LEARNING COMPLEX BASIS FUNCTIONS FOR INVARIANT REPRESENTATIONS OF AUDIO

Stefan Lattner,[1] **Monika Dörfler,**[2] **Andreas Arzt**[3]

[1] Sony Computer Science Laboratories (CSL), Paris, France
[2] NuHAG, Faculty of Mathematics, University Vienna
[3] Institute of Computational Perception, JKU Linz

ABSTRACT

Learning features from data has shown to be more successful than using hand-crafted features for many machine learning tasks. In music information retrieval (MIR), features learned from windowed spectrograms are highly variant to transformations like transposition or time-shift. Such variances are undesirable when they are irrelevant for the respective MIR task. We propose an architecture called Complex Autoencoder (CAE) which learns features invariant to orthogonal transformations. Mapping signals onto complex basis functions learned by the CAE results in a transformation-invariant "magnitude space" and a transformation-variant "phase space". The phase space is useful to infer transformations between data pairs. When exploiting the invariance-property of the magnitude space, we achieve state-of-the-art results in audio-to-score alignment and repeated section discovery for audio. A PyTorch implementation of the CAE, including the repeated section discovery method, is available online. [1]

1. INTRODUCTION

Learning from audio data most commonly involves some prior processing of the raw sound signals. The most popular features are derived from a spectrogram, which consists of the magnitude values of the Fourier transform of a windowed signal of interest. In a Fourier transform, a signal is projected onto sine and cosine functions of different frequencies. One of the main reasons for the spectrogram to be more useful than the usage of the Fourier coefficients in their complex form is the fact that the magnitude spectrum of a signal is invariant to a translation of the original signal.

This invariance to translation, desirable for most learning problems in audio, results from the fact that cosine and sine represent the real and imaginary parts, respectively, of the *complex eigenvectors* of translation. More generally, the eigenvectors of an orthogonal transformation (e.g., translation, rotation, reflection, but most general all permutations - "shuffling pixels") constitute an orthonormal basis of complex vectors with corresponding eigenvalues of magnitude 1. Hence, as we shall see in detail, the absolute value of a signal's coefficients with respect to this basis is invariant to that transformation. We harness this invariance property for learning representations invariant to different orthogonal transformations.

In particular, transposition-invariance is an essential property for several MIR tasks, including alignment tasks, repeated section discovery, classification tasks, cover song detection, query by humming, or representations of acapella recordings with pitch drift. Different methods have aimed at learning transposition-invariant representations. For example, in [33] close time steps in chromagrams are cross-correlated in order to calculate distances between pitch classes, and in [20], successive n-grams of constant-Q transformed (CQT) representations of audio are compared using a Gated Autoencoder (GAE) architecture. Most similar to our approach, the transposition-invariant magnitudes of Fourier transformations applied to chromagram-like representations of audio are facilitated in [3] and [23]. However, instead of using 2D Fourier transforms with fixed basis functions, we learn the relevant basis functions starting from CQT representations of audio. This learning of basis functions has some advantages over using pre-defined bases. For example, the number of basis elements necessary to discriminate between signals can be reduced compared to a common Fourier transform (e.g., for transposition- and time-shift invariance we use $M = 256$ basis functions for input dimensionality $N = 3840$, while usually $N = M$). Furthermore, our approach is generic and has the potential to learn other musically interesting invariances (e.g., towards tempo-change, diatonic transposition, inversion, or retrograde).

The contribution of this paper is a simple training method for learning invariant representations from data pairs and its application to two MIR tasks. First, we show that when using the features learned by the Complex Autoencoder (CAE) from audio in CQT representation, we can improve the state-of-the-art in a transposition-invariant repeated section discovery task in audio. Second, the CAE

[1] https://github.com/SonyCSLParis/cae-invar

features prove useful in an audio-to-score alignment task, where we show that most of the time, they yield better results than Chroma features and features calculated with a GAE. We also compare the CAE with a GAE in classifying rotated MNIST digits, based on rotation-invariant features learned by the CAE. The reason we also perform experiments on MNIST is that it allows us to show the efficacy of the model with respect to rotation-invariance. Furthermore, the class labels available in the MNIST dataset help to highlight the different clusters in the rotation-invariant space (see Figure 4).

In particular for translations, the model can be interpreted as measuring distances in the data. Training the CAE for transposition and time-shift invariance on short windows of audio in CQT representation, therefore, leads to representations of rhythmic structures and tonal relationships present in the windows, what we exploit in the repeated section discovery task. Representing rhythmic structures is less critical in music alignment tasks; it can even be disadvantageous when the aligned signals differ in tempo. We show in the alignment task that it is sufficient to train the CAE only for transposition-invariance (i.e., time-shift transformation) on rather short n-grams of audio in CQT representation. This is because compared to repeated section discovery, where rhythmic patterns can help to identify similar parts, in the alignment task, a dynamic time-warping algorithm keeps track of the respective positions in the music pieces.

