Additive Adversarial Learning for Unbiased Authentication

Jian Liang[1*], Yuren Cao[1,2*], Chenbin Zhang[1,2], Shiyu Chang[3], Kun Bai[1], Zenglin Xu[2]
[1]Cloud and Smart Industries Group, Tencent, China
[2]University of Electronic Science and Technology of China
[3]MIT-IBM Watson AI Lab, IBM Research, USA
{joshualiang,laurenyrcao,kunbai}@tencent.com
ChenbinZhang@std.uestc.edu.cn, shiyu.chang@ibm.com, zenglin@gmail.com

Abstract

Authentication is a task aiming to confirm the truth between data instances and personal identities. Typical authentication applications include face recognition, person re-identification, authentication based on mobile devices and so on. The recently-emerging data-driven authentication process may encounter undesired biases, i.e., the models are often trained in one domain (e.g., for people wearing spring outfits) while required to apply in other domains (e.g., they change the clothes to summer outfits). To address this issue, we propose a novel two-stage method that disentangles the class/identity from domain-differences, and we consider multiple types of domain-difference. In the first stage, we learn disentangled representations by a one-versus-rest disentangle learning (OVRDL) mechanism. In the second stage, we improve the disentanglement by an additive adversarial learning (AAL) mechanism. Moreover, we discuss the necessity to avoid a learning dilemma due to disentangling causally related types of domain-difference. Comprehensive evaluation results demonstrate the effectiveness and superiority of the proposed method.

1. Introduction

Authentication considers the problem of whether the data instances match personal identities. There is a variety of authentication applications including biometric authentication [4, 22] (e.g. face recognition [41] and fingerprint verification [37]) and person re-identification [2, 43]. However, data-driven authentication process often suffers from undesired biases, i.e., domain-difference, which refers to the problem that a model is trained in one domain, but tested and verified in other domains. For example, in the field of person re-identification [2], the prediction may be compromised due to the seasonal outfits changing or the angle variation between a camera and a pedestrian.

	Class Group 1	Class Group 2	Class Group 3
Domain 1	Train	Test	Test
Domain 2	Test	Test	Train
Domain 3	Test	Train	Test

Table 1. An example of the assumptions of our problem.

Faced with the domain-difference problem between training and testing data, simply applying data-driven models may lead to undesired solutions that focus on the biases of domains, even if the training data is sufficient. To alleviate the aforementioned problem, this paper addresses the task of learning for unbiased authentication. For simplicity, we treat authentication as a recognition problem so that each identity corresponds to a class. We consider that there are multiple *domains* and multiple types of *domain-difference*, where a specific type of *domain-difference* may include multiple *domains*. For example, for person re-identification, *season* and *shooting angle* are two types of domain-difference, where *season* includes four domains: `spring`, `summer`, `autumn`, and `winter`, and *shooting angle* includes domains such as `front`, `back`, `side`, *etc*.

To better understand our problem, we present a toy example with only one type of domain difference in Table 1. In the training phase, for each group of classes, we have their data on only one domain. In other words, different domains do not share classes. In the testing phase, we need to do a recognition on data which corresponds to unseen ⟨class, domain⟩ combinations. Mathematically, the problem we attempt to tackle is related to domain adaptation [25, 8, 28, 32], but different from it, because domain adaptation allows source and target domains to share classes but provides no label on target domains. Domain adaptation has been extensively studied in the field of transfer learning [24, 33, 32, 19]. Our problem can be transformed

*Equal contribution from both authors.

