LEARNING A CROSS-DOMAIN EMBEDDING SPACE OF VOCAL AND MIXED AUDIO WITH A STRUCTURE-PRESERVING TRIPLET LOSS

Keunhyoung Luke Kim[1] **Jongpil Lee**[1] **Sangeun Kum**[1] **Juhan Nam**[2]
[1] Neutune Research, Seoul, South Korea
[2] Graduate School of Culture Technology, KAIST, South Korea
{khlukekim, jongpillee, keums}@neutune.com, juhan.nam@kaist.ac.kr

ABSTRACT

Recent advances of music source separation have achieved high quality of vocal isolation from mix audio. This has paved the way for various applications in the area of music informational retrieval (MIR). In this paper, we propose a method to learn a cross-domain embedding space between isolated vocal and mixed audio for vocal-centric MIR tasks, leveraging a pre-trained music source separation model. Learning the cross-domain embedding was previously attempted with a triplet-based similarity model where vocal and mixed audio are encoded by two different convolutional neural networks. We improve the approach with a structure-preserving triplet loss that exploits not only cross-domain similarity between vocal and mixed audio but also intra-domain similarity within vocal tracks or mix tracks. We learn vocal embedding using a large-scaled dataset and evaluate it in singer identification and query-by-singer tasks. In addition, we use the vocal embedding for vocal-based music tagging and artist classification in transfer learning settings. We show that the proposed model significantly improves the previous cross-domain embedding model, particularly when the two embedding spaces from isolated vocals and mixed audio are concatenated.

1. INTRODUCTION

Vocal is the key component in popular music, as it is usually tied to the artist and melody of a song. Research reports that vocal is the most salient part in music listening experience of streaming service [1] and the most effective factor in hit song prediction [2]. A number of MIR tasks are also focused on the singing voice, for example, singer identification [3], melody extraction [4], singing transcription [5], and query-by-humming [6]. However, vocal sound sources are usually available in mixed form with instrumental sounds in popular music and isolated vocal tracks are scarcely available. This has been a barrier in singing voice research.

Recent advances of music source separation achieved a significant level of competency [7]. Pre-trained source separation models are freely available for practical applications [8]. This has opened up a great potential for various down-streaming tasks. Among others, vocal source separation found immediate uses in many relevant tasks including vocal melody extraction [9, 10], vocal tagging [11], singer identification [12, 13], and automatic drum mixing [14].

In this paper, we apply the vocal source separation to learn a cross-domain embedding space between isolated vocal and mix audio. The goal of this research is learning a discriminative vocal embedding space agnostic to mixing with instrumental sounds. This idea of cross-domain embedding space was first proposed in [15]. However, the previous work secured the isolated vocal and mix audio by a simple music mash-up, an artificial mix between two heterogeneous datasets by musical matching (i.e., tempo, beat, and key). This is not a realistic setting for obtaining a practical vocal embedding space. Furthermore, they considered only the correspondence between monophonic vocal and mixed audio in training the model using metric learning, missing vocal similarity within the same domain.

We improve the cross-domain embedding space in two ways. First, we employ a structure-preserving triplet loss that exploits not only cross-domain similarity between vocal and mixed audio but also intra-domain similarity within vocal tracks or mix tracks. Second, we conduct a large-scale cross-domain vocal embedding learning by leveraging a pre-trained vocal separation model to extract isolated vocals with high-quality. We show that the proposed method achieves significant improvements in singing identification, compared to the previous work. In addition, we evaluate the cross-domain vocal embedding for vocal tagging and artist classification in transfer learning settings and show the generalization capability.

2. RELATED WORKS

2.1 Singer Identification in Polyphonic Music

The main challenge of singer identification in polyphonic music is to extract voice features from music signals mixed with instrumental sounds. The most straightforward way to handle this issue is to separate the vocal sound sources from mixed audio as a pre-processing step. Early approaches relied on vocal melody extraction (or predominant pitch estimation) with a combination of voice re-synthesis and voice detection algorithms to obtain en-

hanced vocal sources [16–18]. They then extracted various hand-engineered features such as mel-frequency cepstral coefficient (MFCC), linear prediction mel-frequency cepstral coefficient (LPMCC) as input features for a classifier.

Recently, music source separation algorithms based on deep learning have significantly advanced [7, 19] and some of them focused more on vocal source separation [20]. This has provided a great chance to improve singer identification by allowing to use high-quality isolated vocals [13] or augmenting training data by remixing of vocals and accompaniment [12]. However, the scope of research was limited to achieving high accuracy on a small size dataset without scaling up to general vocal audio embedding.

2.2 Representation Learning

Representation learning allows unorganized input data to be mapped into an structured space [21]. Previously, this was done by a shallow network with hand-crafted features [22], and more recently deep representation learning where the deep neural networks directly learn the mapping from the raw input data became a dominant methodology. Once the representation space is structured, we can utilize it to solve the related domain problems (or downstream tasks), known as *transfer learning* [23].

