STABILIZING TRAINING WITH SOFT DYNAMIC TIME WARPING: A CASE STUDY FOR PITCH CLASS ESTIMATION WITH WEAKLY ALIGNED TARGETS

Johannes Zeitler **Simon Deniffel** **Michael Krause** **Meinard Müller**
International Audio Laboratories Erlangen, Germany
{johannes.zeitler, michael.krause, meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Soft dynamic time warping (SDTW) is a differentiable loss function that allows for training neural networks from weakly aligned data. Typically, SDTW is used to iteratively compute and refine soft alignments that compensate for temporal deviations between the training data and its weakly annotated targets. One major problem is that a mismatch between the estimated soft alignments and the reference alignments in the early training stage leads to incorrect parameter updates, making the overall training procedure unstable. In this paper, we investigate such stability issues by considering the task of pitch class estimation from music recordings as an illustrative case study. In particular, we introduce and discuss three conceptually different strategies (a hyperparameter scheduling, a diagonal prior, and a sequence unfolding strategy) with the objective of stabilizing intermediate soft alignment results. Finally, we report on experiments that demonstrate the effectiveness of the strategies and discuss efficiency and implementation issues.

1. INTRODUCTION AND RELATED WORK

Deep neural networks (DNNs) have been commonly used in many music information retrieval (MIR) tasks, such as music transcription [1], or pitch class estimation (PCE) [2, 3]. The latter provides a widely-used feature representation for various subsequent processing pipelines, e.g., audio thumbnailing [4], or chord recognition [3]. Deep learning-based feature extractors yield the highest prediction accuracy when trained on data from the same distribution, which is, however, often not readily available. Thus, one major challenge is the acquisition of a sufficient amount of correctly labeled training data. In classical music, it is often difficult to automatically annotate strongly aligned targets (short: *strong* targets), i.e., with frame-wise target labels, due to changes of tempo. On the other hand, weakly aligned targets (short: *weak* targets) only globally correspond to the input without containing frame-wise local alignments [5,6]. These weak targets are relatively easy to obtain, e.g., by only annotating start and end of an audio segment and deriving targets from the musical score. In our definition of weak targets, the order of the target vectors is correct, but their duration is unknown. Using weak targets in DNN training requires a loss function that aligns network predictions with the corresponding weak targets.

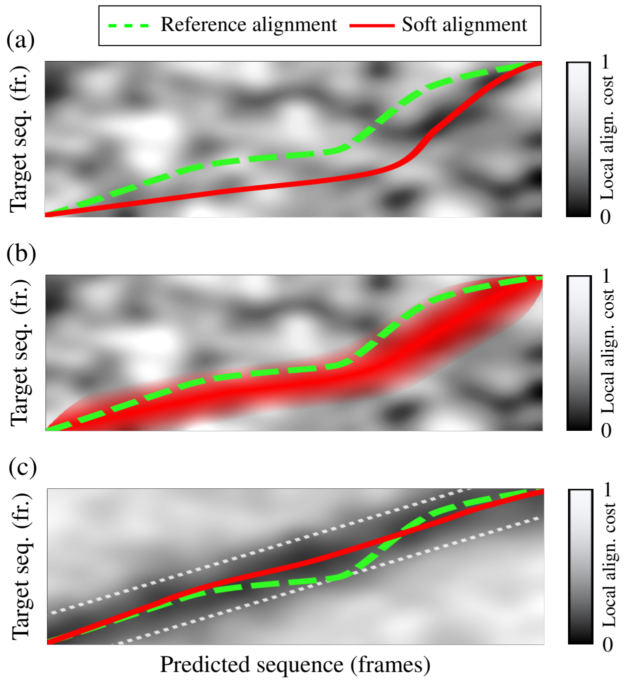

Figure 1: Deviation of strong reference alignments (dashed green) and soft alignments (red) and stabilizing strategies. **(a)** Alignment mismatch of standard SDTW. Stabilizing alignments with **(b)** hyperparameter scheduling and **(c)** diagonal prior.

In classification tasks, one widely used technique for training DNNs with weakly aligned targets is the connectionist temporal classification (CTC) loss [7], which aligns network predictions with a sequence of discrete labels. Despite being extendable to multi-label problems such as multi-pitch estimation (MPE) [8], CTC remains limited to discrete targets and is algorithmically complex.