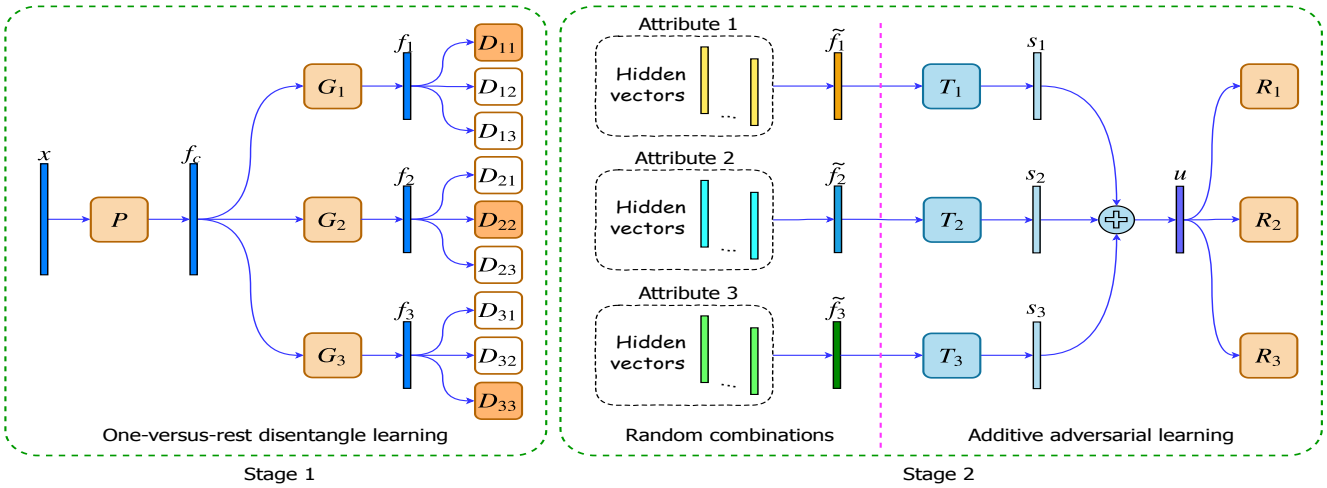

Figure 1. The architecture of our framework. Intuitively, our framework is constructed in a multi-task learning flavor. The output of each task is regarded as an attribute to learn. The attribute-disentanglement pipeline of our work consists of two stages. Stage 1 consists of multiple branches of networks, and each branch learns by a one-versus-rest disentangling mechanism. Stage 2 aims to make further improvements, and the key ideas are illustrated in Fig. 3. Best viewed in color.

into a domain adaptation problem if the data of testing domains are allowed to train without class labels. Thus, we refer to our problem as a generalized cross-domain recognition (GCDR) problem. Similar problems have also been investigated in the field of fairness-oriented machine learning (FML) approaches [7] which concern biases against demographic groups, such as racial minorities or women. FML approaches in this setting usually apply transfer learning methods as solutions as well. In this paper, we also apply transfer learning methods to learn unbiased representations. Specifically, to focus on the main issue, we simply apply symmetric transfer learning methods (see the definition described by Weiss *et al*. [33]).

In this paper, we propose a novel recognition method that learns disentangled representations to handle domain-difference to achieve an unbiased recognition. As shown in Table 1, for a specific group of classes, the classes are different, but the domain is the same. Therefore, it is feasible to learn an unbiased model that can classify classes while neglecting the effects imposed by domain-differences. We also assume that although we have the labels of domains and domain-difference types, how the domain-differences affect the data is unknown. For a data instance, its class and domain values are treated as its attributes. Our method learns unbiased representations by disentangling these attributes. The framework of our method is illustrated in Fig. 1, which consists of two stages. In stage 1, we propose a one-versus-rest disentangle learning (OVRDL) mechanism to map each instance into multiple hidden spaces. In each hidden space, we disentangle one attribute from others. In stage 2, since limited combinations of attribute values are included in the training data, we conduct a data augmentation to randomly combine attribute labels and concatenate their associated hidden feature vectors as new data samples. An additive adversarial learning (AAL) mechanism based on random concatenations of hidden features is proposed to further improve the disentanglement of stage 1. Intuitively, biases are removed by minimizing negative side-effects. We extend the discussion on how to avoid a learning dilemma due to disentangle causally related attributes. The experimental results on benchmark and real-world data sets demonstrate the effectiveness and superiority of our method. We also conduct ablation experiments to show the contribution of each component of our proposed framework.

2. Related Work

To learn unbiased representations from unknown features of domain-difference, there are three thrusts of methods to leverage existing transfer learning methods, which are also the typical solutions for representation-learning based FML. In this section, we review them as well as some other related work, and differentiate them from our work.

Eliminating the marginal-distribution differences The first family eliminates marginal-distribution differences between domains. This family of methods includes Transfer Component Analysis (TCA) [23], Deep Adaptation Network (DAN) [17], Reversing Gradient (RevGrad) [9], Adversarial Discriminative Domain Adaptation (ADDA) [29], among others. FML methods proposed by Goel *et al*. [10] and Zhang *et al*. [39] also fall into this category. Many FML methods adopt RevGrad, such as those proposed by Wadsworth *et al*. [31] and Beutel *et al*. [3].

Generating data with unseen ⟨class, domain⟩ combi-

nations The second family generates data samples associated with unseen ⟨class, domain⟩ combinations, such as ELEGANT [35], DNA-GAN [34], Multi-Level Variational Autoencoder (ML-VAE) [5], CausalGAN [14], ResGAN [27], SaGAN [40], among others. FML methods Fairness GAN [26] and FairGAN [36] also fall into this category. These methods generate synthetic data, then ordinary models can be trained on both real and the generated data.

Hybrid methods The third family performs both marginal-distribution-difference elimination and synthetic-data generation, such as Cross-Domain Representation Disentangler (CDRD) [15], Synthesized Examples for Generalized Zero-Shot Learning (SE-GZSL) [30], Disentangled Synthesis for Domain Adaptation (DiDA) [6], Attribute-Based Synthetic Network (ABS-Net) [18], among others. Madras *et al*. [20] proposed such a FML framework.

