**MetaSID: Singer Identification with Domain Adaptation for Metaverse**

Xulong Zhang, Jianzong Wang*, Ning Cheng, Jing Xiao  
*Ping An Technology (Shenzhen) Co., Ltd., China*

Abstract—Metaverse has stretched the real world into unlimited space. There will be more live concerts in Metaverse. The task of singer identification is to identify the song belongs to which singer. However, there has been a tough problem in singer identification, which is the different live effects. The studio version is different from the live version, the data distribution of the training set and the test set are different, and the performance of the classifier decreases. This paper proposes the use of the domain adaptation method to solve the live effect in singer identification. Three methods of domain adaptation combined with Convolutional Recurrent Neural Network (CRNN) are designed, which are Maximum Mean Discrepancy (MMD), gradient reversal (Revgrad), and Contrastive Adaptation Network (CAN). MMD is a distance-based method, which adds domain loss. Revgrad is based on the idea that learned features can represent different domain samples. CAN is based on class adaptation, it takes into account the correspondence between the categories of the source domain and target domain. Experimental results on the public dataset of Artist20 show that CRNN-MMD leads to an improvement over the baseline CRNN by 0.14. The CRNN-RevGrad outperforms the baseline by 0.21. The CRNN-CAN achieved state of the art with the F1 measure value of 0.83 on album split.

Index Terms—Metaverse, Singer identification, Domain adaptation, CNN, GRU

I. INTRODUCTION

The Metaverse [1] breaks the limitations of the physical boundaries of concert venues, it can bring tens of millions of viewers to an audio-visual feast. Singer identification (SID), also known as artist classification, is an interesting task under the subject of music information retrieval (MIR) [2], [3]. It aims to name the singer in a given audio sample to facilitate users to find and manage the music library. If trained properly, the feature representation of the singing voices will also be learned by the SID model, which can be used in the vocal related tasks, such as similarity singer search, playlist generation, or singing synthesis [4]–[11].

The studies on SID can be categorized into three aspects, there are feature engineering, traditional classifier, and the deep model. In [12], Ellis et al. investigate fusion timbral and Chroma features for artist classification. Eghbal-zadeh et al. [13] proposed an approach to extract song-level descriptors built from frame-level timbral features. The biggest limitation of feature engineering is that it is difficult to find a feature that accurately describes different singing voices [14]. In addition to the feature representation, most research focus on the classifier. Different classifiers have been tried on singer identification, including SVM, GMM, HMM, and random forest [2], [15]–[17]. With the successful application of deep models in various tasks [5], [18], some studies are using deep models to improve performance on singer identification, such as CRNN [19] which is a state of the art method.

SID is a so far unsolved task, there is at least one challenge how to deal with the mismatch among the different styles of singing. The songs of existing studio songs are easy to obtain, and the accuracy of identifying the songs in the existing album can be very high. However, when the model is applied to the songs from the new style album or live performance version, it will bring a great performance degradation [20]. The singer singing in the live performance will have style changed slightly, and even sing other singers sing without being collected by his album. The sing styles of songs included in different albums are usually distinct, and the styles of songs in the same album are often similar. This dissimilarity is called the album effect [3], [21] in the task of singer identification.

*Corresponding author: Jianzong Wang (jzwang@188.com).*
same as the target domain with similar distributions. There are several methods to make the distribution shift between the domains of source and target. The most commonly used methods are maximum mean discrepancy (MMD) [28]–[33], [34], which make a comparison and reduction for the distribution shift. Such as the related work by Tzeng et al. [35], a deep domain confusion network (DDC) was proposed. The DDC used two CNNs with shared weights for the modeling of the source domain and target domain. The whole network is used for the task of classification with the optimization of classification loss, while the MMD metric is applied as an adaptation layer for the measuring of the different domains.

In this paper, we draw on the idea of DDC and the fusion of the structure for singer identification. Domain adaptation is fused with CRNN [6], [36], [37] to solve the problem of the album effect. These three methods of domain adaptation were proposed for the task of singer identification. Finally, the proposed three methods are the distance-based method CRNN-MMD, the gradient reversal method CRNN-RevGrad, and the clustering distance-based CRNN-CAN.

