

SHDA: Sinkhorn Domain Attention for Cross-Domain Audio Anti-Spoofing

Ruiteng Zhang¹, Jianguo Wei^{1,2}, Xugang Lu³, Lin Zhang⁴, Di Jin¹, Junhai Xu^{1†}, Wenhuan Lu^{1†}

Abstract—Audio anti-spoofing algorithms struggle with fake samples from unseen spoofing techniques, even when trained with diverse data sets or data augmentation strategies. Unsupervised domain adaptation (UDA) algorithms have the potential to mitigate this challenge. Typically, UDA assumes that the source and target domains are distinct distributions with clear boundaries and seeks to align model representations between them. However, in anti-spoofing, various spoofing algorithms could cause the distributions of the generated samples to overlap, resulting in unclear domain boundaries. This hinders UDA algorithms from effectively measuring and aligning domain discrepancies. Moreover, forcibly aligning samples with significant discrepancies could diminish the model’s discriminative capability. To solve this problem, we propose a domain attention algorithm with optimal transport (OT), termed Sinkhorn Domain Attention (SHDA). Unlike traditional attention mechanisms, SHDA identifies the optimal transfer plan by analyzing the global probability differences among cross-domain samples. Specifically, we first extract audio representations from various domains to compute the overall cost matrix between the source and target domains. Next, we employ Sinkhorn’s iteration to calculate the OT coupling matrix, where cross-domain samples with minor differences receive higher transfer weights, while those with substantial differences receive lower weights. Finally, we use the coupling and cost matrices to compute the adaptation loss, effectively transferring the anti-spoofing model from multiple sources to the target domain. We conducted eight cross-domain experiments using eleven well-known anti-spoofing corpora. The results indicate that our label-free SHDA surpassed the state-of-the-art model by 40%.

Index Terms—Audio anti-spoofing, Cross-domain, Unsupervised domain adaptation, Optimal transport, Domain attention.

I. INTRODUCTION

AUTOMATIC speaker verification (ASV) systems [1], [2] are widely used for identity verification. However, the growing prevalence of voice spoofing threatens their effectiveness, primarily via text-to-speech (TTS), voice conversion (VC), and deepfake attacks [3], [4]. Prompted by these threats, researchers have developed anti-spoofing algorithms to safeguard ASV systems [5].

Most audio anti-spoofing research focuses on designing front-end feature extractors [6], [7] and back-end encoders [8]–[11]. For front-end extraction [6], studies have explored hand-crafted acoustic features and deep features derived from pre-trained models [12]–[14]. For back-end classification, researchers have explored stacked convolutional neural networks

(CNNs) and heterogeneous graph neural networks (HGNNs) [8]–[11] to capture subtle differences between spoofed and bonafide speech. However, in real-world applications, anti-spoofing algorithms often encounter speech from different devices, background noises, and environments [15], [16]. Furthermore, with the rapid advancement of speech techniques, new attacks continue to emerge, many of which are not represented in the training set [3], [17], [18]. These cross-domain scenarios could significantly degrade performance, even for models that perform well in in-domain settings [4], [15], [16].

To overcome the domain mismatch in anti-spoofing, researchers have explored domain generalization (DG) strategies. Some studies focus on designing data augmentation methods to increase the diversity of the training set, thereby improving performance in target domains [19], such as the RawBoost [20] and CpAug [21] augmentation toolkits. Other studies concentrate on one-class strategies, where the encoder focuses only on real audio and classifies all abnormal samples as spoofing [22]. This approach is typically implemented using the OC-Softmax loss function [22], [23] and the knowledge distillation framework [24]. Recently, to further enhance generalization capabilities, researchers have explored blending anti-spoofing data sets from various domains to train models [25]–[27]. We name this strategy ‘multi-source mixing training.’ Although DG strategies improve model robustness, they cannot align the probability distribution of the model’s representations across different domains, making it difficult to adapt to scenarios with large domain shifts. For example, although commonly used methods are effective in the ASVspoo challenge [4], they struggle with the In-the-Wild (ITW) data set [17], [28]. Therefore, mitigating domain discrepancies solely through artificially designed DG strategies remains challenging.

In this paper, we propose using unsupervised domain adaptation (UDA) [29] to mitigate the performance degradation of audio anti-spoofing algorithms in unknown scenarios. UDA leverages unlabeled target data to align domain discrepancies rather than relying on manually designed DG strategies [29]. Target data can be sourced from terminal devices in target domains [30]. Like DG, UDA does not require data annotation but more effectively mitigates the domain mismatch problem [31]. In audio spoofing detection, UDAs are mainly designed based on domain adversarial learning [31], [32]. By incorporating a gradient reversal layer (GRL) [33], the anti-spoofing model cannot distinguish the domain to which the fake audio belongs, thereby extracting a consistent representation of fake audio across different domains [16]. Although research on

1. College of Intelligence and Computing, Tianjin University, Tianjin, China.

2. Computer College, Qinghai Nationalities University, Xining, China.

3. National Institute of Information and Communications Technology, Kyoto, Japan.

4. Brno University of Technology, Brno, Czechia.

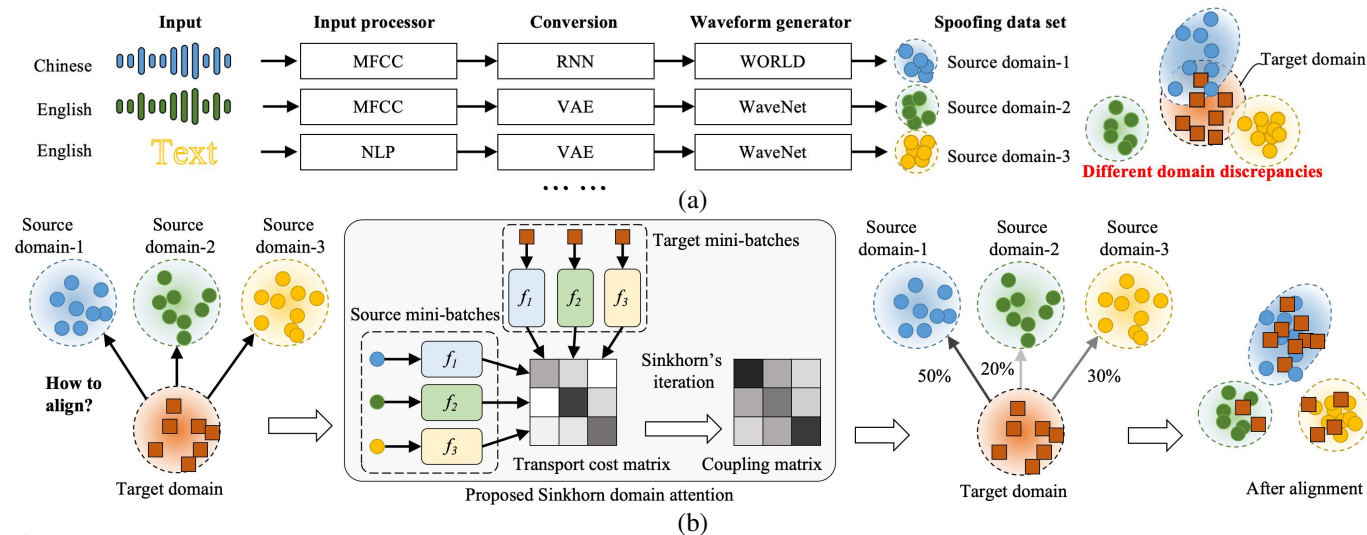


Fig. 1: The basic assumption of our proposal. (a): Each speech sample generated by various techniques represents a distinct domain. However, because different spoofing algorithms may employ similar techniques and data, the discrepancies between these domains could vary. (b): Diagram of the proposed SHDA. First, the output of each source domain branch is merged with the output of the target branch to calculate a unified transport cost matrix. Then, the transport coupling is calculated using the Sinkhorn’s iteration algorithm to assign important source domain samples and optimal transport weights to the current target domain data. After applying SHDA, the anti-spoofing model aligns target samples with source samples that have a close probability distance.

UDA in audio anti-spoofing is limited, it has attracted significant attention in related fields, such as speaker verification and face anti-spoofing [34], [35]. Representative algorithms include maximum mean difference (MMD) [36], correlation alignment (CORAL) [37], and UDA algorithms based on self-supervised learning (SSL) [35].

However, directly adopting existing UDA algorithms fails to effectively mitigate domain shifts in audio anti-spoofing. This is because different spoofing algorithms could utilize similar techniques or data [3], [4], resulting in diverse domain discrepancies between the generated audio, as shown in Fig. 1(a). In this figure, some algorithms generate samples with minor domain discrepancies, while others generate large ones. In anti-spoofing, these diverse discrepancies have received attention in other studies, yet they have not been adequately addressed. For example, in ASVspoof2019 [4], samples generated by A10 and A11 showed minimal differences in probability distributions because both used the same TTS architecture, whereas they differed significantly from samples generated by other algorithms [38]. Similarly, domain generalization studies for the Audio Deep synthesis Detection (ADD) challenge underscored the discrepancies in spoofing samples across different domains [39]. Further supporting these findings, visualizations using the t-distributed stochastic neighbor embedding (t-SNE) [40] algorithm revealed clear clustering differences in speech generated by various vocoders, highlighting the varying domain shifts among spoofing samples [41]. Consistent with these findings, our visualizations in Section V-A show the diverse domain discrepancies between various spoofing samples. Such variability blurs the boundaries between the source and target domains, challenging the UDA’s capability to align domain differences effectively [42]. Additionally, existing anti-spoofing models are typically trained using a multi-source mixing strategy, which could exacerbate the diversity of domain discrepancies between samples [25], [26], [43].

To address this challenge, we propose a domain attention algorithm for anti-spoofing. Conventional attention mecha-

nisms typically evaluate the contributions of specific regions and channels within a feature map, emphasizing key features for downstream tasks [44]. In contrast, our domain attention focuses on assessing the degree of discrepancies between the source and target domain samples, thereby prioritizing ‘important’ transfers. It guides the model to focus primarily on aligning cross-domain samples with minor domain discrepancies, while still moderately accounting for those with larger discrepancies. This could reduce interference in source branch training and prevent misalignment between different categories (e.g., bonafide and spoofing samples). We propose the Sinkhorn Domain Attention (SHDA) algorithm to implement our concept, as shown in Fig. 1(b). Specifically, each source domain branch (f_1, f_2 , and f_3) encodes audio from different source domains, then calculates the total transport cost matrix between these and the target representations. Then, we use the Sinkhorn algorithm [45] to compute the optimal transport (OT) [46] coupling matrix. In SHDA, the Kantorovich-relaxation theorem [47] is applied to define the OT problem. It ensures that SHDA prioritizes cross-domain samples with smaller differences without entirely overlooking those with substantial discrepancies. Finally, SHDA uses the coupling matrix to assign optimal transfer weights to cross-domain samples based on their domain discrepancies. The contributions are summarized as follows.

- (1) We propose a domain attention framework for cross-domain anti-spoofing. It is designed to steer the anti-spoofing model’s focus toward important transfers based on the degree of discrepancies between the samples of different domains.
- (2) Based on the domain attention, we propose the SHDA algorithm for cross-domain audio anti-spoofing. SHDA calculates the transfer coupling between various source and target domains, assigning optimal transfers to them. Additionally, we design a two-step optimization strategy to enable the anti-spoofing model with SHDA to be trained end-to-end.
- (3) We conducted eight cross-domain experiments using eleven well-known anti-spoofing data sets. In these exper-

iments, we evaluated the proposed SHDA across various cross-domain scenarios, which included variations in language, noise, codec, and attacks. For example, the proposed SHDA surpassed the state-of-the-art model by 40% in the ITW corpus. Furthermore, we visualized the working mechanism of SHDA to elucidate its underlying principles.

The remainder of this paper is organized as follows: Section II offers a literature review of cross-domain audio anti-spoofing algorithms. Section III describes the proposed SHDA algorithm and the accompanying two-step optimization strategy. Section IV assesses the performance of SHDA through systematic cross-domain experiments. Section V delves deeper into the domain attention mechanism and provides both ablation and generalization studies. Finally, Section VI presents conclusions and outlines directions for future research.

II. RELATED WORK

This section reviews related work on audio spoofing detection methods, domain generalization strategies, unsupervised domain adaptation algorithms, and optimal transport.

A. Audio Spoofing Detection Methods

In audio anti-spoofing, most studies focus on the front-end feature extractor design, back-end encoder development, and data set construction. We review these studies as follows.

In front-end feature extraction, some studies focus on hand-crafted speech features [48], such as linear filter frequency cepstral coefficients (LFCC) [49]. Others focus on designing learnable feature extractors, among which SincNet is a notable work [50]. Recently, researchers have introduced large-scale pre-training models like Wav2vec and WavLM into anti-spoofing to extract robust deep features [13], [14].

In back-end encoder design, some studies focus on efficiently stacking network layers, such as lightweight CNN (LCNN) [51], ResNet [8], and 1-dimensional CNN [52]. Others integrate the backbone network with machine learning tools, such as attention mechanisms, multi-scale feature modeling, and HGNNs. Representative works include ECAPA-TDNN [53], [54] and AASIST [11]. Additionally, hybrid architectures that combine pre-training models with backbone networks have shown notable advancements, such as MFA [55] and AASIST2 [56].