The CAE can be trained in an unsupervised manner on data pairs obeying the relevant transformations. Thereby, we obtain a "magnitude space" and a "phase space", as it is known from a Fourier transform. The "magnitude space" of the CAE is invariant to all the learned transformations. Remarkably, the *phase shifts* a projected signal undergoes during a transformation (i.e., the relative vector in the "phase space" of the CAE) are discriminative with respect to the type and the distance of a transformation. This is an interesting property which could be exploited for determining types of relations between musical fragments in structure analysis tasks.

The paper is structured as follows. In Section 2 existing work related to the proposed method is discussed. In Section 3 we describe the model and its mathematical background and Section 4 describes the general training procedure. In Section 5, we show results on three different tasks: discovery of repeated themes and sections, audio-to-score alignment, and classification of MNIST digits. We end the paper with a conclusion and a discussion of possible directions for future work (Section 6).

2. RELATED WORK

Generally, mid-level representations in neural networks are highly variant to transformations in the input. The most common and well-known way to obtain shift-invariance in convolutional architectures is max-pooling [4]. However, full shift-invariance can only be achieved step-wise by applying max-pooling over several layers. A whole line of research therefore aims to obtain representations invariant to different kinds of transformations using other approaches. Inspiration for the proposed model was drawn from [25], where complex basis functions are learned using a GAE. An approach similar to ours is to facilitate harmonic functions or wavelets, either in weight initialization [29, 37], for modulating learned filters using Gabor functions [22], or for using fixed wavelets in scattering transforms [5, 32]. Similarly, harmonic functions can be pre-defined, e.g., to obtain rotation invariance in convolutional architectures [36], or learned, e.g., by assuming "temporal slowness" of features in videos [21, 28], while pitch-invariant timbral features are learned in [30] by enabling convolution through the frequency domain.

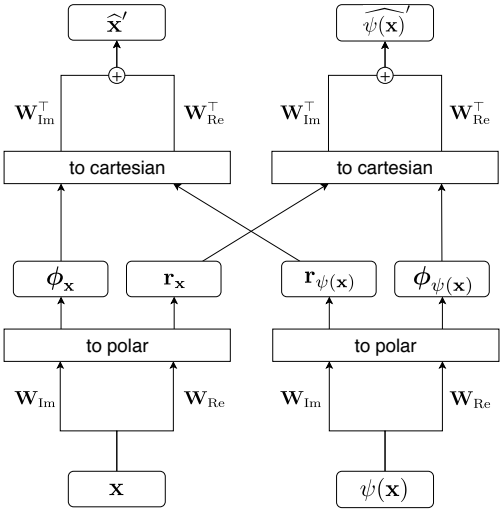


Figure 1: Schematic illustration of reconstruction during training. Both the input $\mathbf{x}$ and its transformed counterpart $\psi(\mathbf{x})$ are projected onto complex basis pairs $\{\mathbf{W}_{\mathrm{Re}}, \mathbf{W}_{\mathrm{Im}}\}$ and expressed in polar form. Then, using the swapped magnitude vectors $\{\mathbf{r}_{\psi(\mathbf{x})}, \mathbf{r}_{\mathbf{x}}\}$ and the original phase vectors $\{\boldsymbol{\phi}_{\mathbf{x}}, \boldsymbol{\phi}_{\psi(\mathbf{x})}\}$, the data is reconstructed by performing the inverse operations.

Most of the approaches mentioned so far (including our approach) aim at invariances to relatively simple, affine transformations. Invariances to more complex, non-linear transformations are usually achieved by redundancy (e.g., an object is presented from different camera angles or under different lighting conditions), which typically requires bigger architectures. That way, invariance can be learned by an explicit transformation of the input [16], by enforcing similarity in the latent space [24], or by using a Siamese architecture and pre-defined transformation sets [18]. Other methods involve rotating convolution kernels during training [38] and dealing with input deformations using learned, dynamic convolution grids [8]. In [9] an end-to-end CNN which acts on raw audio learns Gabor-like filters similar to those extracted by the CAE, see Figure 2.

3. MODEL AND MATHEMATICAL BACKGROUND

We aim at learning orthogonal transformations encoding certain invariances of a class of signals which are known or

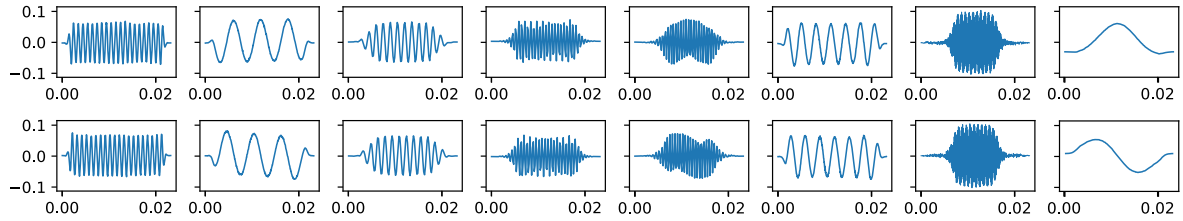


Figure 2: Some examples of real (top) and imaginary (bottom) basis vectors learned from audio signals (time in seconds).