II. RELATED WORKS

A. Baseline Method of CRNN for SID

![Fig. 2. The model architecture of baseline method of CRNN for SID](image)

The state-of-the-art method of SID, CRNN [19], was chosen as the baseline. As shown in Figure 2, the model mainly consists of four CNN blocks and two GRU blocks. Then connected to a fully connected layer and a softmax layer for classification. CRNN combines the advantages of CNN and RNN. Choi et al. [38] have proved that CNN could be used as a feature extractor for MIR tasks. At the same time, due to the characteristic that the audio is continuous along the time axis, the recurrent neural network (RNN) has also achieved great success in audio tasks such as speech recognition [12], [39]. The average F1 score on the album-split Artist20 dataset is 0.603, which is a significant improvement compared to the previous model.

B. Maximum Mean Discrepancy (MMD)

The MMD is based on the fact that, if the distribution of the source domain is identical with the target domain, the related statistic parameters should be similar or same. In MMD, from the marginal distributions $P(X^s)$ and $Q(X^t)$ sampled $x^s_i$ and $x^t_i$ respectively, the data sampling is i.i.d. For the samples, MMD conduct a test of the two-sample by a kernel to determine whether satisfy the hypothesis $P = Q$ or not. Formally, MMD defines the difference between two domains with their feature vector in the reproducing kernel Hilbert space (RKHS), i.e.

$$D_H(P, Q) \triangleq \sup_{f \sim \mathcal{H}} \left( \mathbb{E}_{X \sim P} [f(X^s)] - \mathbb{E}_{X' \sim Q} [f(X^t)] \right)$$

where $\mathcal{H}$ refers to the set of all functions $f$ that map the feature space to the real number set $R$. In fact, the empirical kernel mean embedding estimated the MMD in form of squared value, it build the layer $l$,

$$\hat{D}^{mmd}_l = \frac{1}{n^s} \sum_{i=1}^{n^s} \sum_{j=1}^{n^s} k_l(x^s_i, x^s_j) + \frac{1}{n^t} \sum_{i=1}^{n^t} \sum_{j=1}^{n^t} k_l(x^t_i, x^t_j) - \frac{2}{n^sn^t} \sum_{i=1}^{n^s} \sum_{j=1}^{n^t} k_l(x^s_i, x^t_j)$$

where $x^s \in S', x^t \in T'$.

C. Gradient Reversal (RevGrad)

The basic idea of RevGrad [40] is to use a similar training method with GAN to let the generator generate features, and then let the discriminator determine whether it is a feature of the source domain or the target domain. If the discriminator cannot distinguish which domain it is, it means that the source domain and target domain are consistent in the learned feature space.

Consider the classification task with input space of $X$, and $Y = 0, \ldots, N$ is the possible labels of the predicted class. There are at least two domains over the data of $(X, Y)$, where we call the known label domain as source $D_s$, and the other domain as target $D_t$. Formally express as Equation 3 and Equation 4.

$$S = \{(x_i, y_i)\}_{i=1}^n \sim (D_s)^n$$

$$T = \{x_i\}_{i=n+1}^N \sim (D_t^{X'})^n$$

where $D_t^{X'}$ is the marginal distribution of $D_t$ over $X$. With the total $n + n'$ samples to learn a classifier for $X \rightarrow Y$ without information label of $D_t$.

The optimization process can use the minimization-maximization idea to train. The optimization algorithm of the model is carried out in the following three steps.

- Train the source feature extractor $M_S$ and classifier $C$ on the source domain.
- Fix $M_S$ and $C$, train the target feature extractor $M_T$ and domain discriminator $D$.
- Use $M_T$ and $C$ to classify the target domain.
for the deep neural network, $\phi$ is calculated as the discrepancy between conditional data distributions across domains. The Discrepancy (CDD) in CAN is established on the difference of the information of both the source domain and target domain, very well.

