LEARNING DOMAIN-ADAPTIVE LATENT REPRESENTATIONS OF MUSIC SIGNALS USING VARIATIONAL AUTOENCODERS

Yin-Jyun Luo and Li Su
Institute of Information Science, Academia Sinica, Taiwan
{fredomluo,lisu}@iis.sinica.edu.tw

ABSTRACT

In this paper, we tackle the problem of domain-adaptive representation learning for music processing. Domain adaptation is an approach aiming to eliminate the distributional discrepancy of the modeling data, so as to transfer learnable knowledge from one domain to another. With its great success in the fields of computer vision and natural language processing, domain adaptation also shows great potential in music processing, for music is essentially a highly-structured semantic system having domain-dependent information. Our proposed model contains a Variational Autoencoder (VAE) that encodes the training data into a latent space, and the resulting latent representations along with its model parameters are then reused to regularize the representation learning of the downstream task where the data are in the other domain. The experiments on cross-domain music alignment, namely an audio-to-MIDI alignment, and a monophonic-to-polyphonic music alignment of singing voice show that the learned representations lead to better higher alignment accuracy than that using conventional features. Furthermore, a preliminary experiment on singing voice source separation, by regarding the mixture and the voice as two distinct domains, also demonstrates the capability to solve music processing problems from the perspective of domain-adaptive representation learning.

1. INTRODUCTION

Music is composed, arranged, and performed in various forms residing in different data modalities and domains, yet sharing some common underlying information with each other. Almost all of the music processing tasks essentially extract such commonality as a protocol that enables the transferring or communication among various domains. For example, a piece of music can be either written as a musical score, or rendered as an audio recording; though the later encompasses much more information such as intonation, articulation, emotion, and others not found in the former, they still share common information such as note, pitch, and meter with each other. In light of this property, we devise a framework that aims at eliminating domain-dependent information, to achieve a feature representation that is semantically shared across domains.

In this paper, we study learning representations that embed shared semantic information across different domains, specifically in applications of music signal processing. In order to achieve domain-invariant feature representations, we are essentially considering a domain adaptation problem [28]. Take audio-to-MIDI alignment [30] as an example, while audio and MIDI data are drawn from distinct domains of representation, they share pitch information in common. We explore the transfer learning technique [25] to tackle the problem. Specifically, in addition to transferring model parameters, we also transfer latent representations from one domain to the other.

With its success in computer vision [24, 28, 34] and natural language processing [16, 19, 32], transfer learning has also shown great potential in music information retrieval (MIR). In [6], a linear transformation is learned to project data into a shared latent representation that captures semantic similarity of music. Choi *et al.* uses feature maps of multiple layers derived from a pre-trained convolutional neural network (CNN) for music classification and regression tasks [2], and Park *et al.* exploits the deep model trained for artist recognition as a general feature extractor used for various tasks [26].

Our proposed framework [1] is different from the above-mentioned works. With pairwise training data [2] from two distinct domains, our framework first utilizes a VAE [12], a state-of-the-art unsupervised generative model shown to be effective in representation learning [9, 14], to embed information of data from one domain (the *source domain*) which contains mostly shared semantics into latent representations. Data from the other domain (i.e., the *target domain*) is then mapped to the learned embeddings through a separate neural network, in order to eliminate domain-dependent information. Therefore, the novelty of this paper is an unified framework that combines representation learning and transfer learning altogether, which learns domain-adaptive representations with VAEs that are then transfered from source to target domain. In particular, we empirically validate the framework through three

[1] https://github.com/yjlolo/Domain-Adaptive-VAE

[2] Pairwise data in the context means parallel music events in different domains, e.g., a piece of music written as a score or rendered as an audio, and a recording of singing voice with or without accompaniment.

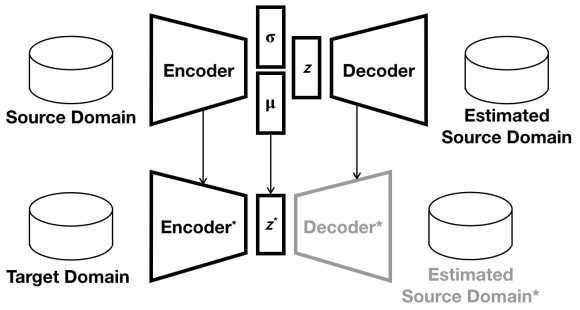

Figure 1. The general architecture of the proposed training framework.

well-known tasks in music signal processing that have not been considered from the perspective of domain adaptation: audio-to-MIDI alignment [1, 13], audio-to-audio alignment [17, 22], and singing voice separation [5, 10].