assumed to be useful for a particular learning task at hand. To this end, we leverage the particular properties of orthogonal transformations, which we now describe. A transformation $\psi : \mathbb{R}^N \to \mathbb{R}^N$ is orthogonal if $\langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle = \langle \mathbf{x}, \mathbf{y} \rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$. By $\langle \mathbf{x}, \mathbf{y} \rangle$ we denote the inner product on $\mathbb{R}^N$ or $\mathbb{C}^N$, respectively. Orthogonal transformations are distinguished by the fact that they possess a diagonalization with eigenvalues which all have absolute value 1. Hence, in any non-trivial case, the eigenvalues are complex and so are the corresponding eigenvectors. More precisely, if ψ is orthogonal, there exists a unitary matrix [2] $\mathbf{W}$ and eigenvalues λ_j, $j = 1, \ldots, N$, with $|\lambda_j| = 1$ for all j, such that

$$\psi(\mathbf{x}) = \mathbf{W}^* \mathbf{D} \mathbf{W} \mathbf{x} \tag{1}$$

Here $\mathbf{D}$ denotes the diagonal $N \times N$ matrix with the eigenvalues λ_j in the diagonal. We hence have the following statement.

Proposition 1. *If an orthogonal transformation $\psi : \mathbb{R}^N \to \mathbb{R}^N$ is diagonalised by a unitary matrix $\mathbf{W}$, then the feature vector given by $|\mathbf{W}\mathbf{x}|$ for all $\mathbf{x} \in \mathbb{R}^N$ is invariant to ψ. In other words, we have $|\mathbf{W}\mathbf{x}| = |\mathbf{W}\psi(\mathbf{x})|$ for all $\mathbf{x} \in \mathbb{R}^N$.*

Proof. According to (1) and since $\mathbf{W}\mathbf{W}^*\mathbf{x} = \mathbf{x}$, we have

$$\mathbf{W}\psi(\mathbf{x}) = \mathbf{D}\mathbf{W}\mathbf{x}, \tag{2}$$

which can be written coordinate-wise as

$$\langle w_j, \psi(\mathbf{x}) \rangle = \lambda_j \langle w_j, \mathbf{x} \rangle, \; j = 1, \ldots, N,$$

where w_j denotes the $j-$th row of the complex, unitary matrix $\mathbf{W}$. Hence, since $|\lambda_j| = 1$ for all j, we have $|\langle w_j, \psi(\mathbf{x}) \rangle| = |\langle w_j, \mathbf{x} \rangle|$ for all $\mathbf{x} \in \mathbb{R}^N$ and thus $|\mathbf{W}\mathbf{x}| = |\mathbf{W}\psi(\mathbf{x})|$ as claimed. □

In CAE-learning, we can only deal with real weights and are usually interested in learning less than N basis vectors. Hence, we split an $M \times N$-dimensional submatrix of the unitary, complex matrix of eigenvectors $\mathbf{W}$ into a real and an imaginary part $\mathbf{W}_{\text{Re}} \in \mathbb{R}^{M \times N}$ and $\mathbf{W}_{\text{Im}} \in \mathbb{R}^{M \times N}$ which map $\mathbf{x}$ onto the real and imaginary part of $\mathbf{W}\mathbf{x}$, respectively. The complex data $\mathbf{W}_{\text{Re}}\mathbf{x} + i\mathbf{W}_{\text{Im}}\mathbf{x}$ is then expressed in its polar form by a phase vector $\boldsymbol{\phi}_{\mathbf{x}} \in [0, 2\pi)^M$ and a magnitude vector $\mathbf{r}_{\mathbf{x}} \in \mathbb{R}^M_{\geq 0}$.

According to Proposition 1, assuming that $\mathbf{W}_{\text{Re}} + i\mathbf{W}_{\text{Im}}$ consist of orthonormal eigenvectors of ψ leads to the magnitude of projections of a signal $\mathbf{x}$ and a transformed version $\psi(\mathbf{x})$ onto these eigenvectors to be equal. This property is imposed during training by expressing $\mathbf{W}\mathbf{x}$ and $\mathbf{W}\psi(\mathbf{x})$ in their respective polar forms

$$\boldsymbol{\phi}_{\mathbf{x}} = \text{atan2}(\mathbf{W}_{\text{Re}}\mathbf{x}, \mathbf{W}_{\text{Im}}\mathbf{x}), \tag{3}$$

and

$$\mathbf{r}_{\mathbf{x}} = \sqrt{(\mathbf{W}_{\text{Re}}\mathbf{x})^2 + (\mathbf{W}_{\text{Im}}\mathbf{x})^2}, \tag{4}$$

and swapping the magnitude vectors before reconstruction.