Other related work Such a phenomenon of grouped classes was also discussed by Bouchacourt *et al*. [5] and Zhao [42]. However, they did not provide learning methods to eliminate domain-differences. It was also discussed by Heinze-Deml and Meinshausen [12]. However, they assumed classes with various domains are already included in the training data. Yu *et al*. [38] also discussed the setting that classes were not necessarily shared by multiple source domains. However, their method assumes that all the ⟨class, domain⟩ combinations are included in the training data set.

Differences between the existing works and our proposed method Despite the achievements, existing approaches either do not handle the GCDR problem or cannot avoid the learning dilemma due to disentangling correlated types of domain-difference. In addition, most of the generative methods generate samples in the original data space. However, if an appropriate model-structure is captured, generating data in the original data space is not necessary, and it may cause additional errors during both data generation and learning on the generated data. The aforementioned concerns are addressed by our proposed method.

3. Methodology

This section details our proposed network. We first define notations and problem settings. Consider a data set $\mathcal{D} = \{(\mathbf{x}^i, y^i, \mathbf{h}^i)\}_{i=1}^n$ consisting of n independent samples. For the ith sample, $\mathbf{x}^i \in \mathbb{R}^d$ is a feature vector with d dimensions, $y^i \in \mathbb{Z}_+$ is a categorical class label of the recognition task, and $\mathbf{h}^i \in \mathbb{Z}_+^m$ is a vector consisting of m categorical domain attributes. For example, in the colored MNIST (C-MNIST) recognition (see the image examples in the Fig. 6 of Lu *et al*. [18]), $\mathbf{x}^i$ can be a colored image of digits with the size of 28×28, the class label denoted by y^i is a value in $\{0, 1, \ldots, 9\}$, the background color (denoted by h_1^i) and foreground color (denoted by h_2^i) of the image

(a) Training

(b) Testing

Figure 2. An experiment setting of C-MNIST with the background color as the domain-difference. Best viewed in color.

are the two types of attributes. The different combinations of background colors and foreground colors can form multiple domains. For the convenience of the presentation, we denote $\mathbf{a}^i = (y^i, \mathbf{h}^i) \in \mathbb{Z}_+^{(m+1)}$ as the generalized attribute vector of the sample i. We denote a_j^i as the jth element of $\mathbf{a}^i$, and $a_j^i \in \{1, 2, \ldots, k_j\}$, where k_j is the size of the set. Throughout the paper, we denote $[k]$ as the index set $\{1, 2, \ldots, k\}$.

In practice, samples of the data set $\mathcal{D}$ are usually incomplete. For the example shown in Fig. 2, one can observe images of digit 5 with the red background, and digit 2 with green background, while one wants to make predictions on images of 5 with the green background. Formally, we define the GCDR problem as follows.

Problem 1. *(Generalized Cross-Domain Recognition (GCDR)) Given a data set $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{a}^i)\}_{i=1}^n$, let $\mathcal{D}_\Omega$ be the partially observed training set. The goal of our learning problem is to train a model over examples with partially observed combinations of attribute values, and then generalize this model to the testing set $\mathcal{D}_{\bar{\Omega}}$ with missing combinations of attribute values.*

Denote the sets of combinations of attribute values for the training and testing sets as $\mathcal{C}_\Omega = \{[a_1^i, \ldots, a_{(m+1)}^i] : i \in \Omega\}$ and $\mathcal{C}_{\bar{\Omega}} = \{[a_1^i, \ldots, a_{(m+1)}^i] : i \in \bar{\Omega}\}$, respectively. We have the constraint that the two sets have no intersection, i.e.*, $\mathcal{C}_\Omega \bigcap \mathcal{C}_{\bar{\Omega}} = \emptyset$. In addition, for the training set, for each jth type of domain-difference, denote the class group corresponding to its rth domain as $\mathcal{G}_j^r = \{y^i : h_j^i = r, r \in [k_j], i \in \Omega\}$. We have the constraint that different domains do not share classes,* i.e.*, for each jth type of domain-difference, $\mathcal{G}_j^r \bigcap \mathcal{G}_j^{r'} = \emptyset$ for all $r, r' \in [k_j]$ and $r \neq r'$.*

The structure of our framework is based on the ABS-Net [18], and further novelly extends with these contributions: (1) a one-versus-rest disentangle learning mechanism, (2) an AAL mechanism to further improve disentangle performances, (3) an extended strategy to cease some disentangling processes to avoid a learning dilemma due to disentangling causally related attributes, which will be introduced in the following.

3.1. One-Versus-Rest Disentangle Learning

Our chief goal is to disentangle the class from all the types of domain-difference. Moreover, as an auxiliary, we also aim to disentangle each type of domain-difference from the class and other types of domain-difference. As mentioned previously, if we regard the class and all the types of domain-difference as attributes, we aim to disentangle each attribute from others. Therefore, we can develop a one-versus-rest strategy for each attribute to achieve two purposes: (1) to learn each attribute itself, and (2) to disentangle it from others.