To encourage further exploration of anti-spoofing, the design of data sets has gained widespread attention. These studies have developed corpora across various scenarios, attacks, and languages. In terms of language, the ASVspooF challenges [3], [4], FoR [57], and Wavefake (LJSpeech subset) [58] have been collected in English. CFAD [59] and FMFCC-A [60] data sets are designed for analyzing fake audio in Chinese scenarios. Wavefake (JSUT subset) [58] focuses on Japanese scenarios, while HABLA [61] targets Spanish conditions. Regarding attack scenarios, most of the aforementioned data sets focus on fake audio generated by generative models like VC or TTS. CFAD and PartialSpooF [62] increasingly emphasize the recently emerging partial spooF scenario. Recently, ITW [17] has garnered widespread attention for its role in evaluating models in real-world scenarios.

B. Domain Generalization Strategies in Anti-Spoofing

Although existing models perform well against known attacks, they struggle with unknown ones [63]. In anti-spoofing, domain generalization strategies are widely used to mitigate domain mismatch. Some studies focus on data augmentation methods, such as RawBoost [20] and DASC [64], which add noise and perturbations to make training data resemble test conditions. Others adopt a one-class strategy [22], training encoders to recognize only bonafide samples and classifying all abnormal samples as spoofed speech. OC-Softmax [22] and SAMO [65] refine one-class loss functions, while OCKD applies knowledge distillation to implement a one-class network [24]. To further improve generalization, researchers have explored blending anti-spoofing data sets from various domains to train models [25]–[27]. We name this approach ‘multi-source mixing training.’ It enhances the model’s ability to capture common features of fake speech, thereby reducing the risk of overfitting to specific attacks. Representative works include ASDG [26] and W2V-ASDG [43]. Despite these efforts, their effectiveness is limited in scenarios with large domain discrepancies [17].

C. Unsupervised Domain Adaptation Algorithms

UDA algorithms utilize unlabeled target data to mitigate domain mismatch in anti-spoofing models [29]. Like DG strategies, UDA does not require target data annotation but better mitigates domain mismatch. The adversarial unsupervised domain adaptation (AUDA) framework [33] is a commonly used UDA architecture in audio anti-spoofing [16]. Specifically, adding a GRL [33] to the anti-spoofing model prevents the model from identifying the domain to which the sample belongs, such as in LCNN-DAT [31]. Some researchers have refined this framework by considering the differences between the spoofing and bonafide samples, such as in CGCNN-DADA [32]. Although research on UDA in audio anti-spoofing is limited, it shows promise as a tool for mitigating channel and language mismatches in related tasks such as ASV. Representative algorithms include MMD [36] and CORAL [37]. In face anti-spoofing, SSL-based UDA has been shown to effectively mitigate the domain mismatch problem [66]. Nevertheless, in audio anti-spoofing, existing UDA algorithms struggle to handle complex scenarios like ITW [17]. This challenge stems from the fact that different spoofing algorithms could employ similar techniques, leading to varied domain discrepancies in the generated audio [3], [4]. This variation blurs the boundaries between the source and target domains, undermining UDA’s effectiveness in aligning domain differences [42]. Moreover, forcing alignment of samples with significant domain discrepancies could impair the model’s ability to detect spoofing [67], [68]. *Therefore, it is necessary to develop a transfer selection mechanism that formulates distinct alignment plans for samples with varying domain discrepancies, rather than aligning them indiscriminately.* In this study, we propose a probability distribution attention mechanism to achieve this.

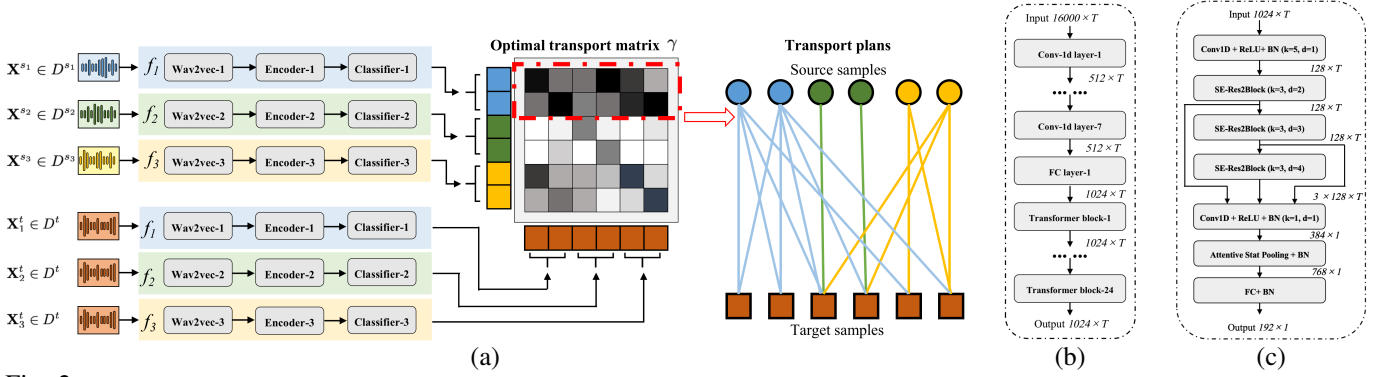


Fig. 2: The network architecture of the proposed SHDA: (a) The diagram illustrates our Siamese anti-spoofing architecture, which includes three source domains represented by three corresponding models as an example. In the figure, we illustrate the example where transmissions from source domain-1 are more significant, showcasing the transport coupling and plan. (b) Model structure of Wav2vec. (c) Model structure of ECAPA-TDNN.

D. Optimal Transport

OT has been widely studied in domain adaptation in machine learning [69]. Its original motivation is to determine an optimal transfer plan that transforms one probability distribution into another with minimal cost [69]. This transfer plan is represented by a transport coupling matrix, which defines how samples in the source distribution are mapped to the target distribution. Here, the transport cost matrix is defined by measuring the difference between source and target domain samples, typically calculated using the Euclidean distance [70]. OT defines a distance measure between different probability distributions. One of its key properties is that it accounts for the geometric structure of distributions when performing optimal transport. Due to this property, OT is a promising tool for image processing, classification, segmentation, and speech enhancement [46], [69], [71]. Motivated by these advantages, we propose SHDA to mitigate the domain mismatch problem in anti-spoofing.

III. PROPOSED METHOD

In this section, we propose the SHDA algorithm, designed to adapt an anti-spoofing model trained on multiple source domains to a specific target domain. Instead of directly aligning cross-domain samples, SHDA uses OT to determine the optimal transport plan based on their discrepancies. This process functions similarly to an ‘attention mechanism’ between the probability distributions of the source and target domains. Specifically, we first construct a unified transport cost matrix for cross-domain samples across all domains to measure their discrepancies. Subsequently, we adopt Sinkhorn’s iteration to compute the optimal transport coupling, which identifies the most relevant cross-domain pairs. Finally, we integrate this strategy into end-to-end anti-spoofing model training to effectively mitigate the domain mismatch problem. Next, we detail the proposed SHDA algorithm and its integration into the anti-spoofing framework.

A. Problem Definition

Given a multi-source domain $D^s = \{D^{s_i}\}_{i=1, \dots, N_s}$, where N_s is the number of source domains it contains. The i -th source domain is defined as $D^{s_i} = \{(\mathbf{x}_j^{s_i}, \mathbf{y}_j^{s_i})\}_{j=1, \dots, n_{s_i}}$, where n_{s_i} is the number of samples in the i -th source domain, and \mathbf{x} and \mathbf{y} are the input speech sample and the corresponding

label, respectively. Additionally, we define a target domain $D^t = \{(\mathbf{x}_j^t, \mathbf{y}_j^t)\}_{j=1, \dots, n_t}$, where n_t is the number of its samples. In UDA, \mathbf{y}^t is unknown. Under domain mismatch, the independently and identically distributed (i.i.d.) assumption is violated as $\mathcal{P}^s \neq \mathcal{P}^t$. To decrease this domain discrepancy, UDA aims to train a neural network to produce consistent representations across domains, effectively approximating $\mathcal{P}^s \approx \mathcal{P}^t$. Considering the varying domain discrepancies between source and target samples in anti-spoofing, we assign globally optimal transfer weights to cross-domain samples during adaptation instead of directly aligning them.

B. Neural Audio Anti-Spoofing Model

As shown in Fig. 2, the proposed adaptive anti-spoofing framework utilizes a Siamese network architecture, where each branch consists of a deep feature extractor, an encoder, and a classifier, arranged sequentially. Each branch functions to transform speech signals into the embeddings and label predictions, defined as follows:

$$\mathbf{e}, \hat{\mathbf{y}} = f(q(\mathbf{x}), k(\mathbf{x})), \quad (1)$$

$$\hat{\mathbf{y}} = k(\mathbf{x}, \theta_g, \theta_h, \theta_v) = g \circ h \circ v(\mathbf{x}), \quad (2)$$

$$\mathbf{e} = q(\mathbf{x}, \theta_h, \theta_v) = h \circ v(\mathbf{x}), \quad (3)$$

where $v(\cdot)$, $h(\cdot)$, $g(\cdot)$ are the feature extractor, encoder, and classifier transforms with parameter sets θ_v , θ_h , θ_g , respectively. ‘ \circ ’ denotes the function composition operator. \mathbf{e} is the embedding, and $\hat{\mathbf{y}}$ denotes the estimated label. To enhance generalization in unseen scenarios, this study integrates Wav2vec and ECAPA-TDNN to construct a backbone model. Wav2vec is a pre-trained model trained on large-scale speech data that extracts generalized features [49]. ECAPA-TDNN is a multi-scale model that captures both long-term and short-term differences between bonafide and spoofing samples [72]. By leveraging these strengths, our backbone model improves the discriminative capability of target samples, thereby enhancing the effectiveness of domain transfer. The details of the feature extractor, encoder, and classifier are outlined as follows:

1) *Deep feature extractor*: We employ the XLM-R (0.3b) version¹ of Wav2vec 2.0 as a deep feature extractor, as shown in Fig. 2(b). It consists of seven 512-dimensional CNN layers, a 1024-dimensional fully connected (FC) layer, and

¹<https://github.com/pytorch/fairseq/tree/main/examples/wav2vec>

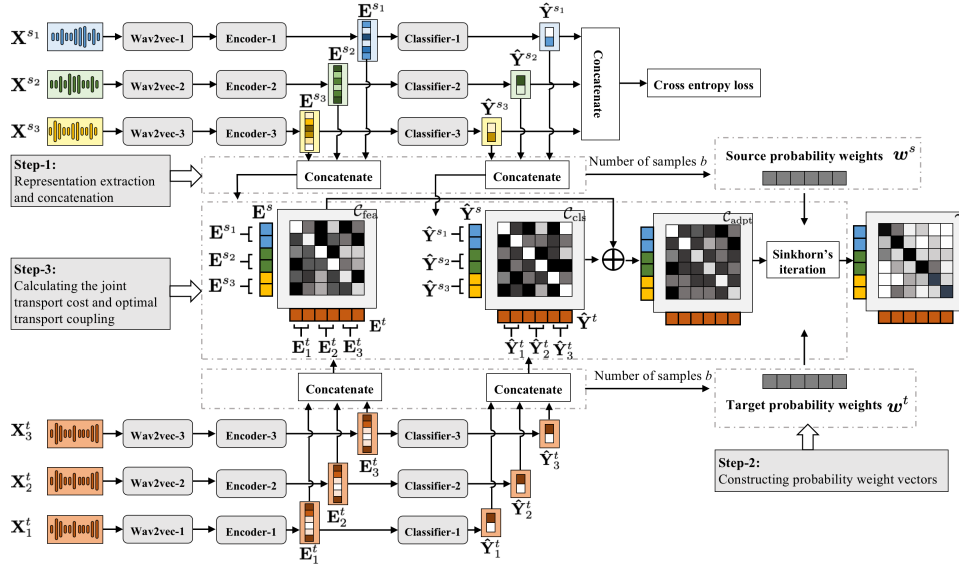


Fig. 3: Three steps of SHDA in a mini-batch.

24 Transformer blocks, each with 1024 dimensions [44]. In our framework, every second of audio is converted into a 50×1024 -dimensional deep feature representation.

2) *Encoder*: Our encoder is built with ECAPA-TDNN² (Fig. 2(c)), stacking three 128-dimensional SERes2NetBlocks [73] for multi-scale feature extraction, producing 192-dimensional embeddings.

3) *Classifier*: The classifier includes two FC layers with the ReLU activation function and a batch normalization layer. The dimensions of these FC layers are 192 and 2, respectively.

In this study, our Siamese architecture comprises three anti-spoofing models (f_1 , f_2 , and f_3) with shared parameters, as shown in Fig. 2(a). These models extract representations for their respective source domains and together for the target domain. The SHDA module then processes the representations to compute optimal transfer weights based on the domain discrepancies among cross-domain samples.

C. Sinkhorn Domain Attention

The proposed SHDA, a probabilistic attention mechanism, is designed to transfer key knowledge from source to target domain samples, directed by the extent of domain discrepancies between them. To achieve this, we first extract representations from N_s source domains. The representations from the target domain are then repeated N_s times to calculate the probability distribution distances between the target domain and each source domain. Next, we calculate the transport coupling matrix between domains using the Sinkhorn iteration algorithm. Finally, based on the transfer plan, we emphasize the transfers between samples from the source and target domains with similar probability distribution distances.