Due to the existence of many suboptimal classifications near the decision boundary, it could lead to an overfit for the source domain dataset but could not classify the target domain classes. The model was classified well in the source classes but had a poor ability to generalization for the classes in the target domain.

While the CAN [41], [42] take into account the class information of both the source domain and target domain, it also measures the same class across the domains and the different class across the domains. The Contrastive Domain Discrepancy (CDD) in CAN is established on the difference between conditional data distributions across domains. The CDD is calculated as

$$D^{\text{add}} = \frac{1}{M} \sum_{c=1}^{M} D^{cc}(y_{1:n_t}, \phi) - \frac{1}{M(M-1)} \sum_{c=1}^{M} \sum_{c'=1\&c'\neq c}^{M} D^{cc'}(y_{1:n_t}, \phi),$$

where $M$ is the classes number, $y_t^i$ is the target data label, for the deep neural network, $\phi$ is used as the mapping from the input to a following layer. The data samples number of a batch from target domain is $n_t$. The discrepancy of intra-class across domains is $D^{cc}$, and the discrepancy of inter-class across domains is $D^{cc'}$. In the optimization, it minimize the inter-class domain discrepancies and maximize the intra-class domain discrepancies.

### III. Methodology

Combined with the deep model CRNN, we proposed three singer identification methods based on domain adaptation methods.

#### A. CRNN-MMD

Given two distributions $s$ and $t$, the MMD is defined as follows:

$$MMD^2(s, t) = \sup_{\|\phi\|_{\mathcal{H}} \leq 1} \left\| E_{x^s \sim s} [\phi(x^s)] - E_{x^t \sim t} [\phi(x^t)] \right\|_{\mathcal{H}}^2$$

where $\phi$ represents the kernel function that maps the original data to a reproducing kernel Hilbert space (RKHS) and $\|\phi\|_{\mathcal{H}} \leq 1$ defines a set of functions in the unit ball of RKHS $\mathcal{H}$.

As shown in Figure 4(a), we use a single hidden layer in the feedforward neural networks for the calculation of the MMD metric. From the different domains’ representation of the embedding in the latent space, to reduce the mismatch of different distributions can calculate the metric of MMD. We use a kernel two-sample test for the effective metric of the MMD. The estimation of the MMD is given as below:

$$MMD^2(D_s, D_t) = \left\| \frac{1}{N} \sum_{i=1}^{N} \phi(x^s_i) - \frac{1}{M} \sum_{j=1}^{M} \phi(x^t_j) \right\|_{\mathcal{H}}^2$$

#### B. CRNN-RevGrad

The CRNN-RevGrad shown in Figure 4(b) is to add a gradient reversal to the CRNN and Dense layer. The feature extractor consists of 4 CNN blocks and 2 GRU blocks same as baseline CRNN. The class label is used for the classifier that classifies the features of the source domain data. The Domain label is a discriminator that distinguishes the characteristics of the source domain data and the target domain data. This discriminator should be continuously enhanced so that it can distinguish the characteristics of the source domain or the target domain. At the same time, the extractor should be enhanced so that the extracted features can confuse the discrimination of the discriminator. In this way, the finally extracted features are consistent in the space of the source domain and target domain.

The model of CRNN-RevGrad to keep the feature distribution across the source domain and target domain integrates a gradient reversal into the standard architecture. The network is built up of several layers for feature extraction with shared weights and two classifiers. The optimization of CRNN-RevGrad, minimizes the domain confusion loss for all samples from the source domain and target domain, and it minimizes the classification loss with the labels from the source domain dataset, while for the target domain data samples it maximizes domain confusion loss with the use of gradient reversal.