Specifically, for each attribute $j \in [m+1]$, we learn the mapping from the raw-data space to a hidden space: $\mathbf{x} \rightarrow \mathbf{f}_j$ (here we omit the indices of sample order). In the hidden space, the two purposes above can be externalized as follows. (1) Samples associated with different categorical values of attribute j can be well separated, *i.e.*, $P(a_j \mid \mathbf{f}_j) = 1$. (2) The distribution of samples is independent with that of any other attribute j', *i.e.*, $P(a_{j'} \mid \mathbf{f}_j) = P(a_{j'})$, which can be achieved by an adversarial learning process.

As shown in Fig. 1, in stage 1, we construct a network to achieve the aforementioned purposes. For the feature vector $\mathbf{x}$ of a raw instance, it is transformed by an *input-feature transformation network* P into a hidden feature vector $\mathbf{f}_c$ which is further transformed by *attribute-feature learning networks* $G_1, \ldots, G_{m+1}$ into *attribute feature vectors* $\mathbf{f}_1, \ldots, \mathbf{f}_{m+1}$, respectively. For each attribute j, we expect the hidden space associated with the attribute feature vector $\mathbf{f}_j$ to achieve the two purposes above.

To achieve the aforementioned purposes, we develop a one-versus-rest disentangle learning (OVRDL) mechanism for each attribute. For each attribute j, we construct $(m+1)$ discriminative networks, $D_{j1}, \ldots, D_{j(m+1)}$. Each discriminative network is trying to discriminate between different categorical values of the associated attribute. We expect that the "diagonal network" D_{jj} learns directly and can correctly predict a_j, while the "non-diagonal" networks, $\{D_{jj'}\}_{j' \neq j}$, learn adversarially and cannot correctly predict $a_{j'}$. Following the adversarial learning mechanism proposed by Alexia [13], a brief version of the adversarial learning for the "non-diagonal" networks can be regarded as the following two alternative steps [13].

Step 1: fix G_j, and for each $j' \neq j$, optimize $D_{jj'}$ to let the outputs approximate $\tilde{\mathbf{a}}_{j'}$ which is the one-hot-coded vector of the target $a_{j'}$;

Step 2: fix $D_{jj'}$s for all $j' \neq j$, and optimize G_j to let the outputs approximate $(\mathbf{1} - \tilde{\mathbf{a}}_{j'})$.

Finally, we establish the OVRDL mechanism in stage 1. For learning each attribute, we optimize by

$$\min_{G_0, \{G_j\}, \{D_{jj}\}} \sum_{i \in \Omega} w_j \mathcal{L}_{at}(D_{jj}(G_j(G_0(\mathbf{x}^i))), \tilde{\mathbf{a}}_j^i), \quad (1)$$

where $\mathcal{L}_{at}$ is the loss function for the attribute learning, $\tilde{\mathbf{a}}_j^i$ is the one-hot encoded vector of a_j^i, and w_j is the weight for the jth attribute, $j \in [m+1]$. For discriminating domains for each type of domain-difference, we optimize by

$$\min_{\{D_{jj'}\}} \sum_{i \in \Omega} \sum_{j' \neq j} \tilde{w}_{jj'} \mathcal{L}_{ad}(D_{jj'}(G_j(G_0(\mathbf{x}^i))), \tilde{\mathbf{a}}_{j'}^i), \quad (2)$$

where $\mathcal{L}_{ad}$ is the loss function for the adversarial learning, and $\tilde{w}_{jj'}$ is the weight for the (j, j') pair, $j, j' \in [m+1]$. To re-enforce attribute learning during the adversarial learning, we optimize by

$$\min_{G_0} \sum_{i \in \Omega} \tilde{w}_{jj} \mathcal{L}_{ad}(D_{jj}(G_j(G_0(\mathbf{x}^i))), \tilde{\mathbf{a}}_j^i). \quad (3)$$

Finally, for eliminating all types of domain-difference, we optimize by

$$\min_{G_0, \{G_j\}} \sum_{i \in \Omega} \sum_{j' \neq j} \tilde{w}_{jj'} \mathcal{L}_{ad}(D_{jj'}(G_j(G_0(\mathbf{x}^i))), \tilde{\boldsymbol{z}}_{j'}^i), \quad (4)$$

where $\tilde{\boldsymbol{z}}_{j'}^i = \mathbf{1} - \tilde{\mathbf{y}}_{j'}^i$.