Fig. 3 illustrates the SHDA architecture with three source domains during a mini-batch training process, where each model consistently extracts representations from both the source and target mini-batches. For clarity, we have labeled the three source mini-batches as $\{X^{s_i} \in D^{s_i}\}_{i=1,2,3}$ and the three target mini-batches as $\{X_i^t \in D^t\}_{i=1,2,3}$. Then, the source

mini-batch X^{s_i} and the target mini-batch X_i^t are input into f_i to extract corresponding embeddings and label predictions, which can be defined as follows:

$$\{(E^{s_1}, \hat{Y}^{s_1}), (E^{s_2}, \hat{Y}^{s_2}), (E^{s_3}, \hat{Y}^{s_3})\} = \{f_i(X^{s_i})\}_{i=1,2,3}, \quad (4)$$

$$\{(E_1^t, \hat{Y}_1^t), (E_2^t, \hat{Y}_2^t), (E_3^t, \hat{Y}_3^t)\} = \{f_i(X_i^t)\}_{i=1,2,3}. \quad (5)$$

First, as illustrated in Step 1 of Fig. 3, we concatenate the embeddings and label predictions from the three source branches, defined as follows:

$$E^s = \text{concatenate}(E^{s_1}, E^{s_2}, E^{s_3}), \quad (6)$$

$$\hat{Y}^s = \text{concatenate}(\hat{Y}^{s_1}, \hat{Y}^{s_2}, \hat{Y}^{s_3}), \quad (7)$$

where E^s and \hat{Y}^s represent the tensors of concatenated embeddings and predictions for each source domain, respectively. Correspondingly, the embeddings and label predictions from the three target branches are also concatenated as E^t and \hat{Y}^t , and can be defined as follows:

$$E^t = \text{concatenate}(E_1^t, E_2^t, E_3^t), \quad (8)$$

$$\hat{Y}^t = \text{concatenate}(\hat{Y}_1^t, \hat{Y}_2^t, \hat{Y}_3^t). \quad (9)$$

In Equations (6) to (9), each concatenated matrix contains b samples that are used to calculate optimal transport. In SHDA, optimal transport coupling determines the significance of each source domain's samples and their corresponding transfer weights during the adaptation process.

Before calculating OT, as detailed in Step 2 of Fig. 3, we must define the probability weights of the source and target samples, which can be defined as follows:

$$w^s \triangleq \{w_1^s, w_2^s, \dots, w_b^s\}_{w_i^s = \frac{1}{b}}, \quad (10)$$

$$w^t \triangleq \{w_1^t, w_2^t, \dots, w_b^t\}_{w_i^t = \frac{1}{b}}, \quad (11)$$

where w^s and w^t are the probability weights of the source and target mini-batches, respectively. Since the target data is unlabeled, we follow [45], [46] and set the probability weights to a uniform distribution to ensure equal treatment of all cross-domain samples, thereby preventing bias toward specific attacks.

²https://github.com/speechbrain/speechbrain/blob/develop/speechbrain/lobes/models/ECAPA_TDNN.py

Subsequently, as illustrated in Step 3 of Fig. 3, the optimal transport problem in SHDA is defined and solved. According to the Kantorovich-relaxation theorem [47], a dual transport polytope can be established to facilitate the conversion between one probability distribution and another, defined as follows:

$$\Pi(\mathcal{P}^s, \mathcal{P}^t) \triangleq \{\gamma \in \mathbb{R}_+^{b \times b} | \gamma \mathbf{1}_b = \mathbf{w}^s, \gamma^\top \mathbf{1}_b = \mathbf{w}^t\}, \quad (12)$$

where $\Pi(\mathcal{P}^s, \mathcal{P}^t)$ contains all of the nonnegative $b \times b$ matrices that plan how to transport \mathcal{P}^s to \mathcal{P}^t , $\mathbf{1}_b$ is the b dimensional vector of ones, and γ is the transport coupling. With this definition, the joint distribution OT between the source \mathcal{P}^s and target \mathcal{P}^t probability distributions can be defined as follows:

$$\mathcal{D}_{\text{JOT}}(\mathcal{P}^s, \mathcal{P}^t) \triangleq \min_{\gamma \in \Pi(\mathcal{P}^s, \mathcal{P}^t)} \sum_{i,j} \mathcal{C}_{\text{adpt}}(\mathbf{z}_i^s, \mathbf{z}_j^t) \gamma(\mathbf{z}_i^s, \mathbf{z}_j^t), \quad (13)$$

where $\mathbf{z}_i^s = \{(\mathbf{e}_i^s, \hat{\mathbf{y}}_i^s)\}_{\mathbf{e}_i^s \in \mathbf{E}^s, \hat{\mathbf{y}}_i^s \in \hat{\mathbf{Y}}^s}$ and $\mathbf{z}_j^t = \{(\mathbf{e}_j^t, \hat{\mathbf{y}}_j^t)\}_{\mathbf{e}_j^t \in \mathbf{E}^t, \hat{\mathbf{y}}_j^t \in \hat{\mathbf{Y}}^t}$ represent tuples of joint samples from the source and target domains, with i and j denoting the indexes of source and target samples, respectively. In Eq. (13), \sum represents a double summation, and $\mathcal{C}_{\text{adpt}}(\mathbf{z}_i^s, \mathbf{z}_j^t)$ is the cost function between samples \mathbf{z}_i^s and \mathbf{z}_j^t from distributions \mathcal{P}^s and \mathcal{P}^t , respectively. Therefore, we can use the features and conditional distribution to calculate the joint cost between the source domains and the target domain, defined as follows:

$$\mathcal{C}_{\text{adpt}}(\mathbf{z}_i^s, \mathbf{z}_j^t) = \alpha \mathcal{C}_{\text{fea}}(\mathbf{e}_i^s, \mathbf{e}_j^t) + \beta \mathcal{C}_{\text{cls}}(\hat{\mathbf{y}}_i^s, \hat{\mathbf{y}}_j^t), \quad (14)$$

where $\mathcal{C}_{\text{fea}}(\cdot)$ and $\mathcal{C}_{\text{cls}}(\cdot)$ measure the distances of embedding and label predictions, respectively. α and β are the weighting coefficients for these two cost functions, respectively. To ensure the stability of OT calculation, following [70], [74], the Euclidean distance is used to compute $\mathcal{C}_{\text{fea}}(\cdot)$ and $\mathcal{C}_{\text{cls}}(\cdot)$, which are defined below:

$$\mathcal{C}_{\text{fea}}(\mathbf{e}_i^s, \mathbf{e}_j^t) = \|\mathbf{e}_i^s - \mathbf{e}_j^t\|_{i,j \in \{1,2,\dots,b\}}^2, \quad (15)$$

$$\mathcal{C}_{\text{cls}}(\hat{\mathbf{y}}_i^s, \hat{\mathbf{y}}_j^t) = \|\hat{\mathbf{y}}_i^s - \hat{\mathbf{y}}_j^t\|_{i,j \in \{1,2,\dots,b\}}^2. \quad (16)$$

After solving Eq. (13), we obtain the transport coupling γ for all source and target domain samples, as shown in Fig. 3. Based on γ , SHDA determines which target domain samples to transfer and assigns appropriate transfer weights to them. In the next section, we integrate the proposed SHDA into the anti-spoofing network training.

D. End-to-End Adaptive Anti-spoofing Framework

Here, we describe the end-to-end training framework that calculates OT using the Sinkhorn iteration algorithm and updates the model parameters of the neural network using the gradient descent algorithm.

In SHDA, the model of the source domain is trained using cross entropy (CE) loss [22], which is defined as follows:

$$\mathcal{L}_{\text{CE}}(\mathbf{y}_i^s, \hat{\mathbf{y}}_i^s) \triangleq - \sum_{j=1}^{N_c} y_{i,j}^s \log \hat{y}_{i,j}^s, \quad (17)$$

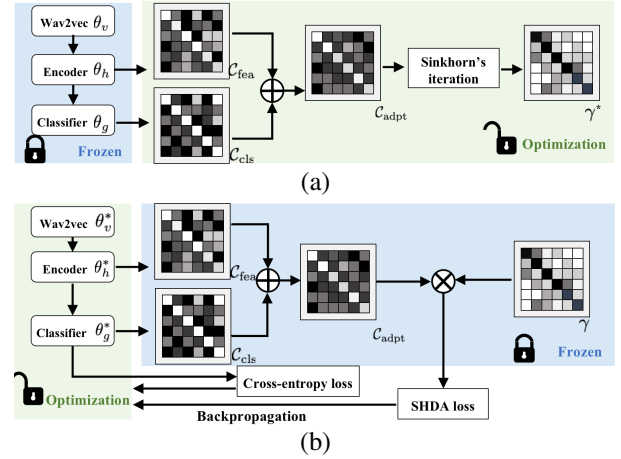


Fig. 4: Flowchart of the proposed two-step optimization strategy. (a) Step 1: solving the optimal transport coupling γ using the Sinkhorn iteration algorithm; (b) Step 2: updating the model parameters of neural networks using the gradient descent algorithm.

where i is the sample index, and N_c is the number of categories. Therefore, the total loss \mathcal{L}_T in our adaptive anti-spoofing model can be defined as follows:

$$\mathcal{L}_T(\cdot) = \mathcal{L}_{\text{CE}}^s(\cdot) + \eta \mathcal{L}_{\text{JOT}}(\cdot), \quad (18)$$

where η is the weighting coefficient. The model parameters can be obtained by minimizing the loss defined in Eq. (18):

$$\gamma^*, \theta_g^*, \theta_h^*, \theta_v^* \triangleq \arg \min_{\gamma, \theta_g, \theta_h, \theta_v} \mathcal{L}_T(\cdot). \quad (19)$$

The parameter optimization in Eq. (19) is a two-step process, as shown in Fig. 4. The two steps are defined as follows:

Step 1: calculating the OT coupling matrix. First, we compute the transport coupling γ , as shown in Fig. 4(a). Based on entropy regularization [45], Eq. (13) is converted into:

$$\mathcal{D}_{\text{JOT}}(\mathcal{P}^s, \mathcal{P}^t) \triangleq \min_{\gamma \in \Pi(\mathcal{P}^s, \mathcal{P}^t)} \left(\sum_{i,j} \mathcal{C}_{\text{adpt}}(\mathbf{z}_i^s, \mathbf{z}_j^t) \gamma(\mathbf{z}_i^s, \mathbf{z}_j^t) - \frac{1}{\sigma} H(\gamma) \right), \quad (20)$$

where $H(\cdot)$ is the information entropy function for the distribution, which is defined as follows:

$$H(\mathbf{Q}) = - \sum_{i,j} q_{i,j} \log q_{i,j}, \quad (21)$$

where \mathbf{Q} is a $b \times b$ matrix, and $q_{i,j}$ is an element of \mathbf{Q} . In Eq. (20), entropy regularization makes the OT objective function strictly convex, prevents excessive sparsity, and stabilizes the optimization process. In SHDA, this smoothing reduces the impact of negative transport and facilitates connections across multiple spoofing types, thereby enhancing robustness in the target domain. $\sigma \in (0, \infty)$ controls the regularization strength. We minimize the OT distance itself by obtaining optimal coupling as follows:

$$\gamma^* \triangleq \arg \min_{\gamma \in \Pi(\mathcal{P}^s, \mathcal{P}^t)} \mathcal{D}_{\text{JOT}}(\cdot). \quad (22)$$

Eq. (22) can be computed by a Lagrange formulation and Sinkhorn's celebrated fixed point iteration³ [45]. γ^* is used in the second step to update the network parameters.

³<https://pythonot.github.io>

TABLE I: Statistics of the source domains data sets.

Corpus	Language	# of Attack types	# of Utterances	<i>Bonafide Spoofing</i>
ASVspooof2019LA training set [4]	English	6	2,580	22,800
LJSpeech [58]	English	7	13,100	91,700
FakeAVCeleb [75]	English	1	500	10,835

Step 2: updating the model parameters of the neural network. As shown in Fig. 4(b), based on the OT matrix γ^* , we can compute the corresponding joint OT distance loss:

$$\mathcal{L}_{\text{JOT}}(\cdot) = \sum_{i,j} c_{\text{adpt}}(\mathbf{z}_i^s, \mathbf{z}_j^t) \gamma^*(i, j). \quad (23)$$

Based on this equation, the model parameters θ_g^* , θ_h^* , and θ_v^* could be updated through a gradient descent algorithm, as shown in Fig. 4(b).

In summary, utilizing our two-step optimization strategy, the total loss as defined in Eq. (18) can be optimized in an end-to-end manner, encompassing the neural anti-spoofing model and the SHDA module. Depending on whether the XLM-R model parameters are fine-tuned, we categorize the proposed SHDA into two variants: SHDA (fine-tuning) and SHDA (frozen).

IV. EXPERIMENTS

A. Experimental Conditions

We designed eight cross-domain experiments utilizing eleven well-known anti-spoofing data sets. Subsequently, we detailed the experimental configurations individually.

1) **Source Domain Configurations:** We selected the ASVspooof2019LA (hereafter referred to as 19LA) [4] train set, the LJSpeech portion of WaveFake [58] (hereafter referred to as LJSpeech), and FakeAVCeleb [75] as the source domains, in line with previous studies [26]. During training, we were permitted to use their labels. Table I presents the statistical data for these spoofing corpora.