Accordingly, we reconstruct $\mathbf{x}$ given its own phase representation $\boldsymbol{\phi}_{\mathbf{x}}$ and the magnitude representation of the transformed signal $\mathbf{r}_{\psi(\mathbf{x})}$ as follows (see Figure 1):

$$\widehat{\mathbf{x}}' = \mathbf{W}_{\text{Re}}^\top(\mathbf{r}_{\psi(\mathbf{x})} \cdot \sin \boldsymbol{\phi}_{\mathbf{x}}) + \mathbf{W}_{\text{Im}}^\top(\mathbf{r}_{\psi(\mathbf{x})} \cdot \cos \boldsymbol{\phi}_{\mathbf{x}}). \tag{5}$$

Likewise, we reconstruct the transformed signal $\psi(\mathbf{x})$ as

$$\widehat{\psi(\mathbf{x})}' = \mathbf{W}_{\text{Re}}^\top(\mathbf{r}_{\mathbf{x}} \cdot \sin \boldsymbol{\phi}_{\psi(\mathbf{x})}) + \mathbf{W}_{\text{Im}}^\top(\mathbf{r}_{\mathbf{x}} \cdot \cos \boldsymbol{\phi}_{\psi(\mathbf{x})}). \tag{6}$$

The CAE is then trained by minimizing the symmetric reconstruction error

$$\frac{1}{N}\sum_i^N (x_i - \widehat{x}_i')^p + \frac{1}{N}\sum_j^N (\psi(x_j) - \widehat{\psi(x_j)}')^p, \tag{7}$$

where $p \in \{1, 2\}$ has shown to work well in practice. Training on sufficiently many transform pairs thus leads to learning the weights of the unitary matrix $\mathbf{W}$, which diagonalises ψ. While the magnitudes of the coefficient vectors $\mathbf{W}\psi(\mathbf{x})$ are equal to $\mathbf{W}\mathbf{x}$, the transformation itself is then represented by the differences in the phase vectors $\Delta\boldsymbol{\phi} = \boldsymbol{\phi}_{\mathbf{x}} - \boldsymbol{\phi}_{\psi(\mathbf{x})}$ (see Figure 4(b)). As an example of complex basis vectors learned by the CAE, see Figure 2, where the CAE was trained on time-shifted audio signals in the time domain, yielding complex Gabor-like filters.

4. TRAINING

For all the experiments described below, we choose 256 complex basis vectors and train the model for 500 epochs with a learning rate of 1e-3. We use a batch size of 1000, and we sample 100k transformations per epoch, generally picking random instances from the train set to be transformed. The training data is standardized, and 50% dropout is used on the input. We set $p = 1$ (see Equation 7) for the audio experiments, and $p = 2$ for the MNIST experiment. In the alignment experiment, we also penalize the mean of norms of all basis vectors and the deviation of the individual basis vectors' norms to the average norm over all basis vectors. In the MNIST experiment, the norm of all basis vectors is set to 0.4 after every batch. For information about the training data see the respective experiment section below.

[2] A complex matrix $\mathbf{W}$ is unitary, if $\mathbf{W}^* = \mathbf{W}^{-1}$.

Algorithm	F_{est}	P_{est}	R_{est}	$F_{\text{o(.5)}}$	$P_{\text{o(.5)}}$	$R_{\text{o(.5)}}$	$F_{\text{o(.75)}}$	$P_{\text{o(.75)}}$	$R_{\text{o(.75)}}$	$\mathbf{F_3}$	P_3	R_3	Time (s)
CA (ours)	52.53	63.10	50.29	63.58	64.60	62.68	67.20	68.73	65.82	**52.16**	**62.51**	49.78	**69**
GAE intervals [20]	**57.67**	**67.46**	59.52	58.85	61.89	56.54	68.44	72.62	64.86	51.61	59.60	**55.13**	194
VMO deadpan [34]	56.15	66.80	57.83	**67.78**	**72.93**	**64.30**	**70.58**	**72.81**	**68.66**	50.60	61.36	52.25	96
SIARCT-CFP [7]	23.94	14.90	**60.90**	56.87	62.90	51.90	-	-	-	-	-	-	-
Nieto [27]	49.80	54.96	51.73	38.73	34.98	45.17	31.79	37.58	27.61	32.01	35.12	35.28	454

Table 1: Different precision, recall and f-scores (adopted from [34], details on the metrics are given in [6]) of different methods in the Discovery of Repeated Themes and Sections MIREX task, for symbolic music and audio. The $\mathbf{F_3}$ score constitutes a summarization of all metrics.

5. EXPERIMENTS

5.1 Discovery of Repeated Themes and Sections

In the MIREX task "Discovery of Repeated Themes and Sections", [3] the performance of different algorithms to identify repeated (and possibly transposed) patterns in symbolic music and audio is tested. The commonly used JKUPDD dataset [6] contains 26 motifs, themes, and repeated sections annotated in 5 pieces by J. S. Bach, L. v. Beethoven, F. Chopin, O. Gibbons, and W. A. Mozart. We use the audio versions of the dataset and preprocess them the same way as the training data described below.

The CAE is trained on 100 random piano pieces of the MAPS dataset [12] (subset MUS) at a sampling rate of 22.05 kHz. We choose a constant-Q transformed spectrogram representation with a hop size of 1984. The range comprises 120 frequency bins (24 per octave), starting from a minimal frequency of 65.4 Hz. The spectrogram is split into n-grams of 32 frames. The set of transformations applied to the data during training $\Psi_{\text{pshift, tshift}}$ contains transposition by $[-24, 24]$ frequency bins and time shifts by $[-12, 12]$.