CRNN-RevGrad minimizes the distance of the representation of source domain and target domain through the function as below:

$$\min_{M^*, C} L_{cls} (X^s, Y^s) = E_{(x^*, y^*) \sim (X^*, Y^*)} \sum_{k=1}^{K} 1[k=y^*] \log C (M^* (x^*))$$

![Fig. 3. Comparison between previous domain-discrepancy minimization methods with class-discrepancy minimization methods](image)
CRNN-MMD

(a) CRNN-MMD

(b) CRNN-RevGrad

(c) CRNN-CAN

Fig. 4. The network architecture of the proposed three methods

\[
\min_D \mathcal{L}_{adv:D} \left(X^s, X^t, M^s, M^t\right) =
\]

\[
- E_{(x^t)\sim (X^t)} \left[ \log D \left( M^t \left( x^t \right) \right) \right]
\]

\[
+ E_{(x^t)\sim (X^t)} \left[ \log \left( 1 - D \left( M^t \left( x^t \right) \right) \right) \right]
\]

where \( M^s \) and \( M^t \) is the representation of the mappings learned from the source domain samples \( X^s \) and target domain samples \( X^t \). The classifier \( C \) is used for the source domain. Through the supervised data from the source domain, the classifier could be trained with the classification loss function of \( L_{cls} \). There are also two loss functions of \( L_{adv:D} \) and \( L_{adv:M} \). \( L_{adv:D} \) is used for the discriminator, and it need to be minimized. \( L_{adv:M} \) is to learn a domain invariant representation, and it need to be maximized.

\[
\min_{M^s, M^t} \mathcal{L}_{adv:M} \left( M^s, M^t \right) =
\]

\[
- E_{(x^t)\sim (X^t)} \left[ \log D \left( M^t \left( x^t \right) \right) \right]
\]

\[
+ \sum_{i=1}^{n_s} \log P(y_i^s | x_i^s) \frac{1}{n_s}
\]

\[
\text{loss}^{ce} = \sum_{i=1}^{n_s} \log P(y_i^s | x_i^s)
\]

\[
\text{loss}_{D_{adv:M}} = \frac{1}{n_s} \sum_{i=1}^{n_s} \log P(y_i^s | x_i^s)
\]

\[
\text{loss}^{ce} + \frac{1}{n_s} \sum_{i=1}^{n_s} \log P(y_i^s | x_i^s)
\]

where \( l \) is the FC layer, and \( L \) is the total number of FC layers. The network was trained with labeled sources data through minimizing the cross-entropy loss.

\[
\text{loss}^{ce} = \sum_{i=1}^{n_s} \log P(y_i^s | x_i^s)
\]

where the discrepancy penalty term is multiply with the weight factor of \( \beta \).

To estimate and optimize with CDD, we need labels of the target domain. However, target labels are unknown. Additionally, during the training in a batch, for class \( C \), there are only contain samples in one domain. To solve the two problems, we adapt two tricks during the training phase. We use CAN alternatively to estimate the label hypothesis of target samples through clustering. After clustering, when estimating CDD, fuzzy target data far from the cluster center is set to zero, and fuzzy classes with few target samples around the cluster center are set to zero. Moreover, to satisfy the training in the manner of mini-batch size, both source and target domains were sampled for each batch, i.e. we randomly sample data from the source domain and target domain for each class.

IV. EXPERIMENTS

A. Dataset and Audio Processing

Artist20, a music artist classification dataset published by Ellis [12], was used in this paper to evaluate classification performance.

The Artist20 dataset is the most widely used dataset for artist classification. It consists of 20 artists, each artist has 6 albums, and a total of 1413 songs. For the artist classification, there are different dataset split methods were used in related works. In addition to ensuring that there is no same frame was...
used both in the test song and train song, it is important to take into account the producer effect found by Whitman et al. [43] in previous research. This refers to inflated classification performance in datasets split by song because of how salient production details can be in comparison to musical style. To combat this, the standard approach is to split the dataset by album such that the test set is composed solely of songs from albums not used in training. We divided four of the six albums into training sets, and the remaining two were divided into validation sets and test sets. We also do a preprocess on the data, cutting each song according to a window length of 30 seconds and a sliding time of 10 seconds [44], [45]. The models are evaluated using Artist20 under the album split, averaging the F1 scores of three independent runs.