2) **Cross-Domain Experiment Configurations:** We selected eight data sets to serve as the target domains. These data sets differ in spoofing algorithms, languages, and recording settings, and their statistics are shown in Table II. Taking advantage of these differences, we designed eight cross-domain experiments. Specifically, the anti-spoofing algorithm was trained with supervised learning on the source domain. Then, on the training set of the target domain, an unsupervised domain adaptation algorithm was used to mitigate the domain mismatch. *The domain adaptation process cannot use target domain labels.* For data sets that did not provide predefined training and evaluation sets, we randomly divided 50% of the data as the training set and the rest as the evaluation set, without crossover. The list of partitions can be obtained at <https://github.com/zrtlemontree/SHDA>. After adaptation, we evaluated the model on the target domain's evaluation set.

3) **Domain Adaptation Algorithms:** To systematically evaluate the proposed SHDA, we implemented several state-of-the-art adaptive algorithms for anti-spoofing as follows:

- **AUDA:** We implemented the AUDA algorithm within the Wav2vec and ECAPA-TDNN framework according to the specifications in [31], [32]. It employs both a classification and a domain adversarial loss function. In this study, the weight of the domain adversarial loss was set to 0.1.

TABLE II: Statistics of the target domains data sets. The asterisk ‘★’ indicates the training and testing sets as divided by us. Please note that during the adaptation process, the labels of target samples cannot be used.

Corpus	Language	# of Attack types	# of Utterances	<i>Bonafide Spoofing</i>
ITW training set ★ [17]	English	Unknown	9,981	5,908
ITW evaluation set ★	English	Unknown	9,982	5,908
JSUT training set ★ [58]	Japanese	2	2,500	5,000
JSUT evaluation set ★	Japanese	2	2,500	5,000
FoR training set [57]	English	6	26,941	26,927
FoR evaluation set	English	1(unseen)	2,264	2,370
HABLA training set ★ [61]	Spanish	6	11,408	29,000
HABLA evaluation set ★	Spanish	6	11,408	29,000
CFAD clean training set [59]	Chinese	8	12,800	25,600
CFAD clean evaluation set	Chinese	8	14,000	28,000
CFAD codec training set [59]	Chinese	8	12,800	25,600
CFAD codec evaluation set	Chinese	8	14,000	28,000
CFAD noisy training set [59]	Chinese	8	12,800	25,600
CFAD noisy evaluation set	Chinese	8	14,000	28,000
FMFCC-A training set [60]	Chinese	5	4,000	6,000
FMFCC-A evaluation set	Chinese	13	3,000	17,000

TABLE III: Baseline performance on ASVspooof19LA evaluation set.

System	Description	EER (%)	F1 (%)
Net-KA	[76]	2.39	-
ResNet-18 + OC-Softmax	[22]	2.19	-
AASIST	[11]	1.13	-
RawGAT-ST	[77]	1.06	-
Wav2vec + OCKD	[24]	0.22	-
AASIST	Our implementation	1.24	94.48
RawGAT-ST	Our implementation	1.22	93.46
Wav2vec + ResNet18	Our implementation	1.68	98.69
WavLM + ECAPA-TDNN	Our implementation	1.52	98.09
Wav2vec + ECAPA-TDNN	Our implementation	0.34	99.60

- **BYOL-based UDA:** This method is a well-established adaptation algorithm in ASV [35], [68] and is also applicable to anti-spoofing. It includes a spoofing detection branch alongside an SSL exploration module that explores domain characteristics and intrinsic information within the target sample. In the BYOL-based UDA, BYOL [78] is employed as the domain exploration module. In this study, the weight of the BYOL loss was set at 0.1.

- **SimSiam-based UDA:** SimSiam [79] is utilized for the domain exploration module in this adaptive algorithm, following the BYOL-based UDA architectures and configurations.

4) **Training Configurations:** To keep the input data consistent across different testing methods, all audio files were resampled to 16 kHz and either pruned or zero-padded to 4 seconds [25], [26]. Consistent with studies using pre-trained features [25], [28], we directly use the waveform as the input to the network. For the SHDA (frozen) training stage, no data augmentation algorithms were employed. The Adam algorithm was used to optimize the adaptation process, set the initial learning rate to 0.0001, and set the mini-batch size to 128. α and β in Eq. (14) were empirically set as 0.1 and 0.001 [74]. η defined in Eq. (18) was set to 0.1. For the SHDA (fine-tuning) training phase, we employed the RawBoost data augmentation tool [20] to ensure adequate training of the XLM-R model parameters. The initial learning rate was set to 0.000001, and the mini-batch size was set to 10. The design of other hyper-

TABLE IV: Comparison of EERs (%) and F1 scores (%) between the proposed SHDA and various anti-spoofing models across the target evaluation sets.

Model	Description	ITW		JSUT		FoR		HABLA		CFAD clean		CFAD codec		CFAD noisy		FMFCC-A	
		EER ↓	F1 ↑	EER ↓	F1 ↑	EER ↓	F1 ↑	EER ↓	F1 ↑	EER ↓	F1 ↑	EER ↓	F1 ↑	EER ↓	F1 ↑	EER ↓	F1 ↑
XceptionNet	[26]	60.29	-	26.43	-	-	-	-	-	13.92	-	-	-	26.10	-	-	-
MesoInception	[26]	66.10	-	21.20	-	-	-	-	-	18.58	-	-	-	32.68	-	-	-
Resnet18	[26]	62.14	-	24.10	-	-	-	-	-	12.09	-	-	-	26.66	-	-	-
Rawnet2	[26], [80]	32.74	-	47.60	-	-	-	-	-	44.94	-	-	-	46.81	-	-	-
RawGAT-ST	[26], [77]	-	-	-	-	-	-	-	-	28.38	-	-	-	40.24	-	-	-
AASIST	[11], [26], [43]	19.38	-	6.95	-	-	-	-	-	20.37	-	-	-	33.84	-	-	-
LCNN	[61]	-	-	-	-	-	-	51.10	-	-	-	-	-	-	-	-	-
Top-1 of the FMFCC-A	[60]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9.50	-
TE-ResNet	[81]	-	-	-	-	4.38	-	-	-	-	-	-	-	-	-	-	-
Wav2vec + FC	[28]	7.55	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Wav2vec + FC	[25]	6.10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Wav2vec + OCKD	[24]	7.68	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ASDG	[26], [43]	5.16	-	4.32	-	-	-	-	-	5.70	-	-	-	23.81	-	-	-
RawGAT-ST	Our implementation	17.01	83.38	15.32	67.75	45.49	44.05	24.64	72.41	27.65	58.67	28.05	59.34	39.26	38.69	23.90	76.69
AASIST	Our implementation	15.87	79.85	10.64	87.19	31.38	40.38	23.65	68.12	24.03	49.11	24.04	50.45	32.66	26.17	22.79	70.18
ASDG	Our implementation	6.14	93.43	6.32	73.77	37.63	48.30	5.21	85.40	6.72	93.01	6.98	92.68	24.40	58.66	20.97	77.55
Wav2vec + ECAPA-TDNN	Our implementation	22.63	36.51	17.12	83.42	24.69	56.79	7.05	93.55	27.88	68.05	29.30	65.57	39.59	61.23	32.43	79.21
AUDA	Our implementation	8.05	89.83	7.16	85.09	14.73	51.67	3.42	96.31	11.41	90.40	11.21	90.27	25.99	77.90	20.35	85.90
BYOL-based UDA	Our implementation	7.59	91.71	7.96	88.89	17.36	48.14	3.53	96.57	10.76	91.05	11.08	90.21	24.44	77.90	20.13	86.37
SimSiam-based UDA	Our implementation	7.96	91.59	7.28	91.79	18.65	53.03	3.20	96.86	13.14	88.96	12.24	86.96	25.09	75.31	18.97	87.25
SHDA (frozen)	Proposed	4.77	95.41	5.16	87.70	7.91	84.46	2.14	97.81	4.23	95.74	4.31	95.69	23.71	78.92	11.43	91.65
SHDA (fine-tuning)	Proposed	2.98	97.09	2.72	88.10	1.74	93.47	1.47	92.64	3.96	93.74	3.74	94.92	13.84	85.57	6.88	88.70

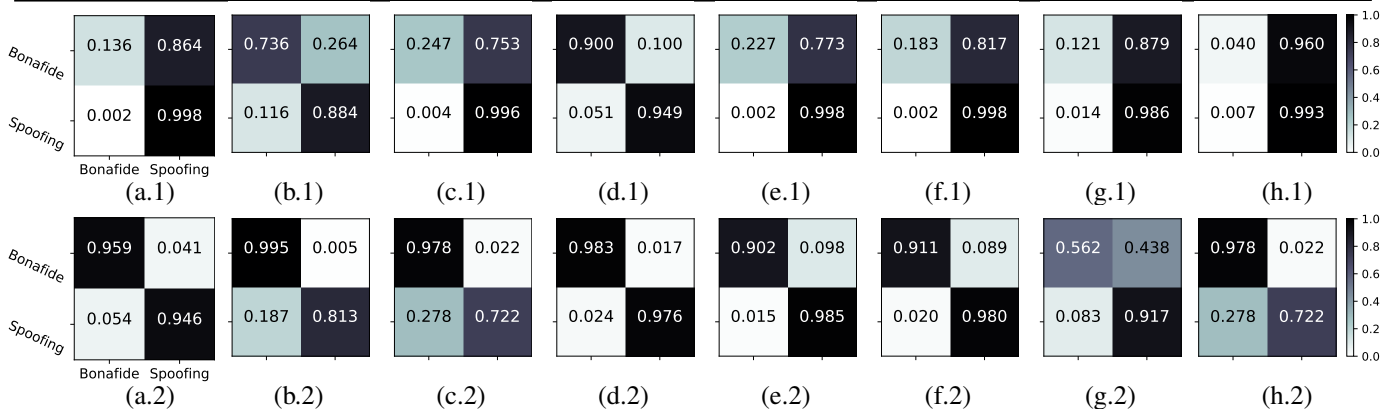


Fig. 5: Confusion matrix showing the classification accuracies without adaptation (*.1) and with SHDA adaptation (*.2) under different target domains. (a) ITW; (b) JSUT; (c) FoR; (d) HABLA; (e) CFAD clean; (f) CFAD codec; (g) CFAD noisy; (h) FMFCC-A.

parameters was aligned with SHDA (frozen). Experimental results were averaged over three runs.

5) **Evaluation Metrics:** Consistent with previous study [26], equal error rate (EER) and F-1 score (F1) were used as the performance metrics.

B. Experimental Results

1) **Performance of Cross-Domain Anti-Spoofing:** First, we evaluated our model's performance against well-known models in the source domain. Additionally, we implemented AASIST⁴ and RawGAT-ST⁵ to further assess our system. The models were trained on the 19LA training set and evaluated on the 19LA evaluation set. Table III shows that our system performs comparably to well-known algorithms.

Building on our baseline system, we undertook comprehensive cross-domain experiments. We integrated 19LA, LJSpeech, and FakeAVCeleb to form the source domain and selected eight challenging anti-spoofing data sets as target domains. In each experiment, we designated one data set as the target domain in turn. The experimental results, shown in Table IV, reveal that the performance of even the most

advanced anti-spoofing models dropped sharply under unknown scenarios. For example, the AASIST implemented in [26] and our implementation achieved EERs of 19.38% and 15.87%, respectively, under ITW. In contrast, integrating adaptive algorithms such as adversarial training led to noticeable improvements within the target domains. This confirms that UDA algorithms enhance the robustness of anti-spoofing models under unknown conditions.

However, current UDA algorithms do not account for the variability in domain discrepancies in spoofing samples. This causes forced alignment of samples with large discrepancies, reducing the model's ability to distinguish between them, particularly in domains with significant shifts such as FoR, CFAD, and FMFCC-A. For example, a BYOL-based UDA obtained a 10.76% EER on CFAD clean. In contrast, the proposed SHDA accounts for the degree of domain discrepancies between the source and target samples. It allocates greater transfer weights to cross-domain samples with minor differences while maintaining some transfers between samples with substantial differences. For example, on CFAD clean, SHDA achieved a 4.23% EER, outperforming existing adaptive algorithms by 60%. Additionally, on ITW, SHDA achieved a 2.98% EER, surpassing the state-of-the-art ASDG [26].