After training, all n-grams of the JKUPDD dataset are projected into the transformation-invariant magnitude space. Using these representations, a self-similarity matrix is built for each piece using the reciprocal of the cosine distance. The matrices are then filtered with an identity matrix of size 10×10. Then, their main diagonals are set to zero. Finally, the matrices are first normalized and then centered by subtracting their medians.

For finding repeated sections, the method proposed in [20] is adopted, which finds diagonals in a self-similarity matrix using a threshold. As we normalized the matrices to zero median, the threshold chosen in this experiment is close to zero (i.e., 0.01).

5.1.1 Results and Discussion

Table 1 shows the results of the experiment. Using our method, we could slightly outperform the Gated Autoencoder approach proposed in [20]. By visual inspection of the self-similarity matrix, we noted very precise diagonals at repetitions, while almost no similarity is indicated on other parts (this is different from the self-similarity plots provided in [20]). This selectivity, which may also result from the cosine distance, probably contributes to the slightly higher precision of the proposed method.

[3] http://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_&_Sections

ID	Dataset	Files	Duration
CE	Chopin Etude	22	$\sim$ 30 min.
CB	Chopin Ballade	22	$\sim$ 48 min.
MS	Mozart Sonatas	13	$\sim$ 85 min.
RP	Rachmaninoff Prelude	3	$\sim$ 12 min.
B3	Beethoven 3	4	$\sim$ 52 min.
M4	Mahler 4	4	$\sim$ 58 min.

Table 2: The evaluation data set for the alignment experiments (see text).

5.2 Invariant Audio-to-Score Alignment

The task of synchronising an audio recording of a music performance and its score has already been studied extensively (see e.g. [10, 11, 14, 15, 17, 26]). Here, we compare synchronisation results using the proposed method (CAE) to traditional Chroma features and the GAE features introduced in [20], which were used for music alignment in [2].

For the alignment experiments we follow [2], using the same setup and the same data (see Table 2 for a summary). *CB* and *CE* consist of 22 recordings of excerpts of the Ballade Op. 38 No. 1 and the Etude Op. 10 No. 3 by Chopin [13], *MS* contains performances of the first movements of the piano sonatas KV279-284, KV330-333, KV457, KV475 and KV533 by Mozart [35], and *RP* consists of three performances of the Prelude Op. 23 No. 5 by Rachmaninoff [1]. Finally, *B3* and *M4* are annotated recordings of Beethoven's 3^{rd} and Mahler's 4^{th} symphonies. Note that CB, CE, MS, and RP consist of piano music, while B3 and M4 consist of orchestral music, but we will use the same model for the whole data set, which was trained on piano music only.

The scores are provided in the MIDI format, with the global tempi set such that the scores roughly match the average length of the given performances, i.e., both representations have the same average tempo, but there still exist substantial differences in local tempi. The scores are then synthesized with the help of *timidity* [4] and a publicly available sound font. The resulting audio files are used as score representations for the alignment experiments. To compute the alignments, a multi-scale variant of the dynamic time warping (DTW) algorithm (see [26] for a detailed description of DTW) is used, namely FastDTW [31] with the radius parameter set to 50.

The CAE is trained the same way and on the same data as described in Section 5.1 but here we choose a CQT hop size of 448. Furthermore, for this experiment, we use an

[4] https://sourceforge.net/projects/timidity/

		'Un-transposed' Data			Transp.
DS	**Metric**	**Chroma**	**GAE**	**CAE**	**CAE**
CB	1st Quartile	15 ms	**10 ms**	**10 ms**	10 ms
	Median	34 ms	22 ms	**21 ms**	21 ms
	3rd Quartile	80 ms	39 ms	**37 ms**	38 ms
	Err. $\leq$ 50 ms	64%	83%	**84%**	84%
	Err. $\leq$ 250 ms	85%	**94%**	**94%**	94%
CE	1st Quartile	13 ms	**10 ms**	**10 ms**	9 ms
	Median	29 ms	21 ms	**19 ms**	18 ms
	3rd Quartile	56 ms	36 ms	**32 ms**	30 ms
	Err. $\leq$ 50 ms	71%	87%	**90%**	91%
	Err. $\leq$ 250 ms	94%	**96%**	**96%**	97%
MS	1st Quartile	7 ms	**6 ms**	**6 ms**	6 ms
	Median	16 ms	13 ms	**12 ms**	12 ms
	3rd Quartile	31 ms	25 ms	**22 ms**	22 ms
	Err. $\leq$ 50 ms	85%	90%	**91%**	92%
	Err. $\leq$ 250 ms	98%	**100%**	**100%**	99%
RP	1st Quartile	17 ms	14 ms	**9 ms**	9 ms
	Median	43 ms	34 ms	**20 ms**	21 ms
	3rd Quartile	113 ms	90 ms	**55 ms**	69 ms
	Err. $\leq$ 50 ms	55%	63%	**74%**	70%
	Err. $\leq$ 250 ms	91%	90%	**95%**	93%
B3	1st Quartile	20 ms	25 ms	**17 ms**	18 ms
	Median	48 ms	54 ms	**39 ms**	42 ms
	3rd Quartile	108 ms	104 ms	**83 ms**	99 ms
	Err. $\leq$ 50 ms	52%	47%	**59%**	56%
	Err. $\leq$ 250 ms	88%	90%	**91%**	88%
M4	1st Quartile	46 ms	50 ms	**42 ms**	46 ms
	Median	110 ms	129 ms	**99 ms**	110 ms
	3rd Quartile	278 ms	477 ms	**255 ms**	290 ms
	Err. $\leq$ 50 ms	27%	25%	**29%**	27%
	Err. $\leq$ 250 ms	73%	66%	**75%**	72%