The rest of the paper is organized as follows. In Section 2, we describe our proposed architecture. The experiments and results are detailed in Section 3 and Section 4, respectively. Conclusions and future work are presented in Section 5.

2. ARCHITECTURE

2.1 Overview

Figure 1 shows our proposed framework in the training phase, which is divided into two modules: the first is a VAE which models the data in source domain, and the second, which can be either an autoencoder (AE) or simply an encoder depending on tasks, models the data in target domain. To facilitate the discussion, we refer *Encoder* (or *Decoder*) and *Encoder** (or *Decoder**) to the *source-domain encoder* (or *source-domain decoder*) and *target-domain encoder* (or *target-domain decoder*), respectively.

The two models are trained sequentially in two steps. First, we train the VAE, using the source-domain data as inputs, and obtain the source-domain latent representations $\boldsymbol{z} := \boldsymbol{z}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. More specifically, given the observation data $\boldsymbol{x}$ in the source domain, and $\boldsymbol{z} \sim p(\boldsymbol{z})$ the latent representation, the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ is modeled as a Gaussian distribution parameterized by the estimated mean and standard deviation of the posterior distribution, namely $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, respectively. In other words, we have $p(\boldsymbol{z}) = \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma^2 I}\right)$ in practice.

Second, after the source-domain model is trained, we train the target-domain model, with the following two transfer learning schemes: 1) the source-domain model parameters are used to initialize the target-domain model parameters, and 2) the source-domain latent representation $\boldsymbol{z}$ is used to regularize the target-domain latent representation $\boldsymbol{z}^*$ with an regularization term $\mathcal{L}(\boldsymbol{z}, \boldsymbol{z}^*)$. The intuition behind this is to leverage knowledge learned by the source-domain VAE to reduce the distributional discrepancy between the source and target domain.

The target-domain decoder, colored in gray in Figure 1, is optional. For example, in the task of music alignment, our purpose is to learn the domain-adaptive features by mapping the data in target domain into the feature distribution of data in source domain, without the need to reconstruct the input data from the latent representation.

It should be noticed that in the inference phase, shown in the left-hand side of Figure 2, the parameter $\boldsymbol{\mu}$ is regarded as $\boldsymbol{z}$. That is, when encoding the source-domain data, $\boldsymbol{\mu}$, the center of a Gaussian distribution, is the representative of $\boldsymbol{z}$. Therefore, $\boldsymbol{\mu}$ is the true latent representation that is transferred to the target domain. More details about the models and experiments are in Section 3.

	Conv1	Conv2	Conv3	Fc1	Gauss
#filters/units	64	128	256	512	L
filter size	$1 \times F$	3×1	2×1	-	-
stride	(1,1)	(2,1)	(2,1)	-	-

Table 1. Encoder network architecture. *Conv* refers to convolutional layers, *Fc* refers to fully connected layers, and *Gauss* refers to the Gaussian parametric layer modeling z.

2.2 Source-domain Model: Variational Autoencoder

Since the source-domain VAE is task-independent, we introduce its detailed architecture first in this subsection, and the task-dependent target-domain model will be introduced later in Section 3. We adopt the VAE architecture proposed in [9], which learns the latent representations and models the generative process of speech segments for voice conversion. In our work, the source-domain input representation is either a segment of singing voice in the tasks of singing voice alignment and separation, or a piano roll for audio-to-MIDI alignment.

The input $\boldsymbol{x}$ of the source-domain VAE is a two-dimensional image of size $T \times F$, where T is the number of time steps and F is the number of frequency bands. The encoder network of this VAE is a CNN with 3 convolution (*Conv*) layers and 1 fully-connected (*Fc*) layer that outputs the latent representation $\boldsymbol{z}$ with dimension L at the *Gauss* layer. The parameters of this CNN are summarized in Table 1. The decoder network is symmetric to the encoder network; it takes $\boldsymbol{z}$ as the input to reconstruct $\boldsymbol{x}$. Batch normalization followed by the activation function `tanh` are used for every layer except for the *Gauss* and the output layers. The objective function for training the VAE is expressed as (1):