Among many techniques for deep representation learning, metric learning with a triplet loss became popular because of its flexibility in training the model with less strict distance metrics [24]. This triplet loss based network is trained with three sampled examples such that the two examples are defined as similar, but not the rest one. Then, the embedding network is trained to locate the two similar examples to be closer than the dissimilar example in the representation space. This similarity-based learning can be easily extended to multi-modal data (e.g. image-to-text [25–28], video-to-text [29], face-to-voice [30], video-to-audio [31, 32]). In this case, two different embedding networks are trained for each modality. For example, if the case is to learn a cross-modal embedding of image and audio [32], an image and an audio from the same category are sampled, and an audio from the different category are sampled. Then, the embedding feature of the image and the embedding features of the audio are compared. By optimizing the embedding features of the image and audio from the same category to be placed closer than the different one, we can build cross-modal embedding space. This representation learning paradigm is close to our study. However, cross-domain embedding learning is different from the cross-modal embedding learning in that the two embedding networks are from the same modality (e.g., sketch-to-photo [33], sketch-to-3Dshape [34], street-view-to-satellite-view [35], vocal-to-mixed [15]). Because the inherent characteristics in each domain may largely differ, it is required to have separate networks for each domain.

2.3 Representation Learning in MIR

Representation learning in MIR has mainly been explored in semantic level [36–38]. Diverse similarity supervisions have been employed to train the triplet networks. Tag labels are one representative similarity supervision by regarding the two examples similar if they belongs to the same tag [39, 40]. User's preference data (similarity judgement [36] or listening history [41]) is another similarity supervision. Artist information has also been explored [37, 42]. In this case, the two examples are treated as similar if they are released from the same artist. The artist based similarity may represent some of the vocal characteristics of the artist. However, artists in some genres do not contain vocal sounds, and vocal-focused representation learning has not been extensively studied [43]. For monophonic singing voice, vocal representation learning was attempted using the DAMP dataset (amateur karaoke vocal recordings) [44]. It was extended to joint embedding between mono and mixed audio by an artificial mash-up of the karaoke recordings and instrumental tracks [15]. However, the outcomes are not directly applicable to commercial popular music.

3. METHODS

We present three training models to learn a vocal embedding space. Each model consists of multiple encoder networks for feature extraction from mixed audio or isolated vocal. We first introduce the backbone network common to all encoders and then describe the three training models.

3.1 Backbone Network

The backbone network for the encoders consists of 8 convolutional layers with 128 3-by-3 filters except the first layer with 64 filters and the last layer with 256 filters. Each convolutional layer is followed by a batch normalization, ReLU, and a 2-by-2 max-pooling layer, while the pooling layer for the last convolutional layer is a global average pooling layer. The network takes mel-spectrogram with 128 mel bins from each audio clip after applying short-time Fourier transform with 1,024 samples of Hann window and 512 samples of hop size. The input size of the CNN encoder is 129 frames, which corresponds to a 3-second-long segment at the sampling rate of 22,050 Hz.

3.2 Training Models

Figure 1 illustrates the training models based on metric learning. The goal of metric learning is to learn an embedding space where inputs from the same class are closer to each other than those from different classes. In our setting, we take either mixed audio or vocal as input, use the output of global average pooling layer in the backbone network as the embedding space, and determine if the embedded inputs belong to the same class or not using artist labels (i.e., singer labels). We build the models upon a triplet network that consists of three encoder networks. They take anchor, positive (same class as the anchor) and negative (different class as the anchor) examples as input. The triplet network is often extended to take multiple negative examples for more effective training. Following the previous works [15, 37], we used four negative examples. The

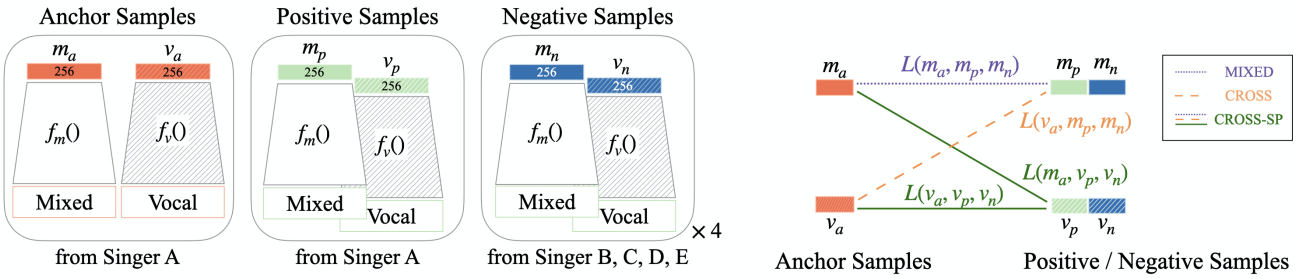

Figure 1. Left: triplet network for metric learning with singer labels. Right: distances used for the loss function in the three models. m and v represent embedding vectors from mixed audio and isolated vocal, respectively. The subscripts (a, p, n) denote for anchor, positive, and negative samples, respectively.

encoders share the weights when they take the same domain of input (vocal or mixed audio). Therefore, we eventually obtain two encoder networks with different weights: one for vocal and the other for mixed audio. In Figure 1, they are denoted as $f_v(\cdot)$ and $f_m(\cdot)$, respectively. Given the common ground, we present three models, differing in the choice of input and the loss function. Each of them is described in detail below.