In contrast to CTC, dynamic time warping (DTW) can be used to measure similarity between two real-valued sequences and has been successfully applied in, e.g., music synchronization and structure analysis [9]. Recently, differentiable approximations of the minimum function [10–12] have been included in DTW, enabling the usage of the DTW principle in gradient-based optimization algorithms. The algorithm proposed in [10], soft dynamic time warping (SDTW), uses so-called *soft alignments* to compute a differentiable cost measure between sequences of different length. In [13], SDTW is used in the context of performance-score synchronization and [6] employed SDTW as a loss function to train DNNs for MPE with weakly aligned pitch annotations. Experiments in [6] indicated training instabilities with SDTW when the sequence lengths of inputs and targets are significantly different. This poses a severe problem in many MIR tasks, where sequences of input audio are typically very long, while weakly labeled targets, i.e., without note durations, are significantly shorter.

In this paper, we investigate the cause of training instabilities under the SDTW loss and show that it is due to a mismatch between the estimated soft alignment and the reference alignment (see Figure 1a) in the early stages of training. This mismatch causes incorrect parameter updates and the training may diverge. Therefore, we introduce and investigate strategies to decrease this alignment error to stabilize training. In particular, we analyze a hyperparameter scheduling strategy to yield smooth alignments in the early training phase (see Figure 1b) as well as the strategy of adding a diagonal prior to the SDTW cost matrix to initially favour diagonal alignments (see Figure 1c). Furthermore, we investigate a sequence unfolding approach, where we uniformly stretch the weak target sequence to the length of the input sequence as proposed in [6]. We choose DNN-based PCE as an exemplary task to study the training process of standard SDTW and the impact of our stabilizing strategies. We demonstrate that the hyperparameter scheduling and the diagonal prior strategies reliably reduce label mismatch in the early training stage and therefore lead to successful trainings. In addition, these two strategies are computationally efficient and require only small modifications to the standard SDTW algorithm.

The remainder of this article is structured as follows. First, in Section 2, we discuss the SDTW loss function and define the concept of soft alignments. Next, in Section 3, we introduce three conceptually different strategies for stabilizing DNN training under SDTW loss. After describing the experimental setup in Section 4, we evaluate cause and effect of training problems with SDTW in Section 5, along with the impact of our stabilizing strategies. Finally, we conclude with Section 6 and give an outlook to potential areas of future research regarding SDTW-based training in MIR.

2. INTRODUCTION TO SDTW

In this section, we introduce SDTW as a loss function in a DNN training framework and define the concept of soft alignments, closely following [10, 14].

2.1 Definition

Let $X = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}\}$ denote a sequence of DNN predictions, $Y = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{M-1}\}$ denote a sequence of weak targets and $Y^{\mathrm{S}} = \left\{\mathbf{y}^{\mathrm{S}}_0, \mathbf{y}^{\mathrm{S}}_1, \ldots, \mathbf{y}^{\mathrm{S}}_{N-1}\right\}$ denote a sequence of strong targets, where $\mathbf{x}_n, \mathbf{y}_m, \mathbf{y}^{\mathrm{S}}_n \in \mathbb{R}^D$ for $n \in \{0, 1, \ldots, N-1\}$ and $m \in \{0, 1, \ldots, M-1\}$. Without loss of generality, we assume $N \geq M$.

Using the mean squared error (MSE) as a local cost function, the elements of the cost matrix $\mathbf{C} := \mathbf{C}_{X,Y} \in \mathbb{R}^{N \times M}$ are computed as

$$\mathbf{C}_{X,Y}(n, m) = \|\mathbf{x}_n - \mathbf{y}_m\|_2^2 . \tag{1}$$

We next define binary alignment matrices $\mathbf{A} \in \{0,1\}^{N \times M}$ which align two sequences of length N and M. Each matrix $\mathbf{A}$ encodes an alignment via a path of ones from cell $(0, 0)$ to $(N-1, M-1)$ using only vertical, horizontal, and diagonal unit steps [10]. All cells not corresponding to the alignment are set to zero. The set of all binary alignment matrices for sequences of length N and M is denoted $\mathcal{A}_{N,M}$. Using a differentiable approximation of the minimum function

$$\mathrm{softmin}^{\gamma}(\mathcal{S}) = -\gamma \log \sum_{s \in \mathcal{S}} \exp(-s/\gamma) \tag{2}$$

for a given finite set $\mathcal{S} \subset \mathbb{R}$ and a hyperparameter $\gamma \in \mathbb{R}$, the SDTW cost is given by

$$\mathrm{SDTW}^{\gamma}_{\mathbf{C}} = \mathrm{softmin}^{\gamma}\left(\{\langle \mathbf{A}, \mathbf{C}\rangle,\ \mathbf{A} \in \mathcal{A}_{N,M}\}\right) \tag{3}$$

and can be computed efficiently via dynamic programming [10]. The inner product $\langle \mathbf{A}, \mathbf{C}\rangle$ is the sum of all elements of $\mathbf{C}$ along the alignment given by $\mathbf{A}$.