2) **Comparisons of the Confusion Matrix:** We used the confusion matrix to analyze the impact of the proposed

⁴<https://github.com/clovaai/aasist>

⁵<https://github.com/eurecom-asp/RawGAT-ST-antispoofing>

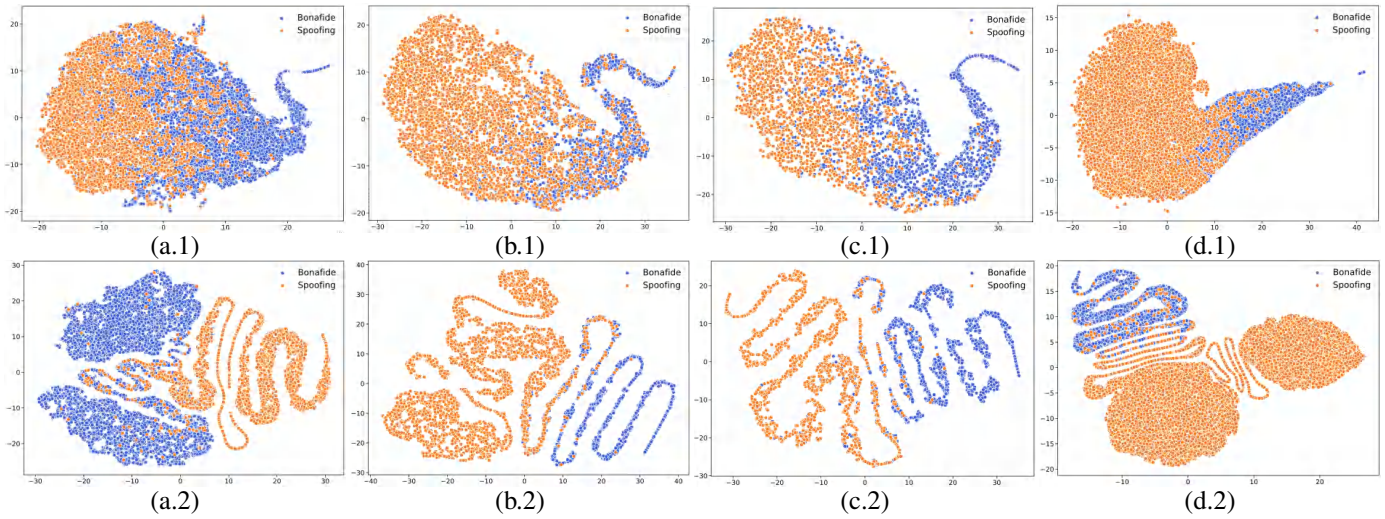


Fig. 6: Embedding cluster distributions based on t-SNE of the bonafide and spoofing samples under different target domains. (a) ITW; (b) JSUT; (c) FoR; (d) HABLA. (*.1) No adaptation baseline; (*.2) Our SHDA. Note that these plots emphasize the discriminability between *bonafide* and *spoofing* samples.

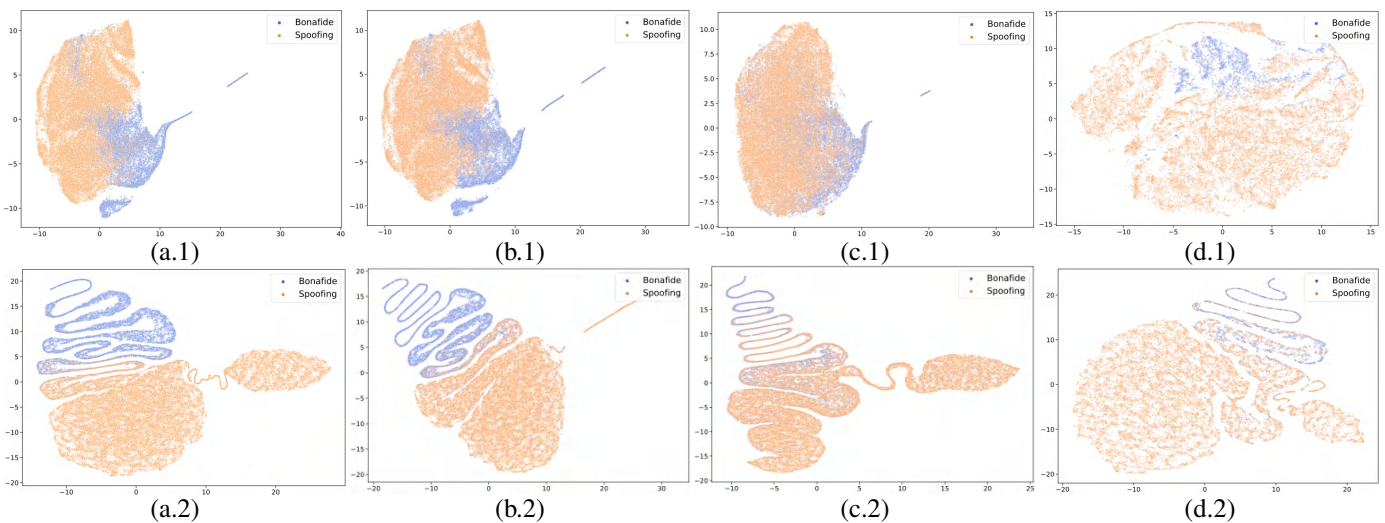


Fig. 7: t-SNE embedding distributions of bonafide and spoofing samples in Mandarin Chinese target domains: (a) CFAD clean, (b) CFAD codec, (c) CFAD noisy, (d) FMFCC-A. (*.1) No adaptation baseline; (*.2) SHDA. Note that these plots emphasize the discriminability between *bonafide* and *spoofing* samples.

algorithm impact on detecting spoofing and bonafide samples.

In this experiment, we froze the model parameters of XLM-R. We compared the confusion matrices of the non-adaptive ECAPA-TDNN and its adaptive version using SHDA across eight target domains. The baseline system was trained with 19LA. Figures 5(a.1) to 5(h.1) show the confusion matrices of the baseline under these target domains. In cross-domain scenario, the non-adaptive anti-spoofing model tended to identify all samples as spoofing speech. Figures 5(a.2) to 5(h.2) show the performance of SHDA in the target domains. We observe that after applying SHDA adaptation, most bonafide and spoofing samples are correctly identified in each scenario.

3) Comparisons of the Embedding Cluster: To further investigate how SHDA improves model performance in the target domain, we used t-SNE [40] to visualize the embeddings extracted by ECAPA-TDNN without adaptation and with SHDA. Other configurations remained consistent with those in Section IV-B2.

Fig. 6 shows the embedding clusters of the anti-spoofing model on ITW, JSUT, FoR, and HABLA. The blue and yellow samples represent the bonafide and spoofing samples, respectively. In Figures 6(a.1) to 6(d.1), there are obvious

overlaps between the embeddings of spoofing and bonafide samples extracted by the non-adaptive model. This overlap interferes with the anti-spoofing model's ability to distinguish the two categories. In contrast, the model with SHDA effectively alleviates this problem, as shown in Figures 6(a.2) to 6(d.2). We further analyzed the embedding clusters across four challenging Chinese data sets, as illustrated in Fig. 7. In scenarios with significant language differences, embeddings from models without adaptation showed substantial overlap between different classes, with their cluster centroids positioned very closely. However, after applying the proposed SHDA algorithm, this issue was mitigated in most scenarios, reducing both sample overlap and the proximity of cluster centroids across different categories.

4) Comparisons of the Detection Error Trade-off Curve: We further used the detection error trade-off (DET) curve to visually illustrate model performance across various thresholds. We assessed the baseline system, three well-known adaptive algorithms, and SHDA across eight cross-domain scenarios.

Fig. 8 shows the DET curves of various adaptive algorithms across eight target domains. Observations reveal that existing adaptive algorithms, such as AUDA, effectively reduce miss

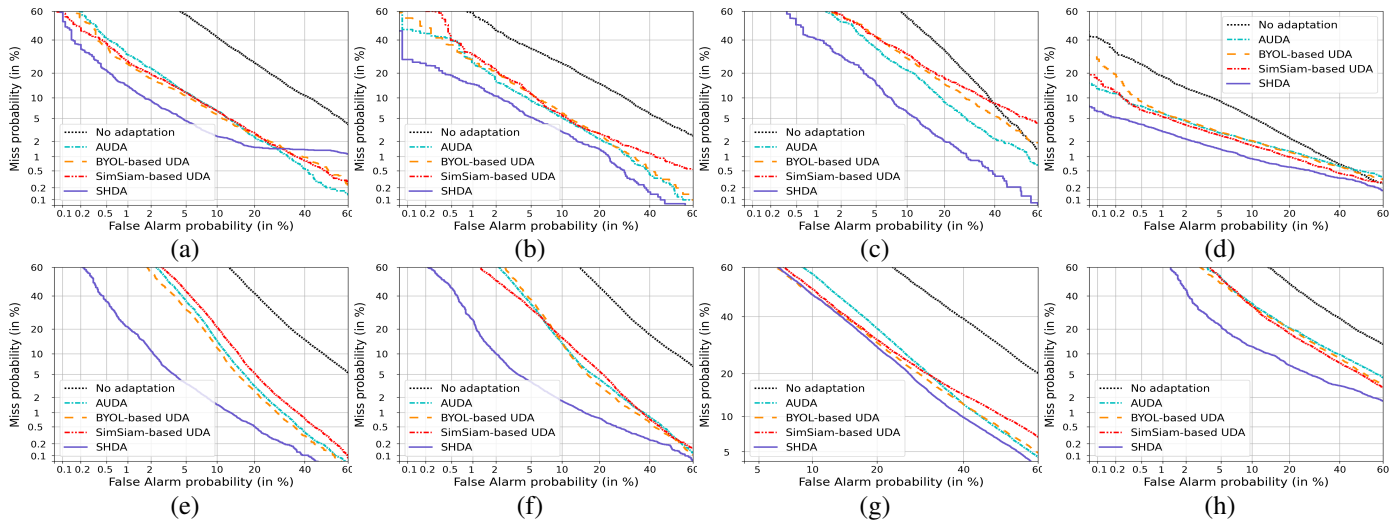


Fig. 8: DET curves for different anti-spoofing models across eight evaluation data sets. (a) ITW; (b) JSUT; (c) FoR; (d) HABLA; (e) CFAD clean; (f) CFAD codec; (g) CFAD noisy; (h) FMFCC-A.

and false alarm rates in the target domain. Nonetheless, the effectiveness of these traditional UDA algorithms is constrained under complex conditions. In contrast, the SHDA algorithm, shown by the solid purple line, consistently achieves the lowest miss and false alarm rates across all thresholds. Additionally, we observe that the adaptive performance of all UDA algorithms is affected in Fig. 8(g). We believe this is because CFAD noisy differs significantly from the source domains, leading to the negative transfer problem [82] in the UDA algorithm and ultimately impacting its adaptation performance. This underscores the vital role of the domain attention mechanism in improving cross-domain anti-spoofing.

V. DISCUSSION

Here, we first explored the working principle of SHDA. Next, we conducted ablation studies and analyzed the impact of hyper-parameter changes. Then, we evaluated the model's performance across various scenarios and components. Finally, we assessed the algorithm's actual training speed.

A. Visualizations of the Domain Attention Mechanism

To understand how the domain attention mechanism helps with the domain mismatches in anti-spoofing, we focus on three crucial questions: 1) *Why is domain attention needed?* 2) *What is its effect on the model?* 3) *How does it select important transfers?* We used 19LA, LJSpeech, and FakeAVCeleb as the source domains, and ITW and CFAD clean as the target domains to explore these aspects.

1) **Necessity: Why is domain attention needed:** First, we trained an anti-spoofing model without adaptation under the three source domains. We then applied this model to extract embeddings from samples across these domains. Finally, we visualized these embeddings using t-SNE, as shown in Fig. 9. The visualizations highlight varying domain discrepancies between the different source domains and the target domain. Taking the ITW experiment as an example, as shown in Fig. 9(a), the target spoofed samples exhibit smaller discrepancies with 19LA and FakeAVCeleb but larger discrepancies with most LJSpeech samples. A similar phenomenon appears in

CFAD clean, as shown in Fig. 9 (b), though with a distinct focus. These findings confirm previous research indicating that domain discrepancies vary among spoofing samples [38], [39], [41]. This variation underscores the need for selective transmission allocation between different source and target domains, rather than indiscriminately aligning them.

2) **Impact: What is the effect of domain attention on the model:** In this experiment, the configurations for data, model, and embedding extraction followed the protocols outlined in Section V-A1, with the addition of SHDA to adapt the model to the target domains. Taking ITW as an example, as shown in Fig. 9(a.2), after SHDA adaptation, the majority of the target samples (illustrated as black plots) are closely clustered around 19LA and FakeAVCeleb, with only a few near LJSpeech. This is because, as shown in Fig. 9(a.1), the cluster distribution of most target samples is closer to 19LA and FakeAVCeleb. A similar clustering phenomenon is observed in CFAD clean, as shown in Fig. 9(b.2). This result confirms that SHDA focuses on aligning cross-domain samples with close probability distances rather than indiscriminately aligning all cross-domain samples.

3) **Transmission selection mechanism: How does domain attention select important transfers:** In this experiment, we first randomly selected eight bonafide and eight spoofing audio samples from each source domain to create a source mini-batch. Subsequently, 24 bonafide and 24 spoofing samples were chosen from the target domain to form a target mini-batch. Then, we employed the SHDA algorithm to compute the transport coupling matrix. Finally, based on this matrix, we established the transport plans between the speech samples of the source and target domains.