3.2.1 *MIXED*

The MIXED model takes only mixed audio as input in the triplet network and thus it uses $f_m(\cdot)$ for feature extraction. This is the baseline model that takes no advantage from vocal source separation. The model was originally proposed as a general music representation learning method using the cost-free artist labels in [37]. The difference in this model is that instrumental music (with no vocals) is excluded in training the model. The triplet loss is formally defined as follows. Let x_a^i, x_p^i and x_{nj}^i denote the anchor, the positive, and the j-th negative sample of the i-th triplet, respectively, and $m_a^i = f_m(x_a^i)$ denote the embedding vector of input sample x_a^i. Following the previous works [15, 37], we use the hinge rank loss defined as below:

$$L(triplet^i) = \sum_j [M + d(m_a^i, m_p^i) - d(m_a^i, m_{nj}^i)] \quad (1)$$

where $d(x, y)$ is given as a cosine distance:

$$d(x, y) = -\frac{x \cdot y}{||x||_2 \cdot ||y||_2} \quad (2)$$

and M is the margin, which was set to 0.4.

3.2.2 *CROSS*

The CROSS model was proposed to learn the cross-domain embedding between monophonic and mixed music signals [15]. In the triplet network, the anchor takes vocal through $f_v(\cdot)$ and the positive and negative takes mixed audio through $f_m(\cdot)$. Therefore, both vocal and mixed audio from the same class (anchor and positive) are expected to be closed to each other in the embedding spaces. The triplet loss is formally defined as follows. Let y_a^i, y_p^i and y_{nj}^i denote the vocal counterpart of mixed samples x_a^i, x_p^i and x_{nj}^i and $v_a^i = f_v(y_a^i)$ denote the embedding vector for vocal. The loss function is defined as follow:

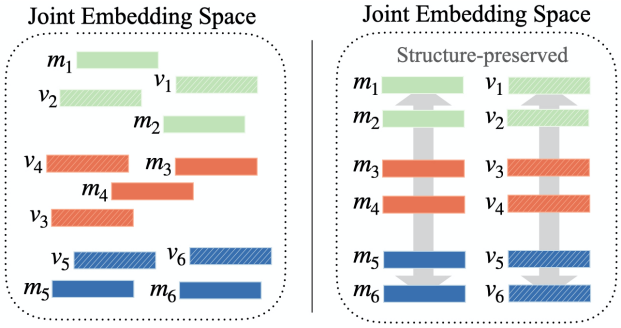

Figure 2. Illustrated examples of cross-domain embedding space when the structure-preserving triplet is not applied (left) and applied (right).

$$L(triplet^i) = \sum_j [M + d(v_a^i, m_p^i) - d(v_a^i, m_{nj}^i)] \quad (3)$$

$d(\cdot)$ is the distance defined in Equation 2. Note that the only difference from the MIXED model is the use of embedding vector from vocal source v for anchor samples.

3.2.3 *CROSS-SP*

The CROSS model effectively learns the cross-domain embedding between mixed audio and its vocal counterpart. However, this does not necessarily learn similarity within the same domain. That is, vocal examples from the same artist or mixed audio examples from the same artist are not enforced to be close to each other in the embedding space. The left column in Figure 2 illustrates a possible distribution of vocal and mixed audio examples. While the pairs of m_i and v_i are closely located by the loss function, the relations among m_i or among v_i can be arbitrary.

This issue has been addressed in the context of cross-modal representation learning where the inputs are image and text [27], or image and audio [32]. They suggested to add constraints to the triplet loss such that the similarity within the same modality is also preserved. Furthermore, they made the loss bi-directional (or symmetric) with regards to two different modalities by having a dual loss where the two triplets take exclusively different modalities of inputs. This setting can be also applied to two different audio domains in our setting: vocal and mixed audio. As a result, we expect that the embedding spaces preserve the structure of similarity in both cross-domains and intra-domain, as illustrated in the right column in Figure