$$\mathcal{L}_{vae} = \mathcal{L}_{rec} + \mathcal{L}_{KL}\,, \tag{1}$$

where the total loss function of the VAE, $\mathcal{L}_{vae}$, contains two terms: the reconstruction loss function $\mathcal{L}_{rec} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})]$, the negative expected log-likelihood of $\boldsymbol{x}$, and the KL-Divergence loss $\mathcal{L}_{KL} = \mathrm{KL}[q(\boldsymbol{z}|\boldsymbol{x})\,||\,p(\boldsymbol{z})]$, which regularizes the distance between the posterior and the Gaussian distribution. In variational inference, the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$ is approximated by $q(\boldsymbol{z}|\boldsymbol{x})$. For more implementation details of the VAE, we refer the readers to [3, 12].

3. EXPERIMENTS

We discuss the following three tasks: 1) audio-to-MIDI alignment, 2) audio-to-audio alignment, and 3) singing voice separation. In this section, we elaborate the goals, the datasets, the task-dependent target-domain models, the input data representations, and the evaluation processes for each of these three tasks. Experiment results will be discussed in Section 4.

All of the models discussed in the following are implemented with `PyTorch` [27], and are trained using stochastic gradient descent with the Adam optimizer [11]. The optimizer is parametrized by: learning rate $= 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The mini-batch size is set to 128 instances of input segments.

3.1 Task 1: Audio-to-MIDI Alignment

The first experiment we consider is to align an audio recording of piano to its corresponding MIDI file. Although this is a rather well-studied task [13, 23, 30], we re-investigate this task from the perspective of domain adaptation: using the learned latent representations for the feature on which dynamic time warping (DTW) is performed.

3.1.1 Dataset

We use a subset of the MAPS dataset [4], *ENSTDkCI*, which contains 30 piano recordings performed by a Yamaha Disklavier auto-piano together with MIDI files that generate the recordings. We use 24 and 6 pieces of the subset for training and validation, respectively.

3.1.2 Model

The goal of our framework is to map a frame of audio feature and piano roll into the same representation if they are of the same music event. To do this, we first define the MIDI pieces as the source domain data, and the audio pieces as the target-domain data. Then, we train the source-domain VAE and obtain the learned source-domain latent representation $\boldsymbol{z}$. We then use this representation $\boldsymbol{z}$ as the learning target to train the target-domain model, a single encoder taking audio data as input, with its architecture the same as the source-domain encoder. To be more specific: given a pair of MIDI-audio input data that are of the same event in the music, the source-domain VAE maps the MIDI into a representation in a low-dimensional Gaussian distribution, and the target-domain encoder is then trained to map the audio input data to that distribution.

The learning task in the target domain is essentially a regression task. The training objective function for source domain is the same as (1), while the objective function for the target domain $\mathcal{L}_{encoder}$ is

$$\mathcal{L}_{encoder} = \mathcal{L}_{MSE}\left(\boldsymbol{z}, \boldsymbol{z}^*\right), \tag{2}$$

which is the mean squared error between the encoded latent representations of the source-domain encoder $\boldsymbol{z}$ and the ones of the target-domain encoder $\boldsymbol{z}^*$. Notice that (2) is only applicable when we have parallel source-target pairs.

Instead of audio, MIDI is regarded as the source-domain data because the latent representation we obtain should be more related to MIDI which contains mostly the shared semantics with audio, i.e., pitch; we let the target-domain encoder eliminate the information residing in audio while unrelated to MIDI (e.g., spectral-related information) in order to get succinct representations for alignment.

3.1.3 Data Representation

MIDI files are represented as piano-roll representation with 128 pitch classes, while its associated audio recordings are represented using Mel-scaled spectrogram with 128 filter banks, derived from power magnitude spectrum of 1024-point short-time Fourier transform (STFT). To compute the STFT, we use Hanning window with window size of 64 ms and hop size of 20 ms. An input data for the source-domain VAE (or the target-domain encoder) is a segment of a piano-roll (or Mel-scaled spectrogram) with 21 frames, or equivalently, 400 ms, leading to the input dimensions of $T = 21$ and $F = 128$. To reduce the memory load, we only collect segments every 10 frames of each clip for training.

3.1.4 Evaluation

To evaluate our proposed feature representation for audio-to-MIDI alignment, we apply non-linear time-stretching to the audio recordings so as to see if the features are robust against the distortion and can still be aligned to the original MIDI well. We follow the methodology in [17] for non-linear time-stretch.