B. Experimental Results

In order to evaluate the performance of different algorithms, the metrics of F1 value are calculated based on precision and recall to evaluate them as a whole [46]. F1 score is the harmonic average of precision and recall, which can reflect the performance of the classification model more objectively.

Table I shows the experimental results, we compared the proposed three-domain adaptation-based methods with the baseline method of CRNN proposed by Nasrullah [19]. The four models share the same CRNN architecture of four CNN blocks and two GRU blocks. And the three proposed methods start with the pre-trained model weights of the CRNN.

| Model        | F1/best | F1/avg |
|--------------|---------|--------|
| CRNN [19]    | 0.61    | 0.60   |
| CRNN-MMD     | 0.76    | 0.74   |
| CRNN-RevGrad | 0.82    | 0.81   |
| CRNN-CAN     | 0.85    | 0.83   |

From the comparison results, the baseline method CRNN is state of the art, while it could achieve only 0.61 on the Artist20 dataset with album split. We use the CRNN-MMD can get an improvement of 0.15 than the baseline CRNN. The improvement of the proposed three methods shows the validity of the domain adaption.

The CRNN-RevGrad and the CRNN-CAN get 0.82 and 0.85 on the F1 score separately. CRNN-RevGrad compare with the CRNN-MMD shows that a unified and robust feature representation between the source domain and target domain is more effective than simply minimizing the distance between domains.

The CRNN-RevGrad through the weight splitting considers the independent source and target mappings for the feature representation. Due to the parameters being from the pre-trained model of the source domain data, it is more flexible for the domain-specific feature extractions.

Finally, the CRNN-CAN compares with the CRNN-RevGrad that adding the correspondence between categories in different domains enhances the precise expression ability of domain adaptation.

C. Visualization

The last dense layer of the network processes a fully connected layer. And the audio sample is converted to a vector by the last dense layer. These vectors represented the feature learned by the network to distinguish the singer’s label. Using t-Distributed Stochastic Neighbor Embedding (t-SNE) can further reduce the dimensional to visualize the classification by the learned feature. Figure 5(a) is the result of the training set with CRNN-CAN. It shows that on the source domain the sample is well separated into different clusters. While Figure 5(b) is the result on the target domain in which are different albums song of the test set. The learned feature representation can be distinguished well in the source domain but is not effective in the target domain. Figure 5(c) is the t-SNE result diagram of the CRNN-CAN model. The yellow dots in the figure are the source domain mapping diagrams, and the blue dots are the target domain mapping diagrams. It can be seen from the figure that the two domains have basically overlapped, and there is still room for improvement.

V. Conclusions

In this paper, we introduced the domain adaption method to address the album effect on singer identification task for the
coming Metaverse. Three different methods were tried and combining with the backbone network of CRNN. The CRNN-MMD adds a dense layer to calculate the domain loss for the domain adaption. The CRNN-RevGrad adds a gradient reversal layer to try to learn a feature representation effective for both source domain and target domain. While the CRNN-CAN adds the class information of the target domain by a clustering process between source data and target data. Experiments on the public dataset of Artist20 show that domain adaption can achieve a better result than the baseline method of CRNN. Finally, the proposed method of CRNN-CAN gets 0.85 in terms of F1 measure with the album split on Artist20.

VI. ACKNOWLEDGEMENT

This paper is supported by the Key Research and Development Program of Guangdong Province under grant No.2021B0101400003. Corresponding author is Jianzong Wang from Ping An Technology (Shenzhen) Co., Ltd (jzwang@188.com).

REFERENCES

[1] L.-H. Lee, T. Braud, P. Zhou, L. Wang, D. Xu, Z. Lin, A. Kumar, C. Bermejo, and P. Hui, “All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda,” arXiv preprint arXiv:2110.05352, 2021.

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

[3] X. Zhang, Y. Yu, Y. Gao, X. Chen, and W. Li, “Research on singing voice detection based on a long-term recurrent convolutional network with vocal separation and temporal smoothing.” Electronics, vol. 9, no. 9, p. 1458, 9 2020.