2. We call this model "CROSS-SP" where SP stands for structure-preserving. The structure-preserving triplet loss is defined as a weighted sum of four separate hinge rank loss functions:

$$L(t^i) = \lambda_1 \sum_j [M + d(v_a^i, m_p^i) - d(v_a^i, m_{nj}^i)] \quad (4a)$$

$$+\lambda_2 \sum_j [M + d(m_a^i, v_p^i) - d(m_a^i, v_{nj}^i)] \quad (4b)$$

$$+\lambda_3 \sum_j [M + d(m_a^i, m_p^i) - d(m_a^i, m_{nj}^i)] \quad (4c)$$

$$+\lambda_4 \sum_j [M + d(v_a^i, v_p^i) - d(v_a^i, v_{nj}^i)] \quad (4d)$$

$d(\cdot)$ is the distance defined in Equation 2. Note that all four possible combinations of distances between anchor and positive/negative samples are present: vocal-mix, mix-vocal, mix-mix and vocal-vocal. The first two terms are the bi-directional cross-domain ranking loss [45, 46] and the last two terms are structure-preserving loss [27]. λ_n is a weight for each loss term. The MIXED model can be regarded as a special case where $\lambda_3 = 1$ and others are 0. The CROSS model is also a special case where $\lambda_1 = 1$ and others are 0. In recent studies, the structure-preserving loss terms tend to have a small weight [28] or can be modified to have weak impacts [32]. However, we used $1/4$ for all λ_n in the CROSS-SP model, because vocal and mixed audio actually share the same modality and only have different content. An extensive grid search for various combinations of λ_n is left to future research.

4. EXPERIMENTS AND RESULTS

4.1 Dataset and Training Details

We used a filtered version of Million Song Dataset (MSD) [47] for the vocal embedding learning. It contains 4,389 singers with 10 to 20 vocal songs [1] . The audio tracks were ensured to include vocal segments using a singing voice detector [48]. For each singer, we used 3 songs for validation, 2 songs for test, and the remaining 5 to 15 songs for training (the test songs were used only for internal evaluation). For vocal source separation, we used the Spleeter vocals/accompaniment separation model from Deezer [8].

To train the vocal embedding models, we used an SGD optimizer with the initial learning rate of 0.01, decay rate of $1e{-}6$ and the Nesterov momentum. We empirically chose to randomly generate 1200 batches of 25 triplets per epoch. The training is stopped when there is no decrease of validation loss for 20 epochs, and took about 100 epochs.

4.2 Task 1: Singer Identification

We first evaluate the cross-domain vocal embedding for singer identification. The task predicts the correct singer of the query audio among a list of candidate singer models.

[1] We used artist labels in MSD and assumed that artists of songs that contain vocal sounds correspond to “singers” in the experiment.

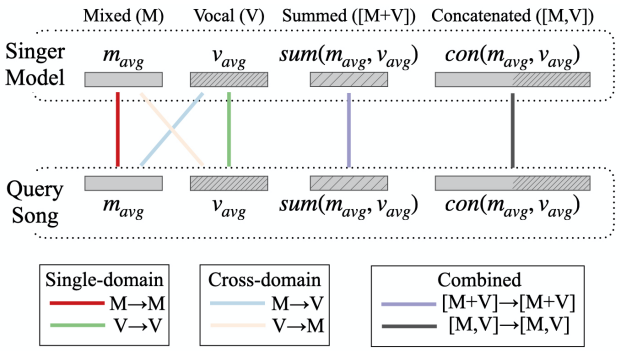

Figure 3. Test scenarios for the singer identification task.

4.2.1 *Experiment Settings*

We chose 300 singers who have 20 vocal songs from MSD, which are unseen in the training phase. We built the singer models by averaging the embedding vectors from vocal segments of 15 training songs. A query was also computed as an average of the embedding vectors from each of the remaining 5 songs, resulting in a total of 1,500 queries. Since we have two encoders to extract vocal embedding vectors ($f_m()$ and $f_v()$), singer models and queries can be formed with a different combination. We investigated the following four possibilities for evaluation:

- `M`: takes mixed audio with $f_m(\cdot)$
- `V`: takes isolated vocal with $f_v(\cdot)$ but, in the MIXED model, it takes isolated vocal with $f_m(\cdot)$
- `[M+V]`: takes both isolated vocal and mixed audio with $f_m(\cdot)$ and $f_v(\cdot)$, and computes the sum of the two embedding vectors
- `[M,V]`: takes both isolated vocal and mixed audio with $f_m(\cdot)$ and $f_v(\cdot)$, and concatenates the two embedding vectors

The entire test scenarios with the different combinations of models and queries are illustrated in Figure 3. They include not only single-domain tests where the singer models and queries are formed from the same encoders (`M`→`M`, `V`→`V`, `[M+V]`→`[M+V]`, `[M,V]`→`[M,V]`) but also cross-domain tests where the singer models and queries are formed from different encoders (`M`→`V`, `V`→`M`). We used the cosine distance between the models and queries to identify the singer.