2.2 Soft Alignments

The expectation over all alignments $\mathbf{A}$ for a cost matrix $\mathbf{C}$ is captured by the soft alignment matrix [14]

$$\mathbf{E}^{\gamma}_{\mathbf{C}} = \sum_{\mathbf{A} \in \mathcal{A}_{N,M}} p^{\gamma}_{\mathbf{A},\mathbf{C}} \mathbf{A} \quad \in \mathbb{R}^{N \times M} , \tag{4}$$

where the probability of an alignment is defined as

$$p^{\gamma}_{\mathbf{A},\mathbf{C}} = \frac{\exp(-\langle \mathbf{A}, \mathbf{C}\rangle/\gamma)}{\sum_{\mathbf{A}' \in \mathcal{A}_{N,M}} \exp(-\langle \mathbf{A}', \mathbf{C}\rangle/\gamma)} . \tag{5}$$

The soft alignment matrix is of particular interest as it is the the gradient of the SDTW cost w.r.t. the local cost matrix

$$\nabla_{\mathbf{C}} \mathrm{SDTW}^{\gamma}_{\mathbf{C}} = \mathbf{E}^{\gamma}_{\mathbf{C}} \tag{6}$$

and is computed during the backward pass of an SDTW training step with a dynamic programming algorithm [10, 14]. In contrast to the *binary* alignments $\mathbf{A}$, the entries of the soft alignment matrix $\mathbf{E}^{\gamma}_{\mathbf{C}}(n, m)$ can be interpreted as the *probability* of an alignment path going through cell (n, m). Only if this soft alignment assigns probability mass to the correct alignments (n, m), the local cost terms (1) between the correct pairs of predictions $\mathbf{x}_n$ and targets $\mathbf{y}_m$ constitute the overall SDTW cost and the DNN parameters can be successfully trained.

The hyperparameter γ, also termed temperature, controls the smoothness of the softmin function (2). Larger values of γ lead to smooth minima in (3), i.e., with contributions of multiple alignments $\mathbf{A}$, and therefore a "blurry" soft alignment matrix $\mathbf{E}_{\mathbf{C}}^{\gamma}$ (see Figure 1b). On the other hand, small values of γ promote "sharp" soft alignments $\mathbf{E}_{\mathbf{C}}^{\gamma}$ with fewer non-zero entries (see Figure 1a), as (2) converges to the hard minimum function in the limit $\gamma \to 0$ and a single binary alignment $\mathbf{A}$ becomes dominant in (3) and (4).

3. STABILIZING TRAINING WITH SDTW

In this section, we introduce three strategies for stabilizing SDTW-based training: hyperparameter scheduling, diagonal prior, and sequence unfolding.

3.1 Hyperparameter Scheduling

As described in Section 2, the softmin temperature parameter γ controls the smoothness of the SDTW soft alignments. While a low value of γ is desirable to ensure exact correspondences between predictions and targets due to sharp alignments, the latter are problematic in the initial training phase as inaccurate predictions from randomly initialized network parameters lead to erroneous alignments, thus hampering convergence. Therefore, as a first strategy to stabilize SDTW training, we discuss an epoch-dependent scheduling of γ. Starting a training with a large softmin temperature $\gamma^{\text{start}} = 10$ makes the soft alignment fuzzier, which leads to coarse, yet mostly meaningful target assignments (see Figure 1b). After ten epochs with $\gamma = 10$, when the trained network predicts meaningful features, we linearly reduce γ during the following ten epochs to a final value of $\gamma^{\text{final}} = 0.1$, which stays constant for the remaining training.