[4] 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 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 1–5.

[5] Y. Sun, X. Zhang, X. Chen, Y. Yu, and W. Li, “Investigation of singing voice separation for singing voice detection in polyphonic music,” in Proceedings of the 9th Conference on Sound and Music Technology (CSMT2021), 2021, pp. 1–6.

[6] K. Lee and J. Nam, “Learning a joint embedding space of monophonic and mixed musical signals for singing voice,” in 20th International Society for Music Information Retrieval Conference, 2019.

[7] S. Si, J. Wang, J. Peng, and J. Xiao, “Towards speaker age estimation with label distribution learning,” in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 4618–4622.

[8] Z. Chen, X. Zhang, J. Deng, J. Li, Y. Jiang, and W. Li, “A practical singing voice detection system based on gru-mu,” in Proceedings of the 6th Conference on Sound and Music Technology (CSMT2018). Springer, 2019, pp. 15–25.

[9] J.-Y. Liu, Y.-H. Chen, Y.-C. Yeh, and Y.-H. Yang, “Score and lyrics-free singing voice generation,” arXiv preprint arXiv:1912.11774, 2019.

[10] E. J. Humphrey, S. Reddy, P. Seetharaman, A. Kumar, R. M. Bittner, A. Demetriou, S. Gulati, A. Jansson, T. Jehan, B. Lehner et al., “An introduction to signal processing for singing-vocalism analysis: High notes in the effort to automate the understanding of vocals in music,” IEEE Signal Processing Magazine, vol. 36, no. 1, pp. 82–94, 2018.

[11] X. Zhang, S. Li, Z. Li, S. Chen, Y. Gao, and W. Li, “Singing voice detection using multi-feature deep fusion with cnn,” in Proceedings of the 7th Conference on Sound and Music Technology (CSMT2019). Springer, 2020, pp. 41–52.

[12] D. P. Ellis, “Classifying music audio with timbral and chroma features,” in Proceedings of the 8th International Conference on Music Information Retrieval, 2007, pp. 339–340.

[13] H. Eghbal-Zadeh, B. Lehner, M. Schell, and G. Widmer, “I-vectors for timbre-based music similarity and music artist classification,” in Proceedings of the 16th International Society for Music Information Retrieval Conference, 2015, pp. 554–560.

[14] L. Wang, H. Zhu, X. Zhang, S. Li, and W. Li, “Transfer learning for music classification and regression tasks using artist tags,” in Proceedings of the 7th Conference on Sound and Music Technology (CSMT2019). Springer, 2020, pp. 81–89.

[15] X. Zhang, J. Qian, Y. Yu, Y. Sun, and W. Li, “Singer identification using deep timbre feature learning with knn-net,” in 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2021). IEEE, 2021, pp. 3380–3384.

[16] R. Sangeetha and N. Balini, “Singer identification using mfcc and crp features with support vector machines,” in Computational Intelligence in Pattern Recognition. Springer, 2020, pp. 295–306.

[17] A. Sun, J. Wang, N. Cheng, M. Tantrawenith, Z. Wu, H. Meng, E. Xiao, and J. Xiao, “Reconstructing dual learning for neural voice conversion using relatively few samples,” in IEEE Automatic Speech Recognition and Understanding Workshop. IEEE, 2021, pp. 946–953.

[18] B. Zhao, X. Zhang, J. Wang, N. Cheng, and J. Xiao, “unspeech: Speaker-guided conditional variational autoencoder for zero-shot multi-speaker text-to-speech,” in 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2022). IEEE, 2022, pp. 1–5.

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

[20] X. Zhang, Y. Jiang, J. Deng, J. Li, M. Tian, and W. Li, “A novel singer identification method using gmm-ubm,” in Proceedings of the 6th Conference on Sound and Music Technology (CSMT2018). Springer, 2019, pp. 3–14.

[21] L. Su and Y.-H. Yang, “Sparse modeling for artist identification: Exploring phase information and vocal separation,” in Proceedings of the 14th International Society for Music Information Retrieval Conference, 2013, pp. 349–354.