4.2.2 *Results*

Figures 4 shows the singer identification results given the three training models in the 6 test scenarios. Each of the bar graphs shows top-1 and top-5 accuracy. In general, the CROSS-SP model significantly outperforms the CROSS and MIXED models in most test scenarios. In the single-domain querying tests (`M`→`M`, `V`→`V`), the CROSS model shows notable improvement over the MIXED model, implying that the encoder for mixed audio, $f_m(\cdot)$, becomes more discriminative when it is jointly trained with the encoder for isolated vocal, $f_v(\cdot)$. This result is not commonly observed in cross-modal embedding research. The difference is presumably attributed to the fact that we use the same modality of audio data although the domains are

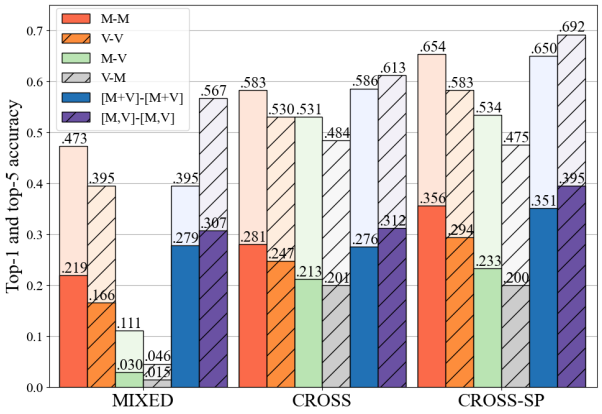

Figure 4. Singer identification result. The filled bars indicate top-1 accuracy while the blank bars indicate top-5 accuracy

Model	Space	R@5	R@10	Pr@5	Pr@10	mAP
MIXED	M	0.1014	0.1519	0.1217	0.0918	0.1176
	V	0.0499	0.0757	0.0598	0.0454	0.0589
	[M+V]	0.1015	0.1515	0.1218	0.0909	0.1200
	[M,V]	0.1139	0.1689	0.1367	0.1013	0.0713
CROSS	M	0.1218	0.1874	0.1462	0.1124	0.1523
	V	0.0971	0.1501	0.1165	0.0901	0.1178
	[M+V]	0.1215	0.1893	0.1458	0.1136	0.1499
	[M,V]	0.1375	0.2010	0.1650	0.1206	0.1672
X-SP	M	0.1617	0.2410	0.1940	0.1446	0.1979
	V	0.1079	0.1689	0.1295	0.1013	0.1350
	[M+V]	0.1625	0.2408	0.1950	0.1445	0.1974
	[M,V]	**0.1817**	**0.2651**	**0.2180**	**0.1591**	**0.2211**

Table 1. Results from the query-by-singer task.

different. The single-domain querying tests are improved further in the CROSS-SP model. This validates that the structure-preserving triplet loss improves the arrangement of similar items on both embedding spaces.

In the cross-domain querying tests (M→V, V→M), the CROSS and CROSS-SP models show little difference. This indicates that the CROSS-SP model maintains the similarity between the cross-domains despite the additional loss terms. On the other hand, the MIXED model shows poor performance. This is expected because the model was not trained to handle isolated vocals and mixed at the same time.

In the combination querying tests, we can observe an interesting result that the concatenation of the two embedding vectors consistently increases the accuracy in all models. This indicates that musical sounds other than isolated vocals provide additional information to identify singers. This make senses in that artists are often associated with a particular style or genre of music. In the meantime, the sum of the two embedding vectors did not help improving the accuracy in all models.

4.3 Task 2: Query-by-Singer

A follow-up task using the vocal embeddings is to retrieve songs with the singer information of a query song.

4.3.1 Experiment Settings

We used the same 300 singers from the singer identification task above. For each singer, we chose 6 songs to include in the dataset to be retrieved and 4 songs as queries.

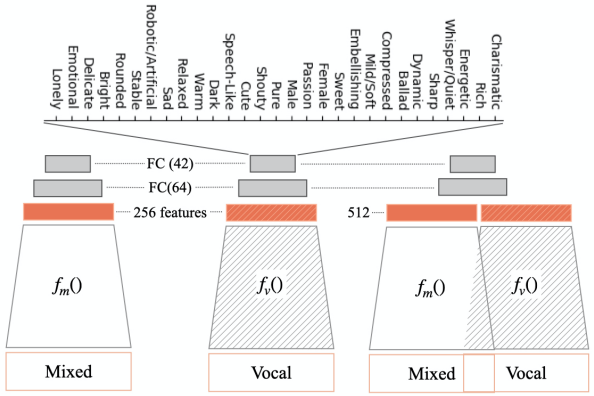

Figure 5. The model structure for the vocal tagging task using transfer learning with the vocal embedding models.

This results in 1800 songs in the search space and 1200 queries. We represented all queries and retrieved songs as an average of the vocal embedding vectors, and calculated the similarity using the cosine distance.