3.2 Diagonal Prior

On average, the correct alignment of two sequences with arbitrary symbol durations has a higher probability to be close to the diagonal than to deviate from it. Therefore, as a second approach to stabilize the initial training phase, we investigate an additive prior $\mathbf{P} \in \mathbb{R}^{N \times M}$ which penalizes elements of the cost matrix $\mathbf{C}$ that are far from the diagonal (see Figure 2 for an illustration of a prior matrix). A similar strategy was employed in [15] for restricting speech-text alignments to the diagonal. Assuming equal symbol durations, the diagonal alignment of a target $\mathbf{y}_m$ starts at input frame $q_m = \lfloor \frac{Nm}{M} \rfloor$ and ends at $q_{m+1} - 1$. To yield no penalty along the diagonal and a smoothly increasing penalty for distant alignments, we define the elements of the prior matrix as

$$\mathbf{P}(n,m) = 1 - \begin{cases} 1, & q_m \leq n < q_{m+1} \\ \exp\left(\frac{(n-q_m)^2}{-2\nu}\right), & n < q_m \\ \exp\left(\frac{(n-q_{m+1})^2}{-2\nu}\right), & n \geq q_{m+1}\,, \end{cases} \tag{7}$$

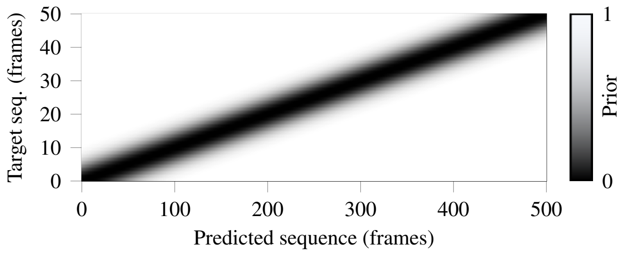

Figure 2: Diagonal prior matrix $\mathbf{P}$ for $N = 500$, $M = 50$ and $\nu = 1000$.

where the parameter ν controls the sharpness of the prior. In our experiments, we use $\nu = 1000$. Finally, the prior matrix is added to the cost matrix with a weight ω to obtain the penalized cost matrix

$$\mathbf{C_P} := \mathbf{C} + \omega \mathbf{P}\,, \tag{8}$$

which replaces $\mathbf{C}$ in (3) to (6). Similarly to the hyperparameter scheduling strategy, we choose a constant prior weight $\omega = 3$ during the first five epochs and then linearly reduce it to $\omega = 0$ during the following five epochs.

Note that the numerical parameters for the strategies presented in Sections 3.1 and 3.2 were determined empirically by the authors and small changes did not affect the training performance. However, when training on sequences of different length, with a different learning rate, or other DNN types, parameters should be adjusted on a validation set. As presented in Section 5, analysis of the soft alignment matrix $\mathbf{E}_{\mathbf{C}}^{\gamma}$ provides a good indication of the current alignment stability.

3.3 Sequence Unfolding

Based on the observation that equal sequence lengths stabilize SDTW training, a third strategy is to uniformly unfold the target sequence (see also [6]). The unfolded target sequence $Y^{\text{U}} = \left\{\mathbf{y}_0^{\text{U}}, \mathbf{y}_1^{\text{U}}, \ldots, \mathbf{y}_{N-1}^{\text{U}}\right\}$ is constructed by uniformly repeating elements from the weakly aligned target sequence, i.e., setting

$$\mathbf{y}_n^{\text{U}} \leftarrow \mathbf{y}_{\lfloor \frac{Mn}{N} \rfloor} \tag{9}$$

to yield equal sequence lengths of the predictions X and the targets Y^{U}. Note that the repetition of target vectors introduces ambiguities, leading to multiple optimum alignments.

4. EXPERIMENTAL SETUP

In this section, we describe the task for our case study, the employed dataset, as well as the used DNN architecture and the training procedure.

4.1 PCE Task

We choose PCE from music recordings as an illustrative case study to investigate the problems of the SDTW loss function and the effect of the stabilizing strategies. In our experimental setting, a DNN takes N frames of input audio (including context) and, for all frames, predicts twelve-dimensional pitch class activation vectors X (see Figure 3