The activation function chosen for the last layer of the discriminative networks is a softmax function. We choose the cross-entropy loss as $\mathcal{L}_{at}$, and the mean square error as $\mathcal{L}_{ad}$ (referring to LSGAN [21]). Eq. (1), (2), (3) and (4) are alternatively optimized. For each mini-batch, Eq. (1) and (2) run one step, while Eq. (3) and (4) run five steps.

For inference by only stage 1, we stack P, G_1 and D_{11} to predict the class label y_i for each sample i. Although disentangling the class from all the types of domain-difference can be accomplished only by the first branch of the network, *i.e.*, the networks directly connected to G_1, we think that such a modeling strategy does not leverage sufficient supervised information to improve the representation ability of P. Later on in Section 4 we demonstrate that this results in a drastic decrease of accuracy by our ablation study.

We show that our optimization scheme can improve *Equality of Odds*, which is a fairness measure defined by Hardt *et al*. [11]. It means that a predictor $\hat{Y}$ and a domain variable Z are conditionally independent given the true label Y, *i.e.*, $P(\hat{Y} \mid Z, Y) = P(\hat{Y} \mid Y)$.

Theorem 1. *For the GCDR problem and the defined model, our optimization scheme defined by Eq.* (1) $\sim$ *Eq.* (4) *can improve the equality of odds defined by Hardt* et al. *[11].*

Proof. The proof is in the supplementary material. □

3.2. Additive Adversarial Learning

To further improve the disentangle performance, we propose an additive adversarial learning (AAL) mechanism taking advantage of the attribute-combinations that are not seen in the training data. The attribute-combinations are generated by a data-augmentation procedure. We expect the AAL mechanism to have the following property: when the

module "sees" the unseen ⟨class, domain⟩ combinations, biases are removed by minimizing negative side-effects.

First, we describe the data-augmentation procedure. For each ith generated data instance, the feature vector is a combination of $(m+1)$ feature vectors, $\tilde{\mathbf{f}}^i_1, \ldots, \tilde{\mathbf{f}}^i_{(m+1)}$, and the associated attribute vector is $[\tilde{a}^i_1, \ldots, \tilde{a}^i_{(m+1)}]$. For each attribute $j \in [m+1]$, $\tilde{\mathbf{f}}^i_j = \mathbf{f}^l_j$ and $\tilde{a}^i_j = a^l_j$, where $\mathbf{f}^l_j$ and a^l_j are the lth attribute feature vector and attribute value for attribute j, respectively, and $l \in \mathbb{Z}_+$ is a random index of training instances. For different attributes, the random indices can be different. For example, assuming $m = 1$, for two samples, $([\mathbf{f}^1_1, \mathbf{f}^1_2], [a^1_1, a^1_2])$ and $([\mathbf{f}^2_1, \mathbf{f}^2_2], [a^2_1, a^2_2])$, a generated data sample can be $([\mathbf{f}^1_1, \mathbf{f}^2_2], [a^1_1, a^2_2])$. The screening strategy of Lu *et al*. [18] is applied.

Next, we derive the AAL mechanism. The n_r generated data samples are separated into two collections: $\Omega_s = \{i \in [n_r]$: the attribute-value combination $[\tilde{a}^i_1, \ldots, \tilde{a}^i_{(m+1)}]$ has been seen in the training data$\}$, and $\Omega_u = \{i \in [n_r]$: the attribute-value combination $[\tilde{a}^i_1, \ldots, \tilde{a}^i_{(m+1)}]$ has not been seen in the training data$\}$. Based on these two collections, we illustrate our key idea of AAL by Fig. 3. Assume there are only two attributes: digit and background color, which are for the learning of two branches of the network, respectively. We assume that the disentanglement of stage 1 is already close to the optimum. Then for the seen attribute-value combinations, for each attribute j, we learn a transformation network T_j to predict $\tilde{a}_j$ only. We assume that this learning process let the networks fit the data of seen combinations, *e.g*., a digit five with red background can be precisely recognized as "5" for digit and "red" for the background. Then for an unseen combination, a digit five and green background, we let the network to output "5" for digit and "green" for the background. Under the assumption above, if the output color is not "green", we believe the error is from the red information of the first branch. Therefore we back-propagate the loss from the second output to the first branch to eliminate the background information within. Finally, the bias in the first branch can be removed.