We evaluated the models using Recall-at-k(R@k), Precision-at-k(Pr@k) and mean average precision (mAP). R@k represents how many songs among the relevant songs are retrieved, which is songs from the same artist in this case. It is the ratio of the number of relevant songs in top-k similar songs over all relevant songs, in our case, 6. Pr@k is a metric which shows how many items are relevant among top-k similar songs. For both metrics, we used 5 and 10 for k. The last metric is mAP, which counts the rank of every relevant song. We tested mixed only embedding space (M), isolated vocal embedding space (V) and concatenated space ([M,V]) for all three strategies. For the MIXED strategy, the same mixed encoder is used for processing isolated vocal too.

4.3.2 Results

Table 1 summarizes the results with the three metrics. The general trend is similar to that in the singer identification task; the vocal embedding space from mixed audio (M) slightly outperforms that from isolated vocals (V) in all three models, and the concatenated embedding space ([M,V]) improves the performance further. Among the three models, CROSS-SP performs the best.

4.4 Task 3: Transfer Learning to Vocal Tagging

In this task, we evaluate the generalization capability of the vocal embedding models trained in the previous section for a downstream task. For this purpose, we used the K-pop Vocal Tag (KVT) dataset designed for music auto-tagging focusing on singing voice [11]. It consists of 6,787 vocal segments from K-pop music tracks. They are annotated with 42 semantic tags which describe various vocal characteristics in the categories of pitch range, timbre, playing techniques, and gender. A subset of the tag labels are shown in Figure 5. Since the vocal tags are also associated with different styles and identities of singers, we hypothesize that the pre-trained vocal embedding space will be useful for the vocal tagging task and thus the transfer learning is effective.

Models	Space	AUC	F1	Prec.	Recall
Baseline	-	0.7116	0.7198	0.6132	0.7661
MIX	M	0.7338	0.7393	0.6495	0.8013
CROSS	M	0.7336	0.7340	0.6459	0.8046
CROSS	V	0.7406	0.7392	0.6511	0.8007
CROSS	[M,V]	0.7449	0.7431	0.6531	0.8052
CROSS-SP	M	0.7376	0.7383	0.6457	0.8054
CROSS-SP	V	0.7401	0.7414	0.6575	0.7751
CROSS-SP	[M,V]	**0.7529**	**0.7469**	**0.6617**	**0.8186**

Table 2. Result from the vocal tagging task using transfer learning with the vocal embedding models.

4.4.1 Experiment Settings

Figure 5 depicts the transfer learning setting for vocal tagging. We first extract embedding vectors using two pretrained encoders. The mixed audio encoder, $f_m(\cdot)$, is obtained from all three models (MIX, CROSS and CROSS-SP) whereas the isolated vocal encoder, $f_v(\cdot)$, is from the CROSS and CROSS-SP models only. Either one or the concatenation of the embedding vectors is used as an input feature vector of a classifier that consists of two fully connected layers (64 and 42 units) with the ReLU activation. The output layer takes the sigmoid function for multi-label classification. The classifier head is trained with the binary cross-entropy loss between the sigmoid predictions and the tag labels.

We compared the transfer learning settings with a simple CNN model trained with the KVT dataset from scratch as a baseline. The baseline model is a modified version of the CNN model in the original study of the KVT dataset [11]. The main difference is that the input segment size changes from 129 frames to 107 frames to match the baseline model to the transfer learning settings.

4.4.2 Results

Table 2 summarizes the vocal tagging accuracy measured with AUC, F1 score, precision and recall. Compared to the baseline model, the training learning models show increased performances in all metrics. This indicates that the vocal embedding learned with MSD generalizes to K-pop music vocals even though if the music genre and the target task are different. In terms of input audio domain, isolated vocals is more effective than mixed audio as shown in the CROSS model (M and V) and CROSS-SP models (M and V). This is contrasted to the results in two previous tasks (singer identification and query-by-singer), presumably because the vocal tagging task requires detailed information about timbre and singing techniques which is mainly found in vocal sounds. The results also show that the concatenated embedding vector ([M,V]) further improves the tag predictions. In particular, the CROSS-SP model achieves the best performance. This result confirms that the structure-preserving triplet-loss helps generalization in learning vocal embedding.

4.5 Task 4: Transfer Learning to Artist Classification

Lastly, we conduct artist classification using the artist20 dataset [49]. The transfer learning setting is identical to the task 3 except that the last layer is the softmax unit with 20 outputs that correspond to 20 artists. Similar to [49], we report average performance from 6-fold cross validation. Table 3 compares the CROSS-SP model to the baseline [49, 50] and recent works [51, 52]. All of them used album-level train-test split, which tends to be more challenging than song-level train-test split [51]. For comparison, we evaluated the input in two duration levels; One is segment-level (3 seconds) and the other is song-level. The song-level prediction was obtained by averaging the softmax activations from segment-level inputs through each song. The results show that the CROSS-SP model outperforms all previous works. In segment-level evaluation, the summed ([M+V]) and concatenated ([M,V]) embedding spaces are better than individual embedding spaces. In song-level evaluation, on the other hand, using only vocal embedding space (V) is better. This contradictory result can be explained by the consistency of identifiable information in vocal embedding space, which can restrain noisier information from background music.