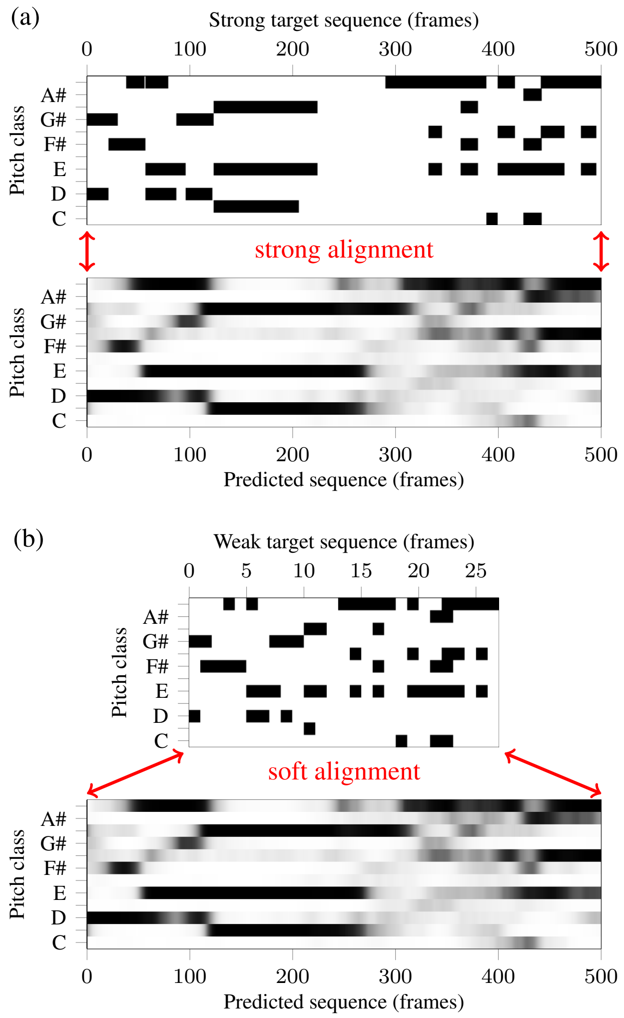

Figure 3: Alignment between training targets and predicted pitch class features X for the running example from *Frühlingstraum*. **(a)** Strong reference alignment for MSE loss with strong targets Y^{S}. **(b)** Soft alignment for SDTW loss with weak targets Y.

for an illustration of predicted pitch class features). We want to train the DNN such that the predictions X match the training targets as close as possible. In the case of strong targets Y^{S}, each predicted frame $\mathbf{x}_n$ is assigned to exactly one target frame $\mathbf{y}_n^{\mathrm{S}}$ using a strong alignment (see Figure 3a). When using weak targets Y, SDTW internally computes a soft alignment based on the cost matrix $\mathbf{C}_{X,Y}$ to assign predictions and targets (see Figure 3b).

4.2 Dataset

Throughout all experiments, we use the Schubert Winterreise dataset (SWD) [16] which contains audio recordings and strongly aligned pitch class annotations. Winterreise is a song cycle for piano and singer, consisting of 24 songs. For each song, SWD comprises nine different performances, resulting in $9 \cdot 24$ recorded songs with a total duration of 10 h 50 min. We split the dataset for training, validation, and testing using a performance split [16]. The publicly available performances by Huesch (HU33, recorded in 1933) and Scarlata (SC06, recorded in 2006)

Layer	Kernel Size	Stride	Output Shape
	Prefiltering		
LayerNorm			$(N+74, 216, 5)$
Conv2D	15×15	(1,1)	$(N+74, 216, 20)$
MaxPool	3×1	(1,1)	$(N+74, 216, 20)$
Dropout			
	Binning to MIDI pitches		
Conv2D	3×3	(1,3)	$(N+74, 72, 20)$
MaxPool	13×1	(1,1)	$(N+74, 72, 20)$
Dropout			
	Time reduction		
Conv2D	75×1	(1,1)	$(N, 72, 10)$
Dropout			
	Chroma reduction		
Conv2D	1×1	(1,1)	$(N, 72, 1)$
Dropout			
Conv2D	1×61	(1,12)	$(N, 12, 1)$

Table 1: Musically motivated CNN architecture [3, 5].

were annotated manually [16] and constitute the test set. For training and evaluation we choose sequences of length $N = 500$, corresponding to approximately 8.7 s of audio at a sampling rate of 22 050 Hz and a hop length of 384 samples. In order to generate weak training targets Y from SWD (which provides strongly aligned pitch class annotations Y^{S}, see Figure 3a), we remove all adjacent repetitions of a pitch class vector (see Figure 3b) [5]. We choose an excerpt from the song *Frühlingstraum*, performed by Randall Scarlata (SC06), as a running example (see Figure 3) to visualize the soft alignment matrices (see Figure 4).