Experimental results are shown in Fig. 10. In Figures 10(a.1) and (b.1), samples from 19LA, LJSpeech, and FakeAVCeleb are marked in blue, green, and orange on the y-axis for easy distinction. Taking ITW as an example, in Figures 10(a.2) and (a.3), the lines of different colors represent the transports between the target and the corresponding source domain samples. In the transport coupling of Fig. 10(a.1), SHDA exhibits distinct preferences: it primarily transfers bonafide audio between ITW (the target domain) and the bonafide audio

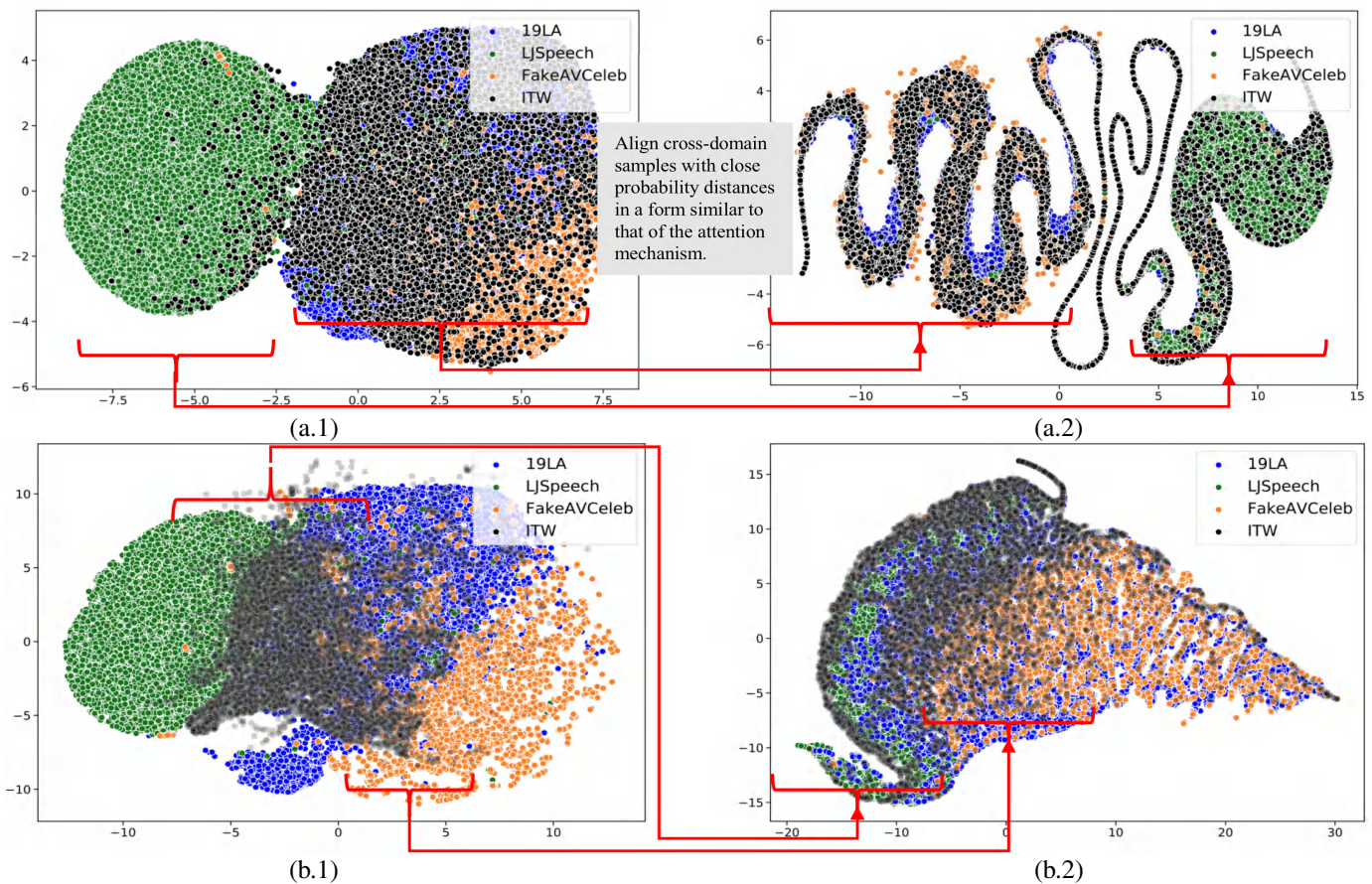


Fig. 9: Distributions of embedding clusters for the source and target domains. 19LA, LJSpeech, and FakeAVCeleb were used as source domains. ITW (a) and CFAD clean (b) were used as target domains. (*.1) Clusters without adaptation; (*.2) Clusters with SHDA adaptation. Note that these plots emphasize the discrepancies between the *source* and *target* domains.

from LJSpeech and FakeAVCeleb, resulting in a predominance of green and orange transports, as shown in Fig. 10(a.2). Conversely, it prioritizes the transfer of spoofing samples between ITW and the spoofing speech from 19LA and FakeAVCeleb, leading to an increase in blue and yellow transports with fewer green ones, as shown in Fig. 10(a.3). A similar phenomenon is observed in CFAD clean, as shown in Fig. 10(b), though it emphasizes different transfers.

In summary, this experiment demonstrates that SHDA can direct the model to focus on aligning target domain samples with source domains that have minor domain discrepancies. This ability to guide the model in learning ‘important’ knowledge is similar to the well-known attention mechanism.

B. Effects of Modules

This section evaluates the effectiveness of various components in our algorithm, including the multi-source domain mixed training strategy, the proposed SHDA, and the fine-tuning of the XLM-R model. Other configurations were aligned with Section IV-B. ITW and CFAD clean were selected as the target domains. The experimental results are presented in Table V. By comparing A1, A2, and A3 in this table, we observe that the mixed source domain training strategy reduced the EER from 22.63% to 11.32% under ITW. Implementing the SHDA strategy further decreased the EER of the B1 system

TABLE V: Ablation studies on the proposed anti-spoofing algorithm. ITW and CFAD clean were selected as the target domains. EER (%) was used as the evaluation metric in these experiments.

ID	Fine-tuning	SHDA	Source-1	Source-2	Source-3	ITW	CFAD clean
A1	✗	✗	✓	✗	✗	22.63	27.88
A2	✗	✗	✓	✓	✗	13.76	14.35
A3	✗	✗	✓	✓	✓	11.32	13.17
B1	✗	✓	✓	✓	✓	4.77	4.23
C1	✓	✓	✓	✓	✓	2.98	3.96

to 4.77%. Finally, by updating the model parameters of XLM-R during training, the C1 system achieved an EER of 2.98% on ITW. The same trend was observed in CFAD clean.

C. Effect of Hyper-Parameters

As described in Section III-D, the proposed adaptive anti-spoofing algorithm consists of two steps for loss calculation, making it necessary to explore the optimal hyper-parameters for each step. In Step 1, the SHDA algorithm calculates the optimal transport coupling matrix, which involves tuning the hyper-parameters α , β , and σ . In Step 2, the proposed adaptive anti-spoofing algorithm must determine the optimal weight (η) for the classification and SHDA loss functions. In this experiment, ITW and CFAD clean were selected as the target domains. The model parameters of XLM-R were frozen. Other configurations remained consistent with those in Section IV-B.

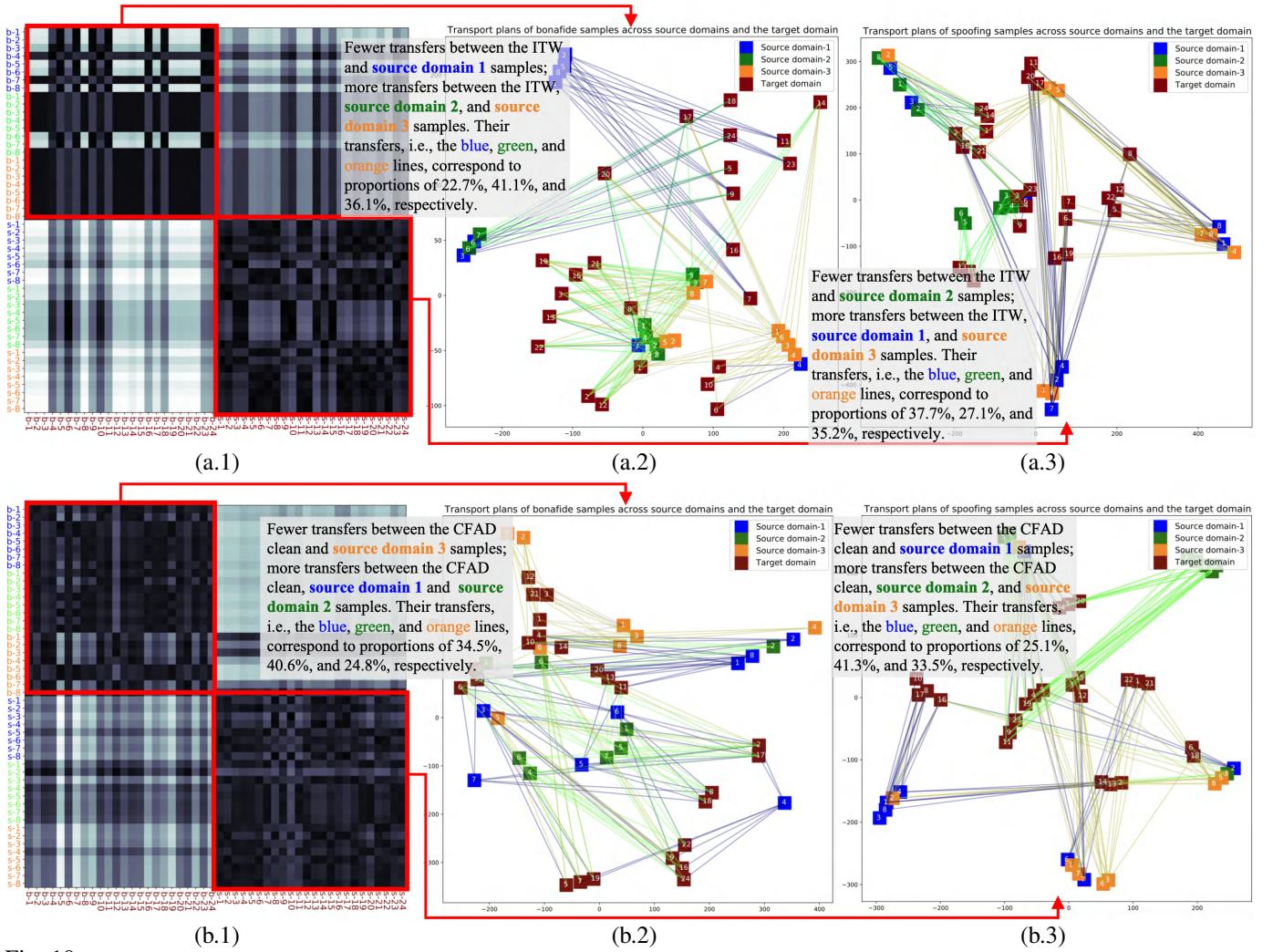


Fig. 10: Visualizations of transport couplings and plans for the proposed SHDA under ITW (a) and CFAD clean (b). (*.1) Transport coupling matrix for the source domains and the target domain. The y-axis of the matrix represents samples from the three source domains, distinguished by blue, green, and orange, while the x-axis represents samples from the target domain. On the axes, ‘b’ represents a bonafide sample, and ‘s’ denotes a spoofing sample. The numbers following these letters indicate the sample index. (*.2) Transport plans for bonafide samples from the source domains and the target domain. Samples from different source domains are shown in the figure using blue, green, and orange. Additionally, their transport plans are illustrated with lines in the corresponding colors. (*.3) Transport plans for spoofing samples from the source and target domains.

TABLE VI: Effect of hyper-parameters (α , β , and σ) on adaptation performance for ITW and CFAD clean. EER (%) was used as the evaluation metric in these experiments.

Adaptation hyper-parameters			ITW	CFAD clean
$\alpha = 0.001$	$\beta = 0.001$	$\sigma = 10.0$	7.10	9.41
$\alpha = 0.01$	$\beta = 0.001$	$\sigma = 10.0$	4.88	4.78
$\alpha = 0.1$	$\beta = 0.001$	$\sigma = 10.0$	4.77	4.23
$\alpha = 1.0$	$\beta = 0.001$	$\sigma = 10.0$	6.54	7.55
$\alpha = 10.0$	$\beta = 0.001$	$\sigma = 10.0$	7.78	10.79
$\alpha = 0.1$	$\beta = 0.0$	$\sigma = 10.0$	5.03	4.52
$\alpha = 0.1$	$\beta = 0.00001$	$\sigma = 10.0$	4.82	4.35
$\alpha = 0.1$	$\beta = 0.0001$	$\sigma = 10.0$	4.81	4.42
$\alpha = 0.1$	$\beta = 0.001$	$\sigma = 10.0$	4.77	4.23
$\alpha = 0.1$	$\beta = 0.01$	$\sigma = 10.0$	4.93	4.43
$\alpha = 0.1$	$\beta = 0.1$	$\sigma = 10.0$	6.26	4.95
$\alpha = 0.1$	$\beta = 0.001$	$\sigma = 1.0$	6.41	8.02
$\alpha = 0.1$	$\beta = 0.001$	$\sigma = 5.0$	4.84	4.82
$\alpha = 0.1$	$\beta = 0.001$	$\sigma = 10.0$	4.77	4.23
$\alpha = 0.1$	$\beta = 0.001$	$\sigma = 50.0$	4.78	5.28
$\alpha = 0.1$	$\beta = 0.001$	$\sigma = 100.0$	5.04	5.36

First, according to Equations (14) and (20), we evaluated the impact of variations in α , β , and σ on the adaptation results.

TABLE VII: EERs (%) and F1 scores (%) of the proposed algorithm with different SHDA weights under ITW and CFAD clean.