The proposed feature representation of MIDI can be derived as follows: we express MIDI as piano roll and use it as the input to the source-domain encoder to obtain the encoded latent representation as our proposed feature; the process is illustrated in Figure 3 with the solid blue line. On the other hand, in the target domain, we firstly apply time-stretching distortion to the audio recordings, represent the audio stream with Mel-scaled spectrogram described in Section 3.1.3, and utilize the outputs of the target-domain encoder as our final feature representation; the green solid line in Figure 3 describes the process. Overall, the derivation of the proposed feature representation during inference is illustrated in the left panel of Figure 2.

For comparison, we consider chroma as a baseline to represent both domains, illustrated in Figure 3 with the loose dash lines colored in blue and green, respectively. Regarding the implementation of chroma, we use `chroma_stft` in the `librosa` library [18] for audio and `get_chroma` in the `pretty_midi` library [29] for MIDI. The other baseline is to use the piano roll for MIDI and Mel-scaled spectrogram for audio, illustrated in Figure 3 with the dense dash lines colored in blue and green, respectively.

We utilize DTW to align the feature representations and compute the alignment accuracy. The accuracy is calculated by an error measure *e* whichp compares the discrepancy between the estimated warping path and the ground-truth one [36] instead of the conventional note-level alignment accuracy, because the error measure *e* allows more subtle comparison on frame-level evaluation.

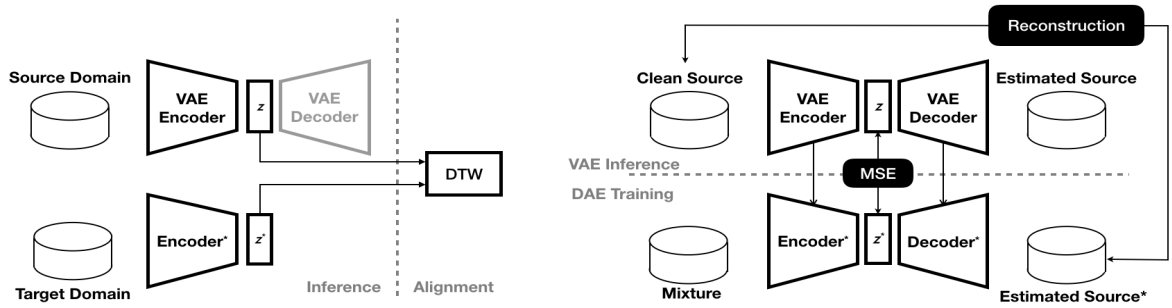

Figure 2. Left: the derivation of the proposed feature representation in *task 1* and *task 2*. Right: the training scheme of the target-domain DAE in *task 3*.

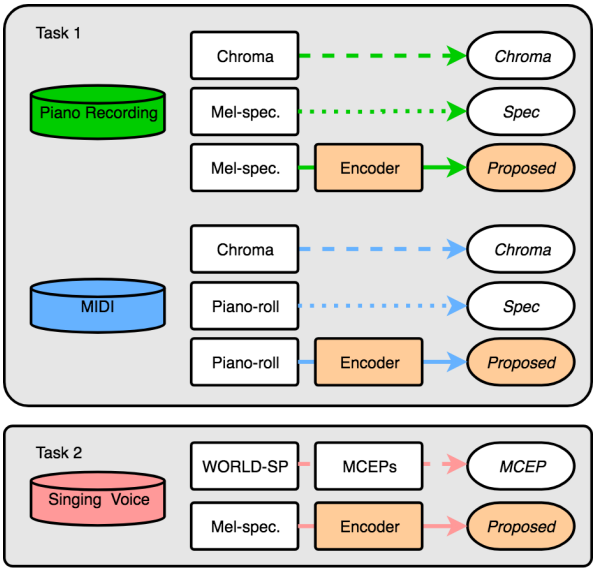

Figure 3. Extraction pipelines of feature representations for the two alignment tasks, i.e., *task 1* and *task 2*.