4.3 DNN Architecture and Training

We adapt a conceptually simple and musically motivated five-layer convolutional neural network (CNN) from [3, 5] with 43383 trainable parameters to predict twelve-dimensional pitch class activation vectors from an input sequence. Table 1 provides an overview of the architecture. We choose the harmonic constant-Q transform (HCQT) [17] with five harmonics as an audio feature representation, spanning six octaves at a resolution of three bins per semitone (resulting in 216 frequency bins starting from C1), a hop length of 384 samples and a frame rate of 57.4 Hz. From an input sequence of length $N + 74$, the CNN sequentially predicts N vectors of pitch class activations. For the prediction of one frame, the CNN's receptive field covers 37 adjacent context frames on each side. Leaky ReLU with a negative slope of 0.3 is used as a non-linearity after all hidden convolutional layers and sigmoid activation is used after the final layer. The dropout rate is set to 0.2. All models are trained using the Adam optimizer [18] with a batch size of 32 and an initial learning rate of 0.001. We reduce the learning rate by a factor of two if the validation loss did not decrease during the last four epochs, and terminate the training if the validation loss did not decrease during the last twelve epochs. At the end of training, the model from the epoch with the lowest validation loss is restored. The source code for reproducing our experiments, as well as the trained models are available on `github.com/groupmm/stabilizing_sdtw`.

Loss	Targets	γ	Strategy	F-measure mean	F-measure std
MSE	strong	-	-	**0.82**	0.07
SDTW	weak	0.1	-	0.56	0.37
SDTW	weak	0.3	-	0.63	0.32
SDTW	weak	1.0	-	0.24	0.36
SDTW	weak	3.0	-	0.31	0.38
SDTW	weak	10.0	-	0.57	0.37
SDTW	weak	$10 \rightarrow 0.1$	hyp. sched.	0.80	0.04
SDTW	weak	0.1	diag. prior	**0.81**	**0.02**
SDTW	weak	0.1	seq. unfold.	0.53	0.04

Table 2: Averaged test results for DNNs trained on strongly aligned reference targets as well as DNNs trained with SDTW on weakly aligned targets using either the standard configuration or the discussed stabilization strategies. We report the mean (higher is better) and standard deviation (lower is better) of the F-measure.

5. EVALUATION

In this section, we investigate the training process as well as the prediction accuracy under the standard SDTW loss, and compare it to the discussed stabilizing strategies. For quantitative evaluation, we repeat all DNN trainings ten times from random initializations. For the test set predictions of each trained model, we compute the F-measure w.r.t. time-pitch class bins using a threshold of 0.5. The mean and standard deviation of the F-measures from all trained models are displayed in Table 2.

5.1 Baseline: Strongly Aligned Targets

As a first baseline and an upper bound for all following experiments, we consider DNN training with strongly aligned targets Y^{S}. For the sequence lengths $M = N = 500$ and an MSE loss function, the networks achieve the overall highest mean F-measure of 0.82 with a standard deviation of 0.07 on the test set.

5.2 Standard SDTW

We next analyze DNN training with weak targets Y and the unmodified SDTW formulation from [10, 19] as a loss function. We investigate five different values of $\gamma \in \{0.1, \ldots, 10\}$ which we keep constant during training. Analyzing the mean F-measure on the test set in Table 2, the five variants with standard SDTW yield comparably low results between 0.24 and 0.57, and high standard deviations between 0.32 and 0.38. Between 20 % ($\gamma = 0.3$) and 70 % ($\gamma = 1.0$) of all training runs converged to the all-zero output, indicating a highly unstable training process of standard SDTW. In order to determine the cause of these instabilities, we analyze the quality of automatically generated soft alignments in the SDTW algorithm by visualizing the soft alignment matrix for the running example after training epochs one and 25, respectively. To highlight the effects of small and large values of γ, we focus on the edge cases $\gamma \in \{0.1, 10.0\}$. For $\gamma = 0.1$, the estimated soft alignment exhibits a sharp structure (see Figure 4a), which, after a collapse to a single target frame