	η	0.000	0.001	0.01	0.1	0.2	0.3	0.4	0.6	0.8	1.0
ITW	EER↓	11.32	7.11	4.82	4.77	5.72	5.72	6.97	7.18	6.94	9.34
	F1↑	85.63	92.66	94.98	95.41	94.49	94.38	93.04	92.85	92.97	90.64
CFAD clean	η	0.000	0.001	0.01	0.1	0.2	0.3	0.4	0.6	0.8	1.0
	EER↓	13.17	7.67	5.62	4.23	4.87	5.25	6.73	4.76	6.59	7.57
	F1↑	87.22	92.84	94.70	95.74	95.28	94.84	94.26	95.38	94.98	92.07

In these experiments, η was fixed at 0.1. The experimental results in Table VI indicate that when $\alpha = 0.1$, $\beta = 0.001$, and $\sigma = 10$, SHDA achieved the best adaptation performance, with EERs of 4.77% and 4.23% on ITW and CFAD clean, respectively. Next, according to Eq. (18), SHDA needs to adjust η to balance classification and domain alignment. Table VII shows that when η ranged between 0.01 and 0.1, the system achieved optimal performance, recording an EER of 4.77% and an F1-score of 95.41%.

TABLE VIII: EER (%) comparisons of the source model, the proposed SHDA, the retrained model using target data (RM-T), and the retrained model using both source and target data (RM-ST). Please note that SHDA did not use target labels, nor did the source model, whereas the other algorithms did.

Target domain	Source model	SHDA	RM-T	RM-ST
ITW	11.32	4.77	0.33	0.64
CFAD clean	13.17	4.23	1.61	1.16

D. Comparisons of Models Trained on Target Data Using Supervised Learning

This section further examines the performance gap between the proposed unsupervised algorithm and its theoretical upper bound. We compared SHDA's performance with two models retrained on different domains using supervised learning. These retrained models (RMs) are defined as follows: (1) *RM-T*: Retrains the anti-spoofing model using target domain data and labels. (2) *RM-ST*: Retrains the model using both target and source domain data and labels. Note that the proposed SHDA algorithm uses only target data, whereas RM-T and RM-ST require both target domain data and labels. Therefore, their performance represents the theoretical upper bound of domain transfer. In this experiment, ITW and CFAD clean were selected as target domains. Other configurations remained consistent with those in Section IV-B.

The experimental results are shown in Table VIII. Two key findings emerge: (1) Compared to the source model (i.e., a model trained on only three source domains), SHDA substantially reduced the EER in each target domain. On ITW and CFAD clean, SHDA decreased the EER from 11.32% to 4.77% and from 13.17% to 4.23%, respectively. (2) Compared to RMs trained with target labels, SHDA still exhibited a performance gap; however, this gap was much smaller than that of the source models. In sum, although SHDA has room for improvement in matching the performance of models retrained with supervised learning, it offers greater flexibility by eliminating the need for target domain data annotation, making it well-suited for real-world applications.

E. Performance of Adapted Models in Unseen Scenarios

This section evaluates the effectiveness of adaptive anti-spoofing algorithms in unseen scenarios. Specifically, we used source domains and unlabeled seen target domains to train the UDA algorithms and tested them in unseen scenarios. JSUT and FoR were selected as seen target domains, while ITW and CFAD clean were chosen as unseen target domains. Other configurations remained consistent with those in Section IV-B. The experimental results are shown in Table IX. Compared with the baseline model, the performance of the adaptive anti-spoofing algorithms in unseen scenarios was improved. For example, when FoR was the target domain, AUDA reduced the model's EER on the unseen CFAD clean from 27.88% to 11.08%. This improvement occurred because the UDA algorithm helps the anti-spoofing model learn more distinctions between bonafide and spoofing samples from the seen target domains, thereby improving its generalization. In contrast, the proposed SHDA achieved an EER of 7.33% on the unseen CFAD clean under the same experimental configuration. These

TABLE IX: EER (%) comparisons of various adaptive anti-spoofing algorithms in unseen scenarios. In this experiment, UDA algorithms used only seen target data during training but were tested in unseen scenarios to further evaluate their generalization.

Model	Seen target domain	Unseen domains	
	JSUT	ITW	CFAD clean
Wav2vec + ECAPA-TDNN	17.12	22.63	27.88
AUDA	7.16	16.51	23.72
BYOL-based UDA	7.96	17.86	22.78
SimSiam-based UDA	7.28	27.61	28.80
Proposed SHDA	5.16	11.10	13.59

Model	Seen target domain	Unseen domains	
	FoR	ITW	CFAD clean
Wav2vec + ECAPA-TDNN	24.69	22.63	27.88
AUDA	14.73	8.89	11.08
BYOL-based UDA	17.36	8.51	22.29
SimSiam-based UDA	18.65	8.94	11.89
Proposed SHDA	7.91	8.26	7.33

results further demonstrate the effectiveness of the proposed algorithm in enhancing the model's generalization.

F. Performance of Adapted Models in Source Domains

This section evaluates the impact of SHDA on source domains after adaptation. 19LA, LJSpeech, and FakeAVCeleb were selected as source domains, while the other eight data sets were designated as target domains. We compared the model's performance in the three source domains before and after applying SHDA. Since LJSpeech and FakeAVCeleb do not provide predefined training and test set divisions, we partitioned them using the same method described in Section IV-A. Other configurations remained consistent with those in Section IV-B.

The experimental results in Fig. 11 show that, in most cross-domain experiments, the model with SHDA performed comparably to or better than the source model on 19LA and LJSpeech, while its performance decreased on FakeAVCeleb. This phenomenon can be attributed to the fact that 19LA and LJSpeech cover a greater variety of attack categories (six and seven, respectively), whereas FakeAVCeleb contains only one attack. After applying SHDA, the anti-spoofing model learns the discriminative features present in the target data. In scenarios with more diverse attacks, such as 19LA and LJSpeech, this knowledge complements the discriminative capability learned from source data, thereby correcting previously misidentified samples. In contrast, in a single-attack scenario like FakeAVCeleb, this additional knowledge could interfere with the model's ability to discriminate between bonafide and spoofing samples in the source domain.

G. Generalization with Different Deep Feature Extractors and Encoders

This section evaluates the generalization of adaptive algorithms across different deep feature extractors and encoders. Following [83] and [84], we replaced the deep feature extractor with WavLM [14] and the encoder with ResNet-18 in our Wav2vec + ECAPA-TDNN architecture. As a result, two backbone models, WavLM + ECAPA-TDNN and Wav2vec

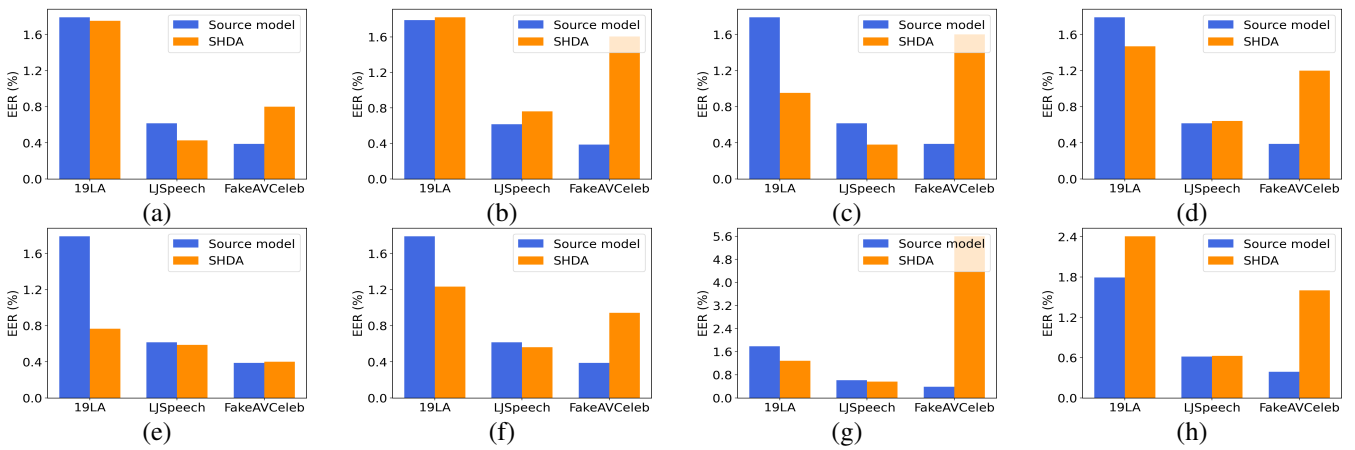


Fig. 11: The effect of applying the SHDA algorithm to different target domains on performance across the three source domains. The target domains include: (a) ITW, (b) JSUT, (c) FoR, (d) HABLA, (e) CFAD clean, (f) CFAD codec, (g) CFAD noisy, and (h) FMFCC-A.

TABLE X: EER (%) comparisons of adaptive anti-spoofing algorithms using WavLM as the deep feature extractor.

Target domain	WavLM + ECAPA-TDNN	AUDA	BYOL-based UDA	SimSiam-based UDA	SHDA
ITW	39.08	12.19	10.77	10.19	6.44
CFAD clean	29.97	22.63	19.38	23.62	15.30

TABLE XI: EER (%) comparisons of adaptive anti-spoofing algorithms using ResNet-18 as the encoder.

Target domain	Wav2vec + ResNet-18	AUDA	BYOL-based UDA	SimSiam-based UDA	SHDA
ITW	31.35	7.89	9.72	8.68	5.79
CFAD clean	24.88	12.81	11.18	11.83	5.22

+ ResNet-18, were formed. We then implemented different adaptive algorithms to assess their generalization. ITW and CFAD clean were selected as the target domains. Other configurations remained consistent with those in Section IV-B.

As shown in Table X, when using WavLM features, adaptive algorithms consistently improved performance in the target domains. Among them, SHDA achieved the best results in both scenarios, with EERs of 6.44% and 15.30%, respectively. Similarly, Table XI shows that with ResNet-18, adaptive algorithms again enhanced target domain performance, with SHDA outperforming others, achieving EERs of 5.79% and 5.22%, respectively. These experiments demonstrate SHDA’s generalization capability. However, employing a more advanced backbone model, such as our Wav2vec + ECAPA-TDNN, could further enhance the model’s ability to distinguish target bonafide and spoofing samples, thereby improving adaptation stability [82]. This also underscores the effectiveness of our backbone model.

H. Comparisons of Computational Efficiency

This section evaluates the actual training speed of different adaptive anti-spoofing algorithms. The training configurations remained consistent with those in Section IV-B. In this experiment, the CPU model used was the Intel 6240R, and the GPU model was the RTX 3090. The experimental results are shown in Table XII. The non-adaptive model required 1,275 seconds per epoch for training. In contrast, adaptive algorithms

TABLE XII: Epoch-wise training time (in seconds) for anti-spoofing algorithms. Although training times vary, inference time remains unchanged, as only the same single-branch structure is retained after training.

Wav2vec + ECAPA-TDNN	AUDA	BYOL-based UDA	SimSiam-based UDA	SHDA
1,275	1,662	3,491	2,297	1,735

required a multi-branch structure, increasing training time to approximately 1,662–3,491 seconds per epoch. Among them, the proposed SHDA required 1,735 seconds of training time per epoch, which is basically the same as that of AUDA. Nevertheless, the inference speed of these adaptive and non-adaptive algorithms remained the same, as adaptive models retained a single-branch structure after training. However, adaptive algorithms significantly improve performance in the target domain compared to the non-adaptive model.

VI. CONCLUSION

In this study, we propose SHDA, a novel domain attention mechanism designed for cross-domain audio anti-spoofing. SHDA addresses the challenge of transferring a model trained on multiple known spoofing data sets to an unknown spoofing data set. Its main idea is to assign optimal transfer weights to cross-domain sample pairs based on the extent of their domain discrepancies during the adaptation process. In SHDA, we first extract representations for different source and target domains, respectively, and calculate their overall transport cost matrix. Finally, we use the Sinkhorn algorithm to iteratively solve the optimal transport coupling, thereby determining the transport plan between cross-domain samples. This design allows SHDA to focus the model on transferring sample pairs with minor domain discrepancies, instead of compelling the alignment of spoofed speech samples with substantial domain discrepancies. Considering the different parameter optimization methods of OT and neural anti-spoofing models, we have developed a two-step optimization strategy to train SHDA end-to-end. Systematic cross-domain experiments demonstrate that the proposed SHDA surpasses the performance of the current state-of-the-art model by 40%.

Although the SHDA algorithm successfully reduces domain mismatches in anti-spoofing, it still requires the manual tuning of the weights for the SHDA and cross-entropy loss functions. We plan to design an adaptive weighting strategy that automatically adjusts the trade-off between these loss functions based on varying scenarios in future work. Additionally, we observed that SHDA's performance could be affected in scenarios with extremely large domain discrepancies. This occurs because, when model representations are insufficient, the transport cost struggles to accurately reflect the true differences between cross-domain samples. In future work, we plan to constrain the transport of target samples that deviate significantly from source samples to mitigate the impact of negative transport on overall alignment.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China (No. 2023YFB2603902), Tianjin Science and Technology Program (No. 21JCZJJC00190), and National Natural Science Foundation of China (No. 62176181). The work was partly supported by BUT IGA project No. FIT-S-23-8278.