3.2 Task 2: Audio-to-Audio Alignment

In this experiment, we consider singing voice alignment, in particular the alignment of song recordings performed by singers with their artificially-distorted versions. Specifically, two subtasks are considered: 1) aligning the distorted monophonic singing recordings to the original version, denoted as *mono-to-mono*, and 2) the distorted monophonic singing recordings to the original singing recordings mixed with the corresponding background music, denoted as *mono-to-poly*. The goal is to demonstrate the robustness of our proposed feature representation against the artificial distortion effects, i.e, pitch-shift and time-stretch, as well as interference of the background music.

3.2.1 Dataset

We adopt the MIR-1k dataset [8] which contains the 1,000 Chinese karaoke excerpts with separated voice and accompaniment tracks, clipped from 110 songs. We then divided the 110 songs into two subsets, one containing 88 songs for training, and the other containing 22 songs for validation.

3.2.2 Model

The training procedure resembles the one mentioned in Section 3.1.2. The difference is that the source domain refers to monophonic singing, and the target domain refers to its polyphonic version; the shared information is the singing voice. Notice that, similar to Section 3.1, the synthetic dataset with artificial distortion is not used for training. The target-domain encoder for modeling polyphonic music learns not only to output features that are comparable to monophonic singing voice, but also features that are more robust to artificial distortion.

3.2.3 Data Representation

The inputs are represented the same way as the audio data described in Section 3.1.1. As suggested by the preliminary experiments, the number of filter banks is set to $F = 256$ instead of 128.

3.2.4 Evaluation

We evaluate our proposed feature representation under both time-stretching and pitch-shifting distortion. The settings of the distortion follow the one in [17].

As shown in Figure 3, for the subtask *mono-to-mono*, we first apply the artificial distortion to the monophonic singing, followed by a Z-score normalization. The Mel-scaled spectrogram is then extracted as the input to the source-domain encoder, which gives the proposed feature representation of monophonic singing after a post Z-score normalization. We align the distorted monophonic singing to the intact version. For the subtask *mono-to-poly*, we adopt the identical process to the monophonic singing. While for polyphonic singing, the target-domain encoder takes the input as the Mel-spectrogram to output the proposed feature representation. We align the distorted monophonic to the original polyphonic version. The overview of derivation of the proposed feature representation and alignment are illustrated in the left panel of Figure 2.

We compare our proposed feature representation with the 24-ordered Mel-cepstral coefficients (MCEPs) [33], a widely used features regarding speech alignment for voice conversion [20], in terms of the error measure e, as in Section 3.1.4. We use the spectral envelope which is extracted by WORLD [21] to derive MCEPs. DTW in this

task searches for the optimal alignment path according to squared Euclidean distance, as suggested by preliminary experimental results.

3.3 Task 3: Singing Voice Separation

Singing voice separation is an essential yet notoriously challenging problem in music signal processing; the goal is to separate singing voice from music mixture. We investigate the potential of domain adaptation on this problem.

3.3.1 Dataset

We again adopt the MIR-1k dataset for experiment, and split the dataset in a way identical to that in Section 3.2.1.

3.3.2 Model

The basic idea is a follow-up of the *mono-to-poly* scheme in Section 3.2: given the fact that we have obtained domain-adaptive latent representations shared across monophonic singing and its polyphonic version with accompaniment, one step further is to consider decoding the outputs of the target-domain encoder in order to reconstruct the monophonic singing voice in the target domain. Therefore, we adopt a Denoising Autoencoder (DAE) [35] in the target domain.

The training scheme of the target domain is illustrated in the right panel of Figure 2. It is important to note that, different from the vanilla DAE for source separation [5], we regularize the bottleneck layer with the learned latent representation encoded by the VAE along with the model parameters for weight initialization. The training objective function for the target domain therefore becomes:

$$\mathcal{L}_{DAE} = \mathcal{L}_{l_1}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + \alpha \mathcal{L}_{MSE}\left(\boldsymbol{z}, \boldsymbol{z}^*\right), \qquad (3)$$

where the reconstruction loss $\mathcal{L}_{l_1}$ denotes the l_1-norm; $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ are the clean source of singing voice and estimated one, respectively. α is the weight of the regularization term which is set to 1 without further investigation in this preliminary work.

3.3.3 Data Representation

For audio representation, the magnitude spectrogram instead of the Mel-scaled spectrogram is used as the input; the parameters for computation of STFT remain the same as in Section 3.1.3.

3.3.4 Evaluation

The music mixture which contains the ground-truth source of singing voice $\boldsymbol{x}$ and background music is firstly normalized with a Z-score normalization, and is represented as the magnitude spectrogram. The trained DAE then takes as the input the magnitude spectrogram, and outputs the estimated source of signing voice $\tilde{\boldsymbol{x}}$.