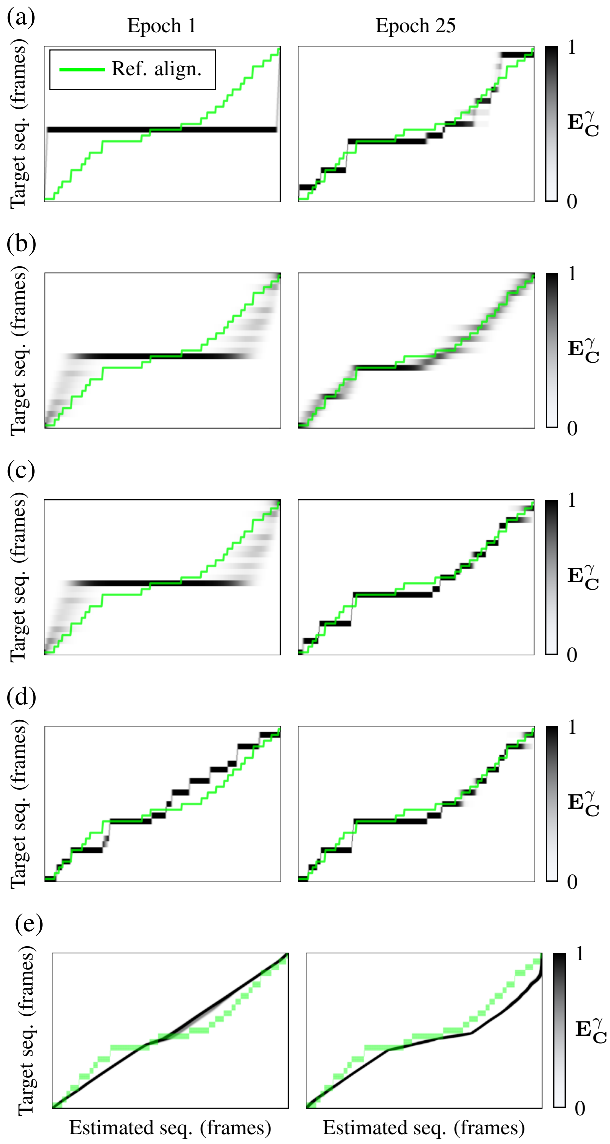

Figure 4: Reference alignment (green) and soft alignment matrix $\mathbf{E}_{\mathbf{C}}^{\gamma}$ (gray/black) for the running example after training epoch 1 (left) and epoch 25 (right) for different training strategies. **(a)** $\gamma = 0.1$, **(b)** $\gamma = 10$, **(c)** hyperparameter scheduling, **(d)** diagonal prior, **(e)** sequence unfolding.

at epoch one, still only marginally overlaps with the reference alignment after 25 epochs. This sharp and erroneous soft alignment causes unstable gradient updates and leads to the collapse of many training runs. When choosing a large softmin temperature $\gamma = 10$, SDTW yields "blurry" soft alignments (see Figure 4b) which at least partially capture the actual target frames in early epochs and coincide well with the reference alignments as training progresses. However, a blurry soft alignment also leads to blurry network predictions as multiple target frames are aligned to each predicted frame, thus resulting in a low F-measure when compared to strongly aligned targets..

5.3 Stabilizing Strategies

After evaluating the unsatisfactory training behavior of standard SDTW, we investigate the effect of the previously introduced training strategies in the following section. We empirically choose $\gamma = 0.1$ as the final softmin temperature in all following experiments, as sharp alignments are necessary for training an estimator with frame-wise precision.

5.3.1 Hyperparameter Scheduling

First, we combine the advantages of high and low values of γ in a hyperparameter scheduling strategy. Starting a training with $\gamma = 10$, the soft alignment matrix for our running example after one epoch is blurry and at least partially overlapping with the reference alignment (see Figure 4c). The successive reduction to $\gamma = 0.1$ until epoch 20 permits sharp alignments at a later training stage. Indeed, Figure 4c shows a soft alignment after epoch 25 which is sharp and coincides well with the reference. The mean F-measure (0.80) in Table 2, as well as the standard deviation (0.04), are the second best of all SDTW-based trainings. However, as the softmin function in (2) is a lower bound for the minimum function [12] which becomes tight for $\gamma \to 0$, the SDTW loss is increasing when decreasing γ, despite unchanged network parameters. Therefore, this strategy does not allow for loss-based learning rate scheduling and early stopping before γ is set to its final value.

5.3.2 Diagonal Prior

The second strategy stabilizes SDTW trainings with low values of γ by adding a penalty cost to off-diagonal elements of the cost matrix. For our running example in Figure 4d, the soft alignment is indeed close to the diagonal after the first training epoch. As, on average, the alignments are diagonal, this often leads to correct assignments of predictions and targets even for randomly initialized DNNs. When the prior weight ω is reduced to zero after the initial training phase, the network is still able to adapt to off-diagonal alignments, as seen in our running example in Figure 4d. Analyzing the performance metrics in Table 2, using a diagonal prior yields the highest mean F-measure (0.81) and the lowest standard deviation (0.02) of all SDTW variants, almost reaching the mean F-measure of the baseline experiments with strong targets and element-wise MSE loss. Moreover, when the prior weight ω is reduced during training, the loss also decreases and therefore learning rate scheduling and early stopping are possible from the beginning.