As shown in the second part of stage 2 in Fig. 1, for each generated data sample, the feature vectors, $\tilde{\mathbf{f}}^i_1, \ldots, \tilde{\mathbf{f}}^i_{(m+1)}$, are transformed into *additive feature vectors* $\mathbf{s}_1, \ldots, \mathbf{s}_{(m+1)}$ by *additive space transformation networks* $T_1, \ldots, T_{(m+1)}$, respectively. The additive feature vectors $\mathbf{s}_1, \ldots, \mathbf{s}_{(m+1)}$ are added as a *summative feature vector* $\mathbf{u}$ which is sent to *recognition networks* $R_1, \ldots, R_{(m+1)}$. For attribute-value combinations seen in the training data, for each attribute $j \in [m+1]$, the loss from R_j is back propagated only to T_j, i.e., we optimize the following problem:

$$\min_{R_j, T_j} \sum_{i \in \Omega_s} w'_j \mathcal{L}_r \left(R_j \left(\sum_{j'=1}^{m} T_{j'}(\tilde{\mathbf{f}}^i_{j'}) \right), \tilde{a}^i_j \right), \tag{5}$$

where $\mathcal{L}_r$ is the recognition loss function, and w'_j is the weight for the attribute $j \in [m+1]$. On the other hand, for attribute-value combinations unseen in the training data, for each attribute $j \in [m+1]$, the loss from R_j is back propagated to $T_{\mathcal{S}_j}$, where $\mathcal{S}_j = \{j' \in [m+1] : j' \neq j\}$:

$$\min_{R_j, T_{\mathcal{S}_j}} \sum_{i \in \Omega_u} w'_j \mathcal{L}_r \left(R_j \left(\sum_{j'=1}^{m} T_{j'}(\tilde{\mathbf{f}}^i_{j'}) \right), \tilde{a}^i_j \right). \tag{6}$$

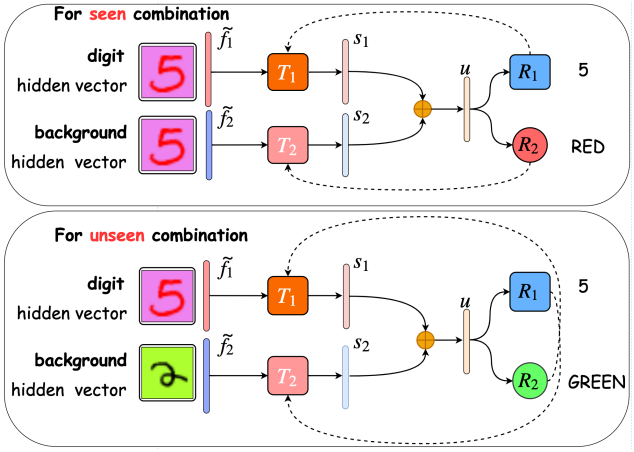

Figure 3. Key ideas of our AAL mechanism. Dotted lines represent the directions of backpropagation. Best viewed in color.

The additive learning mechanism holds two good properties: (1) the discriminative information in each dimension can be expressed in an additive form which is decomposable, and (2) each dimension of $\mathbf{s}_j$ for all $j \in [m+1]$ has the same meaning, which allows us to incorporate sparse penalties to let each group of dimensions of the additive feature vectors correspond to a single attribute.

Same as in stage 1, we choose softmax activation functions and cross-entropy loss for the last layers. For inference, we stack P, G_1, T_1 and R_1 to predict the class label.

3.3. Discussion on Causal Extension

We further consider alleviating a dilemma of disentangling when some attributes are correlated. For the most extreme case, if two attributes are identical, it is not possible that we cannot recognize one attribute based on a feature vector but can recognize another. Therefore, intuitively, we should not disentangle correlated attributes. However, "correlation" is a broad, imprecise concept. If we do not disentangle for all the correlated attributes, we may encounter insufficient disentanglement. We consider a specific type of correlation: causal relationships. We theoretically demonstrate in Theorem 2 that for any attribute j, if another attribute j' causes it, then learning a feature vector $\mathbf{f}_j$ to recognize attribute j while disentangling it from attribute j' may hurt the recognition for attribute j. This is because if $\mathbf{f}_j$ is independent with attribute j', since attribute j' causes attribute j, the correlation between $\mathbf{f}_j$ and attribute j is lim-

ited. Therefore, if the prior information of causal relationships between attributes is given, we should cease some disentangling processes to avoid the learning dilemma.

Specifically, for stage 1, we can use a prior matrix $\mathbf{\Lambda} \in \{0,1\}^{(m+1)\times(m+1)}$ to handle causalities between attributes. For all $j, j' \in [m+1]$ and $j' \neq j$, we multiply the weight $\tilde{w}_{jj'}$ in Eq. (2) and Eq. (4) by $\mathbf{\Lambda}_{jj'}$. Based on Theorem 2, we set $\mathbf{\Lambda}_{jj'} = 0$ if attribute j' causes attribute j, and set $\mathbf{\Lambda}_{jj'} = 1$ otherwise. For stage 2, for the indices collection $\mathcal{S}_j$ in Eq. (6), we can delete the indices of attributes that are caused by attribute j, for each $j \in [m+1]$.