For evaluation, we use `mir_eval` [31] to calculate and report source-to-distortion ratio (SDR), source-to-inference ratio (SIR), and source-to-artifact (SAR) ratio together with normalized SDR (NSDR). All scores are weighted by number of frames of each song. We compare the performance among vanilla DAE with or without our proposed regularization term and weight initialization during training phase.

	error measure
Proposed ($L = 128$)	**2.48**
Proposed ($L = 12$)	4.08
Chroma	6.71
Spec	39.24

Table 2. The error measure *e* of audio-to-MIDI alignment using different feature representations.

4. RESULTS

In this section, we report the performance evaluated on the validation sets for each experiment.

4.1 Task 1: Audio-to-MIDI Alignment

Table 2 lists the median value of *e*, the alignment error measure, over the 6 audio-MIDI pairs in the validation set using four different feature representations: two of them are the proposed latent representations with dimensions $L = 128$ and 12 (*Proposed*), one is the 12-dimensional chroma (*Chroma*), and the other uses Mel-scaled spectrogram for audio and piano-roll representation for MIDI, both are 128-dimension (*Spec*). One can see our proposed domain-adaptive features outperform with both $L = 128$ and 12. This implies that the plane pitch information of MIDI domain is properly modeled in the latent representations by the source-domain encoder, and is efficiently transferred to audio domain by treating the latent representations as learning targets for the target-domain encoder.

4.2 Task 2: Audio-to-Audio Alignment

We evaluate on the validation set of 22 songs and report the alignment error measure *e* of different feature representations under the *mono-to-mono* and *mono-to-poly* subtasks, along with the artificial distortion in pitch-shift and linear/non-linear time-stretch. Figure 4 shows the median of the error measure *e* using different feature representations; the baseline feature and proposed one are denoted as *MCEP* and *Proposed*, respectively. Each individual plot shows the error measure along pitch-shift steps of -2, -1, 0, 1, and 2. The top panel and bottom panel refer to *mono-to-mono* and *mono-to-poly*, respectively. The leftmost to the fifth column correspond to linear time-stretching rates of 0.8, 0.9, 1.0, 1.1 and 1.2, respectively, while the rightmost column corresponds to the non-linear time-stretch.

The results of *momo-to-mono* in the top panel suggest that our proposed feature representation encoded by the source-domain encoder is more robust to the artificial distortion than the baseline feature. The bottom panel, which corresponds to *mono-to-poly*, shows that by transferring the latent representations from source to target domain, the target-domain encoder indeed learns to output features that are robust against both the artificial distortion and the interference of background music.

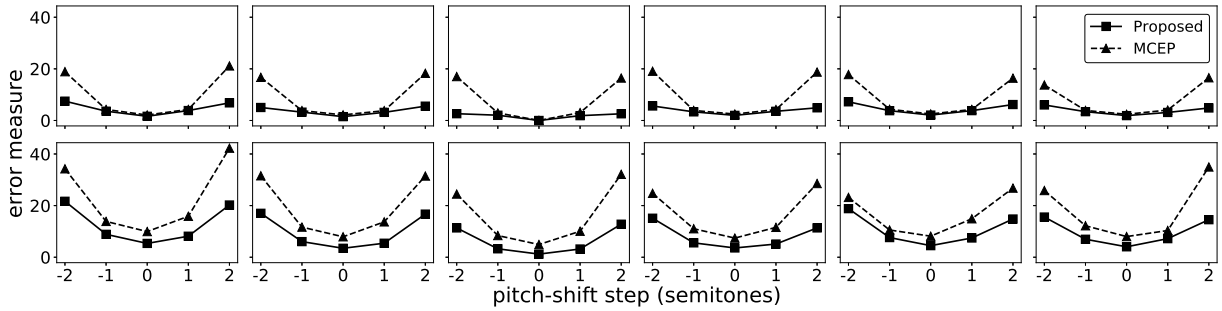

Figure 4. The error measure e of singing voice alignment using our proposed features or MCEPs. Top panel: *mono-to-mono*; bottom panel: *mono-to-poly*. The leftmost column to the fifth column refer to time-stretch rate $r = 0.8, 0.9, 1.0, 1.1,$ and 1.2, respectively; the rightmost column refers to non-linear time-stretch.