Models	Space	Eval. Level	Accuracy	F1
GMM [49]		Frame	0.590	
GMM [50]*		Frame	0.541	
CRNN [51]		Segment		0.527
CROSS-SP	M	Segment	0.596	0.550
CROSS-SP	V	Segment	0.500	0.496
CROSS-SP	[M+V]	Segment	**0.638**	**0.587**
CROSS-SP	[M,V]	Segment	**0.638**	0.576
SVM [50]*		Song	0.687	
CRNN [51]		Song		0.653
i-Vector [52]		Song	0.8545	0.8459
CROSS-SP	M	Song	0.772	0.753
CROSS-SP	V	Song	**0.894**	**0.891**
CROSS-SP	[M+V]	Song	0.815	0.804
CROSS-SP	[M,V]	Song	0.806	0.795

Table 3. Comparison of artist classification using the artist20 dataset (* [50] used 18 artists.)

5. CONCLUSIONS

We presented a method to learn cross-domain embedding spaces between isolated vocals and mixed audio by leveraging the state-of-the-art music source separation algorithm. We show that the structure-preserving triplet-loss used in training the deep neural networks greatly improves the generalization capability when the embedding vectors are used for singer identification, query-by-singer and vocal tagging. Also, we showed the concatenation of the two vocal embedding vectors from isolated vocals and mixed audio are more effective in the tasks. This indicates that unique genres or styles of artists are reflected on their instrumental sounds and thus mixed audio is complementary to vocal sounds. We expect to extend the use of cross-domain vocal embedding to various music applications such as singer-focused music recommendation, vocal-to-music cross retrieval and other vocal-centric MIR tasks. We demonstrate an example at this link as a potential use case [2] .

[2] We implemented a vocal-to-accompaniment matching system using the cross-domain vocal embedding vector. The demo audio examples are found at https://khlukekim.github.io/crossdomainembedding/

6. REFERENCES

[1] A. Demetriou, A. Jansson, A. Kumar, and R. M. Bittner, "Vocals in music matter: the relevance of vocals in the minds of listeners," in *Proc. ISMIR*, 2018.

[2] E. Zangerle, M. Vötter, R. Huber, and Y.-H. Yang, "Hit song prediction: Leveraging low-and high-level audio features." in *Proc. ISMIR*, 2019, pp. 319–326.

[3] Y. Kim and B. Whitman, "Singer identification in popular music recordings using voice coding features," in *Proc. ISMIR*, 2002.

[4] J. Salamon, E. Gómez, D. P. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications, and challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 2, pp. 118–134, 2014.

[5] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii, "Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 161–165.

[6] J. Song, S. Y. Bae, and K. Yoon, "Mid-level music melody representation of polyphonic audio for query-by-humming system," in *Proc. ISMIR*, 2002.

[7] F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2018, pp. 293–305.

[8] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020.

[9] S. Kum, J.-H. Lin, L. Su, and J. Nam, "Semi-supervised learning using teacher-student models for vocal melody extraction," in *Proc. ISMIR*, 2020.

[10] Y. Gao, X. Zhang, and W. Li, "Vocal melody extraction via hrnet-based singing voice separation and encoder-decoder-based f0 estimation," *Electronics*, vol. 10, no. 3, p. 298, 2021.

[11] K. L. Kim, J. Lee, S. Kum, C. L. Park, and J. Nam, "Semantic tagging of singing voices in popular music recordings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1656–1668, 2020.

[12] T.-H. Hsieh, K.-H. Cheng, Z.-C. Fan, Y.-C. Yang, and Y.-H. Yang, "Addressing the confounds of accompaniments in singer identification," in *Proc. ICASSP*, 2020, pp. 1–5.

[13] B. Sharma, R. K. Das, and H. Li, "On the importance of audio-source separation for singer identification in polyphonic music." in *Proc. INTERSPEECH*, 2019, pp. 2020–2024.

[14] M. Martinez Ramirez, D. Stoller, and D. Moffat, "A deep learning approach to intelligent drum mixing with the wave-u-net." Audio Engineering Society, 2021.

[15] K. Lee and J. Nam, "Learning a joint embedding space of monophonic and mixed music signals for singing voice for singing voice," in *Proc. ISMIR*, 2019.

[16] A. Mesaros, T. Virtanen, and A. Klapuri, "Singer identification in polyphonic music using vocal separation and pattern recognition methods," in *Proc. ISMIR*, 2007, pp. 375–378.

[17] H. Fujihara, M. Goto, T. Kitahara, and H. G. Okuno, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 638–648, 2010.

[18] M. Lagrange, A. Ozerov, and E. Vincent, "Robust singer identification in polyphonic music using melody enhancement and uncertainty-based learning," in *Proc. ISMIR*, 2012.