Theorem 2. *For all attribute $j \in [m+1]$, for an arbitrary feature vector $\mathbf{f}_j$, denote the true label of $\mathbf{f}_j$ as $a_j \in \mathbb{Z}_+$. Then for attributes $j, j' \in [m+1]$ and $j \neq j'$, if attribute j' causes attribute j, we cannot reach both the learning goals $P(a_j \mid \mathbf{f}_j) = 1$ and $P(a_{j'} \mid \mathbf{f}_j) = P(a_{j'})$ perfectly. However, if attribute j causes attribute j', it is possible to reach both the learning goals perfectly. The proof can be found in the supplementary material.*

Proof. The proof is in the supplementary material. □

4. Experiments

In this section, we evaluate the proposed method. Both synthetic and real-world data sets are used for evaluations. Our implementation uses Keras with Tensorflow [1] backends, which can be found at https://github.com/langlrsw/AAL-unbiased-authentication.

We consider a recognition task in the presence of several types of domain-difference. For each type of domain-difference, different domains do not share classes in the training set, and the training and testing sets do not share combinations of ⟨class, domain⟩. Comprehensive evaluations are conducted on three data sets: (1) the C-MNIST data set [18] with 10 classes and $m = 2$ types of domain-difference, (2) the re-organized CelebA data set [16] with 211 classes and $m = 1$ type of domain-difference, and (3) our developed authentication data set based on mobile sensors with 29 classes and $m = 1$ type of domain-difference. For each data set, the re-organization will be described in detail in the corresponding section, and 10% data of the testing set were randomly selected for validation. The evaluations follow the GCDR setting defined in Problem 1.

Methods for Comparison As discussed in Section 2, there are three thrusts of methods to leverage existing transfer learning methods to handle domain-difference. For the first thrust that eliminates the marginal distribution differences, we chose RevGrad [9], which also serves as the solution of the FML methods of Beutel *et al*. [3]. For the second thrust that generates data with unseen ⟨class, domain⟩ combinations, we chose ELEGANT [35] which only uses domain labels and ML-VAE [5] which only uses class labels. For the third thrust that uses hybrid solutions, we chose ABS-Net [18] which is the base method of ours without an adversarial mechanism, and CDRD [15] and SE-GZSL [30], which can be treated as advanced instantiated algorithms under the FML framework of Madras *et al*. [20]. Finally, we compare the direct learning strategy that stacks P, G_1, and D_{11} as the whole network.

Evaluation Metrics We investigate prediction performances for both multi-label and multi-class types of recognition. Therefore, for the multi-label type, we use average AUC (aAUC) which is defined as the mean value of the area under the ROC curve for each class, the average false acceptance rate (aFAR), and the average false rejection rate (aFRR). Because the number of negative samples is far greater than that of positive samples for each class, we report aAUC and $(\text{aFAR} + \text{aFRR})/2$. For the multi-class type, we report top-1 accuracy (ACC@1).

Implementation Details For all experiments, $G_1, \ldots, G_m$ and $\{T_j\}$ are built by a single hidden layer with hyperbolic tangent as the activation function, respectively. Hidden units are flattened before being fed into attribute-feature learning networks. $\{D_{jj'}\}$ and $\{R_j\}$ are built by generalized linear layers. A Convolutional Neural Network (CNN) with two convolutional layers is used as the input feature transformation network P for image data sets. A fully-connected neural network with one hidden layer P is used for vector based data sets. The weights $\{w_j\}$, $\{\tilde{w}_{jj'}\}$, and $\{w'_j\}$ are set as follows. We set $w_1 = w'_1 = 1$ and $\tilde{w}_{j1} = 1$ for each $j \in [m+1]$. Other weights are equally set to 0.1.

4.1. Handwritten Digital Experiments

We re-construct the C-MNIST data set originally built by Lu *et al*. [18] for performance evaluation. For original gray images of MNIST, 10 different colors are added as background colors (b-colors) and other 10 different colors are added as foreground colors (f-colors), which results in a new C-MNIST that consists of 70k colored RGB digital images with resolution of 28×28 (60k for training and 10k for testing) from 1k possible combinations (10 digits $\times$ 10 b-colors $\times$ 10 f-colors). Examples from C-MNIST are shown in Fig. 6 of Lu *et al*. [18].

In this paper, we set the digit recognition as the primary learning task. The background and foreground colors can be treated as two types of domain-difference. It is evident that the background colors are independent with the digits. However, they have server influence on the prediction accuracy because they occupy most of the image area. Therefore, we use the background color as the domain-difference that groups the digits. As shown in Fig. 2, in the training data, digits $0 \sim 4$ are associated with a green b-color, while $5 \sim 9$ are associated with a pink b-color, other data are dropped. The data with $\langle 0 \sim 4$, pink b-color$\rangle$ and $\langle 5 \sim 9$, green b-color$\rangle$ combinations are for testing. The foreground

Methods	aAUC	$(\text{aFAR} + \text{aFRR})/2$	ACC@1
Direct	78.35	26.70	20.96
RevGrad [9, 3]	80.71	24.45	21.68
CDRD [15, 20]	84.83	35.79	33.49
SE-GZSL [30, 20]	**99.79**	**2.72**	**94.83**
ELEGANT [35]	79.94	24.61	10.68
ML-VAE [5]	77.26	28.06	18.73
ABS-Net [18]	77.69	27.41	15.92
Ours	98.42	6.14	84.27

Table 2. Performances (%) comparison on the C-MNIST data set. "Direct" means stacking P, G_1, and D_{11} as the whole network.

attribute is also used to disentangle, but we allowed it share digits in the training data. We have 5970 training instances and 1003 testing instances in total.