[22] L. Bazzon and M. Marconetti, “Domain adaptation problems: A dasvm classification technique and a circular validation strategy,” IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 5, pp. 770–787, 2009.

[23] W.-S. Chu, F. De la Torre, and J. F. Cohn, “Selective transfer machine for personalized facial action unit detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3515–3522.

[24] X. Zhang, J. Wang, N. Cheng, E. Xiao, and J. Xiao, “CycleGAN: cycle generative enhanced adversarial network for voice conversion,” in IEEE Automatic Speech Recognition and Understanding Workshop (ASRU2021). IEEE, 2021, pp. 1–6.

[25] B. Gong, K. Grauman, and F. Sha, “Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation,” in International Conference on Machine Learning. PMLR, 2013, pp. 222–230.

[26] J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 199–210, 2010.

[27] H. Tang, X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Aqvvc: One-shot voice conversion by vector quantization with applying contrastive learning,” in 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2022). IEEE, 2022, pp. 1–5.

[28] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Deep transfer learning with joint adaptation networks,” in International Conference on Machine Learning. PMLR, 2017, pp. 2208–2217.

[29] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, “Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2272–2281.

[30] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Singer identification for metaverse with timbral and middle-level perceptual features,” in International Joint Conference on Neural Networks, IJCNN 2022. IEEE, 2022, pp. 1–7.

[31] M. Long, Y. Cao, J. Wang, and M. Jordan, “Learning transferable features with deep adaptation networks,” in International conference on Machine learning. PMLR, 2015, pp. 97–105.

[32] X. Qu, J. Wang, and J. Xiao, “Enhancing data-free adversarial distillation with activation regularization and virtual interpolation,” in IEEE
[33] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Unsupervised domain adaptation with residual transfer networks,” in Proceedings of the 30th International Conference on Neural Information Processing Systems. Curran Associates Inc., 2016, p. 136–144.

[34] M. Ghifary, W. B. Kleijn, and M. Zhang, “Domain adaptive neural networks for object recognition,” in Pacific Rim international conference on artificial intelligence. Springer, 2014, pp. 898–904.

[35] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” arXiv preprint arXiv:1412.3474, 2014.

[36] K. Choi, G. Fazekas, and M. Sandler, “Explaining deep convolutional neural networks on music classification,” arXiv preprint arXiv:1607.02444, 2016.

[37] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Susing: Su-net for singing voice synthesis,” in International Joint Conference on Neural Networks, IJCNN 2022. IEEE, 2022, pp. 1–7.

[38] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional recurrent neural networks for music classification,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 2392–2396.

[39] Q. Wang, X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Drvc: A framework of any-to-any voice conversion with self-supervised learning,” in 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2022). IEEE, 2022, pp. 1–5.

[40] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” The journal of machine learning research, vol. 17, no. 1, pp. 2096–2030, 2016.

[41] A. Aljanaki and M. Soleymani, “A data-driven approach to mid-level perceptual musical feature modeling,” in Proceedings of the 19th International Society for Music Information Retrieval Conference, 2018, pp. 615–621.

[42] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Tdass: Target domain adaptation speech synthesis framework for multi-speaker low-resource tts,” in International Joint Conference on Neural Networks, IJCNN 2022. IEEE, 2022, pp. 1–7.

[43] B. Whitman, G. Flake, and S. Lawrence, “Artist detection in music with minnowmatch,” in Proceedings of the 2001 IEEE Signal Processing Society Workshop. IEEE, 2001, pp. 559–568.

[44] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in Proceedings of the 14th python in science conference, vol. 8, 2015, pp. 18–25.

[45] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Mdconv-sid: Multi-scale dilated convolution network for singer identification,” in International Joint Conference on Neural Networks, IJCNN 2022. IEEE, 2022, pp. 1–7.

[46] X. Li, Y.-Y. Wang, and A. Acero, “Learning query intent from regularized click graphs,” in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 2008, pp. 339–346.