Table 3: Comparison of the proposed features CAE to Chroma features and features computed via a gated autoencoder GAE. The first three columns show results on normal, i.e., un-transposed data. The rightmost column shows the average result of alignments of the original performances to scores in 12 different transpositions.

n-gram size of 8. The set of transformations applied to the data during training Ψ_{pshift} are transpositions by $[-24, 24]$ frequency bins.

5.2.1 Results and Discussion

In the alignment experiments, we compare the proposed CAE features to the results presented in [2], where Chroma features and features computed via a gated autoencoder (GAE) were compared to each other. Table 3 gives an overview of the results. The first three columns show that the proposed CAE features consistently outperform the other two methods in the normal alignment setting (i.e., without any transpositions). Additionally, the rightmost column shows that for CAE, the results essentially stay the same, even when the alignment is computed with transposed versions of the score. This demonstrates the invariance to transpositions, which is a serious advantage over the Chroma features.

As has been shown in [2], the GAE features are highly sensitive to tempo differences between the score representation and the performance. To see if the proposed CAE features suffer from the same problem, we repeated this experiment and performed alignments on artificially slowed-down and sped-up score representations. The results are shown in Table 4. For all tested features, the degree to which the tempi of the score representation and the performance match influences the alignment quality. The experiments suggest that CAE is less sensitive to differences in tempi than GAE, but the Chroma features still have the advantage over GAE in this matter. We also conducted experiments with more extreme tempi, which further confirmed this trend. The reason for the higher robustness to tempo differences of the CAE features over the GAE features may be found in the way the GAE features are computed. In a GAE, two inputs $\{\mathbf{x}_{t-n,\dots,t}, \mathbf{x}_{t+1}\}$ are compared to one another, and the features are sensitive to the position and order of events in $\mathbf{x}_{t-n,\dots,t}$. When training a CAE only for transposition-invariance, the resulting features represent mainly distances in the frequency-dimension of the input and tend to be invariant to the position of events in time.

5.3 Classification of MNIST digits

We test the ability of the CAE to learn rotation-invariance in 2D images using randomly rotated MNIST digits (the dataset was first described in [19]). Given the set of rotations Ψ_{rot} with rotation angles $[0, 2\pi)$ about the origin of the images. For any MNIST instance $\mathbf{x}_k$, we create a rotated version $\psi_i(\mathbf{x}_k)$ and a further rotated version $\psi_j(\psi_i(\mathbf{x}_k))$, where $\psi_i, \psi_j \in \Psi_{\text{rot}}$, resulting in pairs $\{\psi_i(\mathbf{x}_k), \psi_j(\psi_i(\mathbf{x}_k))\}$. After the CAE is trained on 50k such pairs, single randomly rotated instances are projected into the magnitude space. On these projections, a logistic regression classifier is trained to predict the class labels. We test different train set sizes (sampled from the main train set with balanced class distribution). 50-fold cross-validation is used, where evaluation is always performed on 10k test instances, independent of the train set size. For comparison, we perform k-nn classification on the randomly rotated images (i.e., the input space), and unrotated images directly. We choose logistic regression for the magnitude space and k-nn classification for the input space because they showed the overall best results for those representations. This choice, as well as the overall experiment setup, reflects that in [25].

5.3.1 Results and Discussion

Figure 3 shows the results of the rotated MNIST classification task. The error rates of *magnitudes GAE* were obtained by a GAE architecture which was extended to learn basis functions, as reported in [25]. Classification on the magnitude space of the CAE (*magnitudes CAE*) leads to substantially better results than those of the GAE, even though only 256 basis elements are used in the CA, compared to 1000 in the GAE. This is probably due to the explicit training of the CA to learn an invariant magnitude space, while the magnitude space of the GAE is learned indirectly during the learning of transformations. Overall, classification on the rotation-invariant magnitude spaces performs much better than on the input space of rotated images in particular for small train set sizes. The difference in performance between *images (original)* and *magnitudes CA* reflects the gap between a theoretically optimal rotation-invariant representation (as *images (original)* are

