription," in *Proc. 11th Int. Symposium on Computer Music Multidisciplinary Research (CMMR)*, ser. Lecture Notes in Computer Science, Plymouth, UK, 2015, pp. 309–321.

[19] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[20] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland music data (SMD)," in *Demos and Late Breaking News of the Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Miami, USA, 2011.

[21] 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 *Proc. Int. Conf.*

on Learning Representations (ICLR), New Orleans, USA, 2019.

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

[23] J. Fritsch and M. D. Plumbley, "Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, 2013, pp. 888–891.

[24] M. Miron, J. Carabias-Orti, J. Bosch, E. Gómez, and J. Janer, "Score-informed source separation for multichannel orchestral recordings," *Journal of Electrical and Computer Engineering*, vol. 2016, 2016.

[25] 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.

[26] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *Proc. Int. Conf. on Learning Representations (ICLR)*, Toulon, France, 2017.

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

[28] C. Weiß, J. Zeitler, T. Zunner, F. Schuberth, and M. Müller, "Learning pitch-class representations from score–audio pairs of classical music," in *Proc. Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Online, 2021.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. on Learning Representations (ICLR)*, San Diego, USA, 2015.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conf. on Neural Information Processing Systems (NeurIPS)*, Long Beach, USA, 2017.

[31] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. on Machine Learning (ICML)*, Pittsburgh, USA, 2006, pp. 369–376.

[32] F. Zalkow and M. Müller, "Using weakly aligned score–audio pairs to train deep chroma models for cross-modal music retrieval," in *Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Montréal, Canada, 2020, pp. 184–191.

[33] C. Wigington, B. L. Price, and S. Cohen, "Multi-label connectionist temporal classification," in *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, Sydney, Australia, 2019, pp. 979–986.

[34] C. Weiß and G. Peeters, "Learning multi-pitch estimation from weakly aligned score–audio pairs using a multi-label CTC loss," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2021.

[35] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 181–185.

[36] 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 *Proc. Int. Soc. for Music Information Retrieval Conf., (ISMIR)*, Paris, France, 2018, pp. 50–57.

[37] J. Abeßer and M. Müller, "Jazz bass transcription using a U-net architecture," *Electronics*, vol. 10, no. 6, pp. 670:1–11, 2021.

[38] J. Thickstun, Z. Harchaoui, D. P. Foster, and S. M. Kakade, "Invariances and data augmentation for supervised music transcription," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 2241–2245.

[39] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 tracking in polyphonic music," in *Proc. Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Suzhou, China, 2017, pp. 63–70.

[40] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[41] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. G. 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. [Online]. Available: https://doi.org/10.5281/zenodo.5139893

[42] H. Schreiber, C. Weiß, and M. Müller, "Local key estimation in classical music recordings: A cross-version study on Schubert's Winterreise," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 501–505.

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