	SDR	SIR	SAR	NSDR
DAE	4.73	16.13	5.35	3.16
$DAE + wt$	6.50	**20.40**	6.85	4.93
$rDAE$	4.97	14.96	5.74	3.40
$rDAE + wt$	**7.20**	18.98	**7.74**	**5.63**

Table 3. The source-to-distortion ratio (SDR), source-to-inference ratio (SIR), and source-to-artifact ratio (SAR) and normalized SDR (NSDR) of different models.

4.3 Task 3: Singing Voice Separation

Table 3 demonstrates the SDR, SIR, and SAR together with NSDR of different models in the task of singing voice separation. Four models are compared: 1) DAE referring to the vanilla DAE, the baseline model, 2) $DAE + wt$ denoting the DAE trained with weight initialization using the source-domain model parameters, 3) $rDAE$ referring to the DAE trained with the objective function whose weight of the regularization term $\alpha = 1$ in (3), and 4) $rDAE+wt$, the DAE trained with both the weight initialization and regularization term.

From the SDR in Table 3, one can observe that $DAE + wt$ outperforms DAE by 1.77 dB, while $rDAE$ outperforms DAE by only 0.24 dB. However, by combining weight initialization and regularization together, $rDAE + wt$ achieves an improvement of 2.47 dB over DAE. This implies that the effect of transferring the latent representation from the source to target domain as a regularization term can be optimized by the transfer of the source-domain model parameters.

Notice that though the reported performance is not on par with the state-of-the-art method [10], our model still show potentials in solving the singing voice separation problem from the perspective of domain adaptation. Meanwhile, as a preliminary work, we evaluate the framework on a relatively small dataset without data augmentation and fine-tuning parameters.

5. CONCLUSION AND FUTURE WORK

In this paper, we re-investigate three well-known tasks of music signal processing from the perspective of domain adaptation, namely *task 1*: audio-to-MIDI alignment, *task 2*: audio-to-audio alignment and *task 3*: singing voice separation. To this end, we devise an unified framework that achieve both representation learning and transfer learning at once. Specifically, we use a VAE to learn latent representation of source-domain data, which is then transfered to train a separate model that maps target-domain data to the representation.

We empirically validate our idea by demonstrating the superiority of our proposed feature representations over baseline ones across all the tasks. In both *task 1* and *2*, the proposed features are shown to properly model the source-domain data and are efficiently transfered to the target domain; they are more robust against various settings of artificial distortion compared to baseline features. In *task 3*, it is shown that transferring of both model parameters and latent representations, used for weight initialization and as a regularization term, respectively, can benefit the performance of singing voice separation, which indicates the potential of the framework for such a challenging problem.

As a preliminary work, though we share most of the parameters and model architectures across all the tasks without tailoring for each individual task, the proposed framework consistently outperforms the baselines. For future work, we would like to include larger datasets and optimize the system architectures and their parameters. Moreover, expanding the framework for classification is of particular interest. For example, it is possible to transfer the latent representation from source to target domain by directly leveraging it as the classifying feature [15] or intermediate condition to models in target domain [7].

6. ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work is partially supported by MOST Taiwan, under the contract MOST 106-2218-E-001-003-MY3.

7. REFERENCES

[1] J. J. Carabias-Orti, F. J. Rodríguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Cañadas-Quesada. An Audio to Score Alignment Framework Using Spectral Factorization and Dynamic Time Warping. In *ISMIR*, pages 742–748, 2015.

[2] K. Choi, G. Fazekas, M. Sandler, and K. Cho. Transfer Learning for Music Classification and Regression Tasks. In *ISMIR*, 2017.

[3] C. Doersch. Tutorial on Variational Autoencoders. *ArXiv e-prints*, June 2016.

[4] V. Emiya, R. Badeau, and B. David. Multipitch Estimation of Piano Sounds using a New Probabilistic Spectral Smoothness Principle. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010.

[5] E. M. Grais and M. D. Plumbley. Single Channel Audio Source Separation using Convolutional Denoising Autoencoders. *ArXiv e-prints*, March 2017.

[6] P. Hamel, M. E. Davies, K. Yoshii, and M. Goto. Transfer Learning in MIR: Sharing Learned Latent Representations for Music Audio Classification and Similarity. In *ISMIR*, 2013.