		Chroma			GAE			CAE		
DS	**Metric**	$\frac{2}{3}$ **T.**	**Base T.**	$\frac{4}{3}$ **T.**	$\frac{2}{3}$ **T.**	**Base T.**	$\frac{4}{3}$ **T.**	$\frac{2}{3}$ **T.**	**Base T.**	$\frac{4}{3}$ **T.**
CB	Error $\leq$ 50 ms	54%	64%	67%	47%	83%	33%	**80%**	**84%**	**85%**
	Error $\leq$ 250 ms	82%	85%	85%	87%	**94%**	84%	**91%**	**94%**	**94%**
CE	Error $\leq$ 50 ms	69%	71%	73%	40%	87%	38%	**85%**	**90%**	**88%**
	Error $\leq$ 250 ms	90%	94%	94%	**93%**	**96%**	80%	**93%**	**96%**	**95%**
MS	Error $\leq$ 50 ms	79%	85%	75%	84%	90%	74%	**86%**	**91%**	**76%**
	Error $\leq$ 250 ms	98%	98%	97%	**99%**	**100%**	**98%**	**99%**	**100%**	**98%**
RP	Error $\leq$ 50 ms	53%	55%	56%	43%	63%	37%	**67%**	**74%**	**63%**
	Error $\leq$ 250 ms	92%	91%	87%	82%	90%	85%	**95%**	**95%**	**91%**
B3	Error $\leq$ 50 ms	**44%**	52%	**36%**	–	47%	–	39%	**59%**	33%
	Error $\leq$ 250 ms	**83%**	88%	**82%**	–	90%	–	82%	**91%**	**82%**
M4	Error $\leq$ 50 ms	**26%**	27%	**24%**	–	25%	–	24%	**29%**	22%
	Error $\leq$ 250 ms	**75%**	73%	**71%**	–	66%	–	72%	**75%**	65%

Table 4: Results on score representations with different tempi (higher is better). *Base T.* refers to a globally set tempo that ensures that the duration of the score representation is roughly equal to the duration of a typical performance. $\frac{2}{3}$ *T.* and $\frac{4}{3}$ *T.* refer to score representation with the tempo set to $\frac{2}{3}$ and $\frac{4}{3}$ of the base tempo. The two metrics used are the percentage of events that are aligned with an error lower or equal 50 ms and 250 ms (i.e. higher is better). The missing numbers for GAE were not provided in [2].

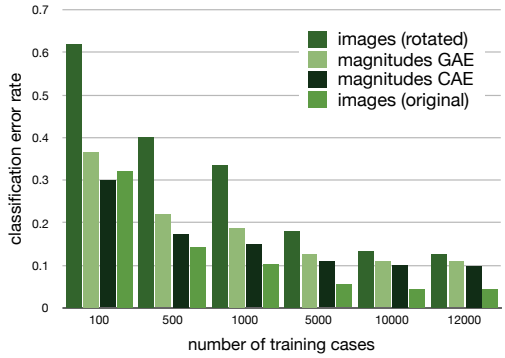
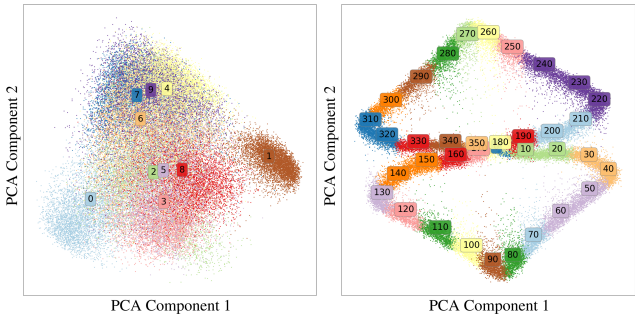


Figure 3: Classification error rates in the input space (images) and the magnitude space (magnitudes) on the rotated MNIST dataset with different train set sizes. "Images (original)" denotes the results of the unrotated MNIST dataset for comparison.



(a) (b)

Figure 4: PCAs of rotated MNIST digits in the magnitude space (a) and the phase-difference space (b) (best viewed in color). The magnitude space represents the data in the absence of the transformations leading to clusters of the digit classes (colored and labeled accordingly). The phase-difference space represents the transformations between images, independent of their identity (colors and labels denote rotation angles quantized into 36 bins).

not rotated), and the representations learned by the CA. On 100 training cases, logistic regression would outperform the k-nn classification on the input spaces, while for all other train set sizes k-nn is superior over logistic regression. Thus, we obtain slightly worse classification performance of *images (original)* on 100 training cases compared to *magnitudes CA*.

Figure 4 shows PCAs of the randomly rotated MNIST digits projected into the magnitude space and the phase-difference space of the CAE. The clusters in the magnitude space indicate that images with the same content (i.e., class label) yield similar projections, independent of their rotations. The clusters in the phase-difference space show that phase differences clearly represent the transformations in the data.

6. CONCLUSION AND FUTURE WORK

The empirical results in this work show that for music alignment, structure analysis, and invariant classification tasks, the features learned by the CAE have advantages over other features, like Chroma features, and features learned by a GAE. As opposed to Chroma features, the CAE features are transposition-invariant, and generally perform better in the alignment task. Compared to the features learned by a GAE, the CAE features are more robust to differences in tempo between alignment data.

Future work should involve investigating the use of the "phase-difference" space of the CAE. For example, qualifying transformations between sections in music could lead to a richer musical structure analysis (e.g., determining mutually transposed parts, or finding sections with similar rhythm but different tonality).

The learned bases could also be used in scattering transforms (i.e., as convolutional filters). As opposed to conventional scattering transforms, where the bases are fixed in general, learned bases may help reducing model sizes or to cover different invariances. Using rotation-invariant filters in convolutional settings may lead to rotation-invariant architectures, similar to what is proposed in [36].

7. ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowsa-Curie grant agreement No. 765068. Monika Dörfler is supported by the Vienna Science and Technology Fund (WWTF) project SALSA (MA14-018).

8. REFERENCES

[1] Andreas Arzt. *Flexible and Robust Music Tracking*. PhD thesis, Johannes Kepler University Linz, 2016.

[2] Andreas Arzt and Stefan Lattner. Audio-to-score alignment using transposition-invariant features. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Paris, France, 2018.

[3] Thierry Bertin-Mahieux and Daniel P. W. Ellis. Large-scale cover song recognition using the 2d fourier transform magnitude. In Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Müller, editors, *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*, pages 241–246. FEUP Edições, 2012.

[4] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 111–118. Omnipress, 2010.

[5] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1872–1886, 2013.

[6] Tom Collins. Discovery of repeated themes and sections. `http://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_%26_Sections`, 2017.

[7] Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. Siarct-cfp: Improving precision and the discovery of inexact musical patterns in point-set representations. In *ISMIR*, pages 549–554, 2013.

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 764–773. IEEE Computer Society, 2017.

[9] S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968, May 2014.

[10] Simon Dixon and Gerhard Widmer. MATCH: A music alignment tool chest. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 492–497, London, UK, 2005.

[11] Daniel P.W. Ellis and Graham E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 1429–1432, Honolulu, Hawaii, USA, 2007.

[12] Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010.

[13] Werner Goebl. The vienna 4x22 piano corpus, 1999. `http://dx.doi.org/10.21939/4X22`.

[14] Maarten Grachten, Martin Gasser, Andreas Arzt, and Gerhard Widmer. Automatic alignment of music performances with structural differences. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 607–612, Curitiba, Brazil, 2013.

[15] Ning Hu, Roger B. Dannenberg, and George Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2003.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2017–2025, 2015.

[17] Cyril Joder, Slim Essid, and Gaël Richard. A comparative study of tonal acoustic features for a symbolic level music-to-score alignment. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, USA, 2010.

[18] Dmitry Laptev, Nikolay Savinov, Joachim M. Buhmann, and Marc Pollefeys. TI-POOLING: transformation-invariant pooling for feature learning in convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 289–297. IEEE Computer Society, 2016.

[19] Hugo Larochelle, Dumitru Erhan, Aaron C. Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In Zoubin Ghahramani, editor, *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 473–480. ACM, 2007.

[20] Stefan Lattner, Maarten Grachten, and Gerhard Widmer. Learning transposition-invariant interval features from symbolic music and audio. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27*, 2018.

[21] Quoc V. Le, Will Y. Zou, Serena Y. Yeung, and Andrew Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3361–3368. IEEE Computer Society, 2011.

[22] Shangzhen Luan, Chen Chen, Baochang Zhang, Jungong Han, and Jianzhuang Liu. Gabor convolutional networks. *IEEE Trans. Image Processing*, 27(9):4357–4366, 2018.

[23] Matija Marolt. A mid-level representation for melody-based retrieval in audio collections. *IEEE Transactions on Multimedia*, 10(8):1617–1625, 2008.

[24] Tadashi Matsuo, Hiroya Fukuhara, and Nobutaka Shimada. Transform invariant auto-encoder. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 2359–2364. IEEE, 2017.

[25] Roland Memisevic and Georgios Exarchakis. Learning invariant features by harnessing the aperture problem. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 100–108. JMLR.org, 2013.

[26] Meinard Müller. *Fundamentals of Music Processing*. Springer Verlag, 2015.

[27] Oriol Nieto and Morwaread M Farbood. Identifying polyphonic patterns from audio recordings using music segmentation techniques. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 411–416, 2014.

[28] Bruno A Olshausen, Charles Cadieu, Jack Culpepper, and David K Warland. Bilinear models of natural images. In *Electronic Imaging 2007*, pages 649206–649206. International Society for Optics and Photonics, 2007.

[29] Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 2056–2063. IEEE Computer Society, 2013.

[30] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik M. Schmidt, Andreas F. Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 637–644, 2018.

[31] Stan Salvador and Philip Chan. FastDTW: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[32] Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 1233–1240. IEEE Computer Society, 2013.

[33] Thomas C Walters, David A Ross, and Richard F Lyon. The intervalgram: an audio feature for large-scale melody recognition. In *Proc. of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*. Citeseer, 2012.

[34] Cheng-i Wang, Jennifer Hsu, and Shlomo Dubnov. Music pattern discovery with variable markov oracle: A unified approach to symbolic and audio representations. In Meinard Müller and Frans Wiering, editors, *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, pages 176–182, 2015.

[35] Gerhard Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146(2):129–148, 2003.

[36] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 7168–7177. IEEE Computer Society, 2017.

[37] Zhuoyao Zhong, Lianwen Jin, and Zecheng Xie. High performance offline handwritten chinese character recognition using googlenet and directional feature maps. In *13th International Conference on Document Analysis and Recognition, ICDAR 2015, Nancy, France, August 23-26, 2015*, pages 846–850. IEEE Computer Society, 2015.

[38] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4961–4970. IEEE Computer Society, 2017.