[19] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix-a reference implementation for music source separation," *Journal of Open Source Software*, 2019.

[20] A. Jansson, E. J. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *Proc. ISMIR*, 2017.

[21] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[22] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. ICML workshop on unsupervised and transfer learning*, 2012, pp. 17–36.

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, vol. 27, 2014.

[24] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International workshop on similarity-based pattern recognition*. Springer, 2015, pp. 84–92.

[25] N. C. Mithun, R. Panda, E. E. Papalexakis, and A. K. Roy-Chowdhury, "Webly supervised joint embedding for cross-modal image-text retrieval," in *Proc. ACM*

International Conference on Multimedia, 2018, pp. 1856–1864.

[26] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, "Learning cross-modal embeddings for cooking recipes and food images," in *Proc. CVPR*, 2017, pp. 3068–3076.

[27] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proc. CVPR*, 2016, pp. 5005–5013.

[28] L. Wang, Y. Li, J. Huang, and S. Lazebnik, "Learning two-branch neural networks for image-text matching tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 394–407, 2019.

[29] N. C. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury, "Learning joint embedding with multimodal cues for cross-modal video-text retrieval," in *Proc. ACM on International Conference on Multimedia Retrieval*, 2018, p. 19–27.

[30] A. Nagrani, S. Albanie, and A. Zisserman, "Learnable pins: Cross-modal embeddings for person identity," in *Proc. ECCV*, 2018.

[31] D. Surís, A. Duarte, A. Salvador, J. Torres, and X. Giró-i Nieto, "Cross-modal embeddings for video and audio retrieval," in *Proc. ECCV Workshops*, 2019, pp. 711–716.

[32] S. Hong, W. Im, and H. S. Yang, "Cbvmr: Content-based video-music retrieval using soft intra-modal structure constraint," in *Proc. ACM on International Conference on Multimedia Retrieval*, 2018, p. 353–361.

[33] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse, "Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network," *Computer Vision and Image Understanding*, vol. 164, pp. 27–37, 2017.

[34] F. Wang, L. Kang, and Y. Li, "Sketch-based 3d shape retrieval using convolutional neural networks," in *Proc. CVPR*, 2015, pp. 1875–1883.

[35] N. Khurshid, T. Hanif, M. Tharani, and M. Taj, "Cross-view image retrieval-ground to aerial image retrieval through deep learning," in *International Conference on Neural Information Processing*. Springer, 2019, pp. 210–221.

[36] R. Lu, K. Wu, Z. Duan, and C. Zhang, "Deep ranking: Triplet matchnet for music metric learning," in *Proc. ICASSP*, 2017, pp. 121–125.

[37] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," in *Proc. ISMIR*, 2018.

[38] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Metric learning vs classification for disentangled music representation learning," in *Proc. ISMIR*, 2020.

[39] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot learning for audio-based music classification and tagging," in *Proc. ISMIR*, 2019.

[40] M. Won, S. Oramas, O. Nieto, F. Gouyon, and X. Serra, "Multimodal metric learning for tag-based music retrieval," in *Proc. ICASSP*, 2020.

[41] B. McFee, L. Barrington, and G. R. Lanckriet, "Learning similarity from collaborative filters." in *Proc. ISMIR*, 2010, pp. 345–350.

[42] J. Lee, J. Park, and J. Nam, "Representation learning of music using artist, album, and track information," in *Machine Learning for Music Discovery Workshop, ICML*, 2019.

[43] J. Park, D. Kim, J. Lee, S. Kum, and J. Nam, "A hybrid of deep audio feature and i-vector for artist recognition," in *Joint Workshop on Machine Learning for Music, ICML*, 2018.

[44] C.-i. Wang and G. Tzanetakis, "Singing style investigation by residual siamese convolutional neural networks," in *Proc. ICASSP*, 2018, pp. 116–120.

[45] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *Proc. ICML*, vol. 32, no. 2, 2014, pp. 595–603.

[46] A. Karpathy, A. Joulin, and L. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," in *Proc. NIPS*, 2014, p. 1889–1897.

[47] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. ISMIR*, 2011.

[48] S. Kum and J. Nam, "Joint detection and classification of singing voice melody using convolutional recurrent neural networks," *Applied Sciences*, vol. 9, no. 7, p. 1324, 2019.

[49] D. P. Ellis, "Classifying music audio with timbral and chroma features," 2007.

[50] M. I. Mandel and D. P. Ellis, "Song-level features and support vector machines for music classification," 2005.

[51] Z. Nasrullah and Y. Zhao, "Music artist classification with convolutional recurrent neural networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[52] H. Eghbal-Zadeh, B. Lehner, M. Schedl, and G. Widmer, "I-vectors for timbre-based music similarity and music artist classification." in *Proc. ISMIR*, 2015, pp. 554–560.