[7] J. A. Hennig, A. Umakantha, and R. C. Williamson. A Classifying Variational Autoencoder with Application to Polyphonic Music Generation. *ArXiv e-prints*, November 2017.

[8] C.-L. Hsu and J.-S. R. Jang. On the Improvement of Singing Voice Separation for Monaural Recordings using MIR-1k Dataset. *TASLP*, 18(2):310–319, 2010.

[9] W.-N. Hsu, Y. Zhang, and J. Glass. Learning Latent Representations for Speech Generation and Transformation. In *Interspeech*, pages 1273–1277, 2017.

[10] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde. Singing Voice Separation With Deep U-Net Convolutional Networks. In *ISMIR*, 2017.

[11] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*, December 2014.

[12] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, December 2013.

[13] T. Kwon, D. Jeong, and J. Nam. Audio-to-score Alignment of Piano Music Using RNN-based Automatic Music Transcription. *ArXiv e-prints*, November 2017.

[14] S. Latif, R. Rana, J. Qadir, and J. Epps. Variational Autoencoders for Learning Latent Representations of Speech Emotion: A Preliminary Study. *ArXiv e-prints*, December 2017.

[15] S. Latif, R. Rana, J. Qadir, and J. Epps. Variational Autoencoders for Learning Latent Representations of Speech Emotion: A Preliminary Study. *ArXiv e-prints*, December 2017.

[16] Q. Li. Literature Survey: Domain Adaptation Algorithms for Natural Language Processing. Technical report, Department of Computer Science The Graduate Center, The City University of New York, 2012.

[17] Y.-J. Luo, M.-T. Chen, T.-S. Chi, and L. Su. Singing Voice Correction using Canonical Time Warping. *ArXiv e-prints*, November 2017.

[18] B. McFee, M. McVicar, O. Nieto, S. Balke, C. Thome, D. Liang, E. Battenberg, J. Moore, R. Bittner, R. Yamamoto, and et al. librosa 0.5.0, February 2017.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *ArXiv e-prints*, January 2013.

[20] S. H. Mohammadi and A. Kain. An Overview of Voice Conversion Systems. *Speech Communication*, 2017.

[21] M. Morise, F. Yokomori, and K. Ozawa. WORLD: A Vocoder-based High-quality Speech Synthesis System for Real-time Applications. *IEICE Trans. Information and Systems*, 99(7):1877–1884, 2016.

[22] M. Müller, F. Kurth, and M. Clausen. Audio Matching via Chroma-Based Statistical Features. In *ISMIR*, page 6th, 2005.

[23] M. Müller, F. Kurth, and T. Röder. Towards an Efficient Algorithm for Automatic Score-to-Audio Synchronization. In *ISMIR*, 2004.

[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and Transferring Mid-level Image Representations using Convolutional Neural Networks. In *CVPR*, pages 1717–1724. IEEE, 2014.

[25] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Trans. on knowledge and data engineering*, 22(10):1345–1359, 2010.

[26] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam. Representation Learning of Music Using Artist Labels. *ArXiv e-prints*, October 2017.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic Differentiation in PyTorch. In *NIPS-W*, 2017.

[28] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual Domain Adaptation: A survey of Recent Advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

[29] C. Raffel and D. P. W. Ellis. Intuitive Analysis, Creation and Manipulation of MIDI Data with pretty_midi. In *ISMIR Late Breaking and Demo Papers*, 2014.

[30] C. Raffel and D. P. W. Ellis. Optimizing DTW-based Audio-to-MIDI Alignment and Matching. In *ICASSP*, pages 81–85. IEEE, 2016.

[31] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. mir_eval: A Transparent Implementation of Common MIR Metrics. In *ISMIR*, 2014.

[32] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML*, pages 759–766. ACM, 2007.

[33] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai. Mel-generalized Cepstral Analysis-A Unified Approach to Speech Spectral Estimation. In *International Conference on Spoken Language Processing*, 1994.

[34] T. Tommasi, N. Quadrianto, B. Caputo, and C. H. Lampert. Beyond Dataset Bias: Multi-task Unaligned Shared Knowledge Transfer. In *Asian Conference on Computer Vision*, pages 1–15. Springer, 2012.

[35] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[36] F. Zhou and F. De la Torre. Generalized Time Warping for Multi-modal Alignment of Human Motion. In *CVPR*, pages 1282–1289, 2012.