5.3.3 Sequence Unfolding

Last, we investigate the strategy of unfolding the weak target sequence to the length of the input, which was employed in [6]. For this strategy, we observe fully diagonal soft alignments in the initial training phase, as visualized for our running example in Figure 4e. This is caused by the equal length of the predicted and the target sequence, which can be aligned using only diagonal steps. In the SDTW formulation from [10], the cost of a diagonal step is equal to the cost of a vertical or horizontal step. Thus, for a uniform cost matrix (which is probable at the initial training phase due to random network initialization), taking a diagonal step only accumulates half the cost compared to going "around the corner", i.e., one step in the vertical and one in the horizontal direction, or vice versa. This diagonalizing behavior leads, on average, to decent soft alignments in the early training phase (as discussed in Section 5.3.2). However, in contrast to the additive diagonal prior strategy, the implicit diagonalization of alignments is not reduced during the training, as can be seen in Figure 4e, which still exhibits strong diagonal components after 25 training epochs. Thus, the softly aligned SDTW targets seldom match the reference targets and performance remains low, resulting in a mean F-measure of 0.53 in Table 2.

Note that the sequence unfolding strategy adds a significant computational overhead compared to the previous two strategies, as unfolding always corresponds to using a target sequence length of $M = N$. The forward and backward pass of the SDTW loss function both have linear complexity w.r.t. the sequence lengths $\mathcal{O}(MN)$ [10]. Thus, in our setting with $N = 500$ and a mean length of the weak target sequences in the test set of $M = 24$, the unfolding strategy leads to an increase in the computational cost of the SDTW loss by a factor of more than 20.

6. CONCLUSION AND OUTLOOK

In this paper, we analyzed DNN training instabilities with SDTW as a loss function by the example of PCE. By analysis of the soft alignment matrix, we argued that alignment mismatch in the early training phase often causes a collapse of the training procedure. Motivated by these findings, we investigated three strategies for stabilizing the early training phase. We found that the previously applied strategy of unfolding the weakly aligned target sequence leads to almost exclusively diagonal alignments due to a naïve weighting of horizontal, vertical, and diagonal alignment steps. Furthermore, this strategy is computationally inefficient, as it increases the target sequence length. In contrast, the two introduced strategies of hyperparameter scheduling and diagonal prior can be implemented with negligible additional computational cost and stabilize SDTW-based training by two different mechanisms. The hyperparameter scheduling strategy promotes smooth alignments in the early training phase, which increases the probability of the predicted frame being at least partially aligned to the correct target. Penalizing off-diagonal alignments in the SDTW cost matrix by an additive diagonal prior is a strategy that initially restricts the soft alignment to a region of high probability. Experimental evaluation showed that these strategies reliably stabilize the SDTW training process. Implementing them as a default in the SDTW loss highly increases convergence rates.

Future research on SDTW-based loss functions in MIR applications might incorporate musically informed prior information, e.g., based on note durations or tempo annotations extracted from the musical score. Furthermore, the preference of diagonal alignment steps could be addressed by choosing different step weights.

7. ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG MU 2686/7-2). 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.

8. REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] F. Korzeniowski and G. Widmer, "Feature learning for chord recognition: The deep chroma extractor," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 37–43.

[3] C. Weiß, J. Zeitler, T. Zunner, F. Schuberth, and M. Müller, "Learning pitch-class representations from score–audio pairs of classical music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 746–753.

[4] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 96–104, 2005.

[5] C. Weiß and G. Peeters, "Training deep pitch-class representations with a multi-label CTC loss," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 754–761.

[6] 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.

[7] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, Pittsburgh, Pennsylvania, USA, 2006, pp. 369–376.

[8] 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.

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

[10] 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.

[11] A. Mensch and M. Blondel, "Differentiable dynamic programming for structured prediction and attention," in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholmsmässan, Stockholm, Sweden, 2018, pp. 3459–3468.

[12] 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.

[13] 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.

[14] M. Blondel, A. Mensch, and J. Vert, "Differentiable divergences between time series," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Virtual, 2021, pp. 3853–3861.

[15] K. Shih, R. Valle, R. Badlani, A. Łańcucki, W. Piang, and B. Catanzaro, "RAD-TTS: Parallel flow-based TTS with robust alignment learning and diverse synthesis," in *International Conference on Machine Learning (ICML), Third Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, Virtual, 2021.

[16] 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.

[17] 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.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference for Learning Representations (ICLR)*, San Diego, California, USA, 2015.

[19] 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.