REFERENCES

- [1] A. Sizov, E. Khoury, T. Kinnunen, Z. Wu, and S. Marcel, "Joint speaker verification and antispoofing in the i -vector space," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 821–832, 2015.
- [2] R. Zhang, J. Wei, W. Lu, L. Wang, M. Liu, L. Zhang, J. Jin, and J. Xu, "Aret: Aggregated residual extended time-delay neural networks for speaker verification." in *Proc. INTERSPEECH*, 2020, pp. 946–950.
- [3] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch *et al.*, "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [4] A. Nautsch, X. Wang, N. Evans, T. H. Kinnunen, V. Vestman, M. Todisco, H. Delgado, M. Sahidullah, J. Yamagishi, and K. A. Lee, "Asvspoof 2019: spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, 2021.
- [5] J. Sanchez, I. Saratxaga, I. Hernaez, E. Navas, D. Erro, and T. Raitio, "Toward a universal synthetic speech spoofing detection using phase information," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 810–820, 2015.
- [6] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, 2017.
- [7] S. Jelil, R. K. Das, S. M. Prasanna, and R. Sinha, "Spoof detection using source, instantaneous frequency and cepstral features." in *Proc. INTERSPEECH*, 2017, pp. 22–26.
- [8] M. Yang, K. Zheng, X. Wang, Y. Sun, and Z. Chen, "Comparative analysis of asv spoofing countermeasures: Evaluating res2net-based approaches," *IEEE Signal Processing Letters*, 2023.
- [9] X. Li, N. Li, C. Weng, X. Liu, D. Su, D. Yu, and H. Meng, "Replay and synthetic speech detection with res2net architecture," in *Proc. ICASSP*. IEEE, 2021, pp. 6354–6358.
- [10] J. Kim and S. M. Ban, "Phase-aware spoof speech detection based on res2net with phase network," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.
- [11] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *Proc. ICASSP*. IEEE, 2022, pp. 6367–6371.
- [12] W. Ge, X. Wang, J. Yamagishi, M. Todisco, and N. Evans, "Spoofing attack augmentation: can differently-trained attack models improve generalisation?" in *Proc. ICASSP*. IEEE, 2024, pp. 12 531–12 535.
- [13] A. Baeovski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, vol. 33, 2020, pp. 12 449–12 460.
- [14] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [15] Y. Qian, N. Chen, H. Dinkel, and Z. Wu, "Deep feature engineering for noise robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1942–1955, 2017.
- [16] I. Himawan, F. Villavicencio, S. Sridharan, and C. Fookes, "Deep domain adaptation for anti-spoofing in speaker verification systems," *Computer Speech & Language*, vol. 58, pp. 377–402, 2019.
- [17] N. Müller, P. Czempin, F. Diekmann, A. Froggyar, and K. Böttinger, "Does Audio Deepfake Detection Generalize?" in *Proc. INTERSPEECH*, 2022, pp. 2783–2787.
- [18] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132–157, 2020.
- [19] A. Cohen, I. Rimon, E. Aflalo, and H. H. Permuter, "A study on data augmentation in voice anti-spoofing," *Speech Communication*, vol. 141, pp. 56–67, 2022.
- [20] H. Tak, M. Kamble, J. Patino, M. Todisco, and N. Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *Proc. ICASSP*. IEEE, 2022, pp. 6382–6386.
- [21] L. Zhang, K. A. Lee, L. Zhang *et al.*, "Cpaug: Refining copy-paste augmentation for speech anti-spoofing," in *Proc. ICASSP*. IEEE, 2024, pp. 10 996–11 000.
- [22] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.
- [23] G. Lin, W. Luo, D. Luo, and J. Huang, "One-class neural network with directed statistics pooling for spoofing speech detection," *IEEE Transactions on Information Forensics and Security*, 2024.
- [24] J. Lu, Y. Zhang, W. Wang, Z. Shang, and P. Zhang, "One-class knowledge distillation for spoofing speech detection," in *Proc. ICASSP*. IEEE, 2024, pp. 11 251–11 255.
- [25] X. Wang and J. Yamagishi, "Can large-scale vocoded spoofed data improve speech spoofing countermeasure with a self-supervised front end?" in *Proc. ICASSP*. IEEE, 2024, pp. 10 311–10 315.
- [26] Y. Xie, H. Cheng, Y. Wang, and L. Ye, "Domain generalization via aggregation and separation for audio deepfake detection," *IEEE Transactions on Information Forensics and Security*, 2023.
- [27] —, "Learning a self-supervised domain-invariant feature representation for generalized audio deepfake detection," in *Proc. INTERSPEECH*, vol. 2023, 2023, pp. 2808–2812.
- [28] X. Wang and J. Yamagishi, "Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.
- [29] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 5, pp. 1–46, 2020.
- [30] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1794–1809, 2018.
- [31] H. Wang, H. Dinkel, S. Wang, Y. Qian, and K. Yu, "Cross-domain replay spoofing attack detection using domain adversarial training," in *Proc. INTERSPEECH*, 2019, pp. 2938–2942.
- [32] —, "Dual-adversarial domain adaptation for generalized replay attack detection," in *Proc. INTERSPEECH*, 2020, pp. 1086–1090.
- [33] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*. PMLR, 2015, pp. 1180–1189.
- [34] Y. Liu, Y. Chen, M. Gou, C.-T. Huang, Y. Wang, W. Dai, and H. Xiong, "Towards unsupervised domain generalization for face anti-spoofing," in *Proc. CVPR*, 2023, pp. 20 654–20 664.
- [35] R. Zhang, J. Wei, X. Lu, W. Lu, D. Jin, L. Zhang, Y. Ji, and J. Xu, "Self-supervised learning based domain regularization for mask-wearing speaker verification," *Speech Communication*, p. 102953, 2023.
- [36] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," in *Proc. NeurIPS*, vol. 19, 2006.
- [37] K. A. Lee, Q. Wang, and T. Koshinaka, "The coral+ algorithm for unsupervised domain adaptation of plda," in *Proc. ICASSP*. IEEE, 2019, pp. 5821–5825.
- [38] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans *et al.*, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

- [39] Y. Xie, H. Cheng, Y. Wang, and L. Ye, "Single domain generalization for audio deepfake detection," in *Proc. IJCAI workshop*, 2023, pp. 58–63.
- [40] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.
- [41] X. Yan, J. Yi, J. Tao, C. Wang, H. Ma, T. Wang, S. Wang, and R. Fu, "An initial investigation for detecting vocoder fingerprints of fake audio," in *Proc. DDAM*, 2022, pp. 61–68.
- [42] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, "Maximum density divergence for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3918–3930, 2020.
- [43] Y. Xie, H. Cheng, Y. Wang, and L. Ye, "Learning A Self-Supervised Domain-Invariant Feature Representation for Generalized Audio Deepfake Detection," in *Proc. INTERSPEECH*, 2023, pp. 2808–2812.
- [44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, vol. 30, 2017.
- [45] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Proc. NeurIPS*, vol. 26, 2013.
- [46] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Proc. NeurIPS*, vol. 30, 2017.
- [47] D. A. Edwards, "On the kantorovich–rubinstein theorem," *Expositiones Mathematicae*, vol. 29, no. 4, pp. 387–398, 2011.
- [48] A. Javed, K. M. Malik, H. Malik, and A. Irtaza, "Voice spoofing detector: A unified anti-spoofing framework," *Expert Systems with Applications*, vol. 198, p. 116770, 2022.
- [49] M. Todisco, H. Delgado, and N. W. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients," in *Proc. Odyssey*, vol. 2016, 2016, pp. 283–290.
- [50] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *Proc. SLT*. IEEE, 2018, pp. 1021–1028.
- [51] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks," in *Proc. INTERSPEECH*, 2017, pp. 82–86.
- [52] M. G. Kumar, S. R. Kumar, M. Saranya, B. Bharathi, and H. A. Murthy, "Spoof detection using time-delay shallow neural network and feature switching," in *Proc. ASRU*. IEEE, 2019, pp. 1011–1017.
- [53] J.-H. Choi, J.-Y. Yang, Y.-R. Jeoung *et al.*, "Hyu submission for the sasv challenge 2022: Reforming speaker embeddings with spoofing-aware conditioning," in *Proc. INTERSPEECH*, 2022, pp. 2873–2877.
- [54] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdn based speaker verification," in *Proc. INTERSPEECH*, 2020, pp. 3830–3834.
- [55] Y. Guo, H. Huang, X. Chen, H. Zhao, and Y. Wang, "Audio deepfake detection with self-supervised wavlm and multi-fusion attentive classifier," in *Proc. ICASSP*. IEEE, 2024, pp. 12 702–12 706.
- [56] Y. Zhang, J. Lu, Z. Shang, W. Wang, and P. Zhang, "Improving short utterance anti-spoofing with aasist2," in *Proc. ICASSP*. IEEE, 2024, pp. 11 636–11 640.
- [57] R. Reimao and V. Tzerpos, "For: A dataset for synthetic speech detection," in *Proc. International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE, 2019, pp. 1–10.
- [58] J. Frank and L. Schönherr, "Wavefake: A data set to facilitate audio deepfake detection," in *Proc. NeurIPS Datasets and Benchmarks*, vol. 1, 2021.
- [59] H. Ma, J. Yi, C. Wang, X. Yan, J. Tao, T. Wang, S. Wang, and R. Fu, "Cfad: A chinese dataset for fake audio detection," *Speech Communication*, p. 103122, 2024.
- [60] Z. Zhang, Y. Gu, X. Yi, and X. Zhao, "Fmfcc-a: a challenging mandarin dataset for synthetic speech detection," in *Proc. IWDW*. Springer, 2021, pp. 117–131.
- [61] P. A. Tamayo Flórez, R. Manrique, and B. Pereira Nunes, "HABLA: A Dataset of Latin American Spanish Accents for Voice Anti-spoofing," in *Proc. INTERSPEECH*, 2023, pp. 1963–1967.
- [62] L. Zhang, X. Wang, E. Cooper, N. Evans, and J. Yamagishi, "The partialspoof database and countermeasures for the detection of short fake speech segments embedded in an utterance," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 813–825, 2022.
- [63] X. Zhang, J. Yi, C. Wang, C. Y. Zhang, S. Zeng, and J. Tao, "What to remember: Self-adaptive continual learning for audio deepfake detection," in *Proc. AAAI*, vol. 38, no. 17, 2024, pp. 19 569–19 577.
- [64] R. K. Das, J. Yang, and H. Li, "Data augmentation with signal companding for detection of logical access attacks," in *Proc. ICASSP*. IEEE, 2021, pp. 6349–6353.
- [65] S. Ding, Y. Zhang, and Z. Duan, "Samo: Speaker attractor multi-center one-class learning for voice anti-spoofing," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.
- [66] Y. Liu, Y. Chen, W. Dai, M. Gou, C.-T. Huang, and H. Xiong, "Source-free domain adaptation with domain generalized pretraining for face anti-spoofing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [67] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, and Y. Fu, "Semi-supervised domain adaptive structure learning," *IEEE Transactions on Image Processing*, vol. 31, pp. 7179–7190, 2022.
- [68] Z. Chen, S. Wang, and Y. Qian, "Self-supervised learning based domain adaptation for robust speaker verification," in *Proc. ICASSP*. IEEE, 2021, pp. 5834–5838.
- [69] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, 2016.
- [70] B. B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty, "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation," in *Proc. ECCV*, 2018, pp. 447–463.
- [71] H.-Y. Lin, H.-H. Tseng, X. Lu, and Y. Tsao, "Unsupervised noise adaptive speech enhancement by discriminator-constrained optimal transport," in *Proc. NeurIPS*, vol. 34, 2021, pp. 19 935–19 946.
- [72] Z. Lv, S. Zhang, K. Tang, and P. Hu, "Fake audio detection based on unsupervised pretraining models," in *Proc. ICASSP*. IEEE, 2022, pp. 9231–9235.
- [73] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2019.
- [74] X. Lu, P. Shen, Y. Tsao, and H. Kawai, "Unsupervised neural adaptation model based on optimal transport for spoken language identification," in *Proc. ICASSP*. IEEE, 2021, pp. 7213–7217.
- [75] H. Khalid, S. Tariq, M. Kim, and S. S. Woo, "FakeAVCeleb: A novel audio-video multimodal deepfake dataset," in *Proc. NeurIPS 2021 Datasets and Benchmarks Track (Round 2)*, 2021.
- [76] Y. Ren, H. Peng, L. Li, X. Xue, Y. Lan, and Y. Yang, "Generalized voice spoofing detection via integral knowledge amalgamation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [77] H. Tak, J.-w. Jung, J. Patino, M. Kamble, M. Todisco, and N. Evans, "End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection," in *Proc. ASVspoof workshop*, 2021.
- [78] J.-B. Grill, F. Strub, F. Althé, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent—a new approach to self-supervised learning," in *Proc. NeurIPS*, vol. 33, 2020, pp. 21 271–21 284.
- [79] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. CVPR*, 2021, pp. 15 750–15 758.
- [80] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with rawnet2," in *Proc. ICASSP*. IEEE, 2021, pp. 6369–6373.
- [81] Z. Zhang, X. Yi, and X. Zhao, "Fake speech detection using residual network with transformer encoder," in *Proc. ACM IH&MMSec*, 2021, pp. 13–22.
- [82] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *Proc. CVPR*, 2019, pp. 2985–2994.
- [83] A. Ito and S. Horiguchi, "Spoofing attacker also benefits from self-supervised pretrained model," in *Proc. INTERSPEECH*, 2023, pp. 5346–5350.
- [84] Y. Xie, Z. Zhang, and Y. Yang, "Siamese network with wav2vec feature for spoofing speech detection," in *Proc. INTERSPEECH*, 2021, pp. 4269–4273.