Table 2 summarizes the performance comparisons on C-MNIST. The results send a clear message that our method significantly outperforms the direct learning method, which shows the effectiveness of our method. Furthermore, our method outperforms other baseline methods significantly, except for the SE-GZSL method. We conjecture that C-MNIST is easy for SE-GZSL because the domain-difference is the background color which is simple and stable. However, in real applications, these properties barely hold, which on the succeeding two real-world data sets we will show that its performance drops.

We extended the experiments for other background colors and the foreground colors. Please find more details in the supplementary material.

4.2. Face Recognition

We use aligned, and cropped version of the CelebA data set [16] and scale all images to 64×64. We chose the *Eyeglasses* attribute as the domain-difference. We select individuals who have at least 20 images and $\#(Eyeglasses = 0)/\#(Eyeglasses = 1) \in [3/7, 7/3]$, resulting in 211 individuals. Half of the individuals wear glasses during training and without glasses during testing. The other half wear no glasses during training and wear glasses during testing. Table 3 shows the comparisons conducted on CelebA. Our method significantly outperforms other methods in aAUC and $(\text{aFAR} + \text{aFRR})/2$—the multi-label type of metrics. The SE-GZSL method underperforms, which suggests its insufficient inconsistency. The result demonstrates that complex and variable domain-difference types on real-world data sets are difficult for SE-GZSL to learn. The CDRD method underperforms in aAUC and $(\text{aFAR} + \text{aFRR})/2$, but outperforms in ACC@1, because the positive samples of the majority of individuals have lower prediction scores, but the positive samples of more individuals have high prediction scores, which shows less satisfactory authentication performance for the majority of individuals.

Methods	aAUC	$(\text{aFAR} + \text{aFRR})/2$	ACC@1
Direct	78.54	42.10	11.07
RevGrad [9, 3]	80.12	31.18	10.96
CDRD [15, 20]	80.20	39.90	**16.47**
SE-GZSL [30, 20]	84.96	26.62	12.76
ELEGANT [35]	75.88	32.02	10.05
ML-VAE [5]	75.29	36.07	7.97
ABS-Net [18]	75.80	34.90	8.09
Ours	**87.07**	**22.19**	14.99

Table 3. Performances (%) comparison on the CelebA data set.

	No. 1-6	No. 7-12	No. 13-15	No. 16-29
IOS	Train	Test	×	Train
Android	Test	Train	Train	×

Table 4. The authentication problem on mobile devices. The numbers in the first row indicate groups of subjects. "×" means there are no data for this condition.

Methods	aAUC	$(\text{aFAR} + \text{aFRR})/2$	ACC@1
Direct	76.90	28.21	3.56
RevGrad [9, 3]	75.88	32.38	0.38
CDRD [15, 20]	89.17	20.26	46.05
SE-GZSL [30, 20]	78.83	26.12	20.54
ML-VAE [5]	77.16	27.18	4.68
ABS-Net [18]	76.58	28.09	5.13
Ours	**93.40**	**13.59**	**46.37**

Table 5. Performances (%) comparison on the Mobile data set.

4.3. Authentication on Mobile Devices

We also build a data set containing sensor information of smart-phones from 29 subjects, which records two-second time-series data from multiple sensors, such as accelerometer, gyroscope, gravimeter, *etc*. Statistical features from both time and spectrum domains are extracted with 191 dimensions for all the 5144 data instances. The OS types (IOS/Android) are considered to be the domain-difference. We select 12 subjects that have used both types of the phone system and then construct a biased learning task as shown in Table 4. The ELEGANT method is not suitable for non-image data and therefore is removed. The results are reported in Table 5, in which our method significantly outperforms other methods in aAUC and $(\text{aFAR} + \text{aFRR})/2$. The SE-GZSL method also underperforms for this difficult type of domain-difference as well. The CDRD method still underperforms in the multi-label type of metrics, especially in $(\text{aFAR} + \text{aFRR})/2$.

4.4. Ablative Study

We conduct a series of ablation experiments on the C-MNIST data set in the aforementioned setting to demon-

Methods	aAUC	(aFAR + aFRR)/2	ACC@1
Stage 1+2	**98.42**	**6.14**	**84.27**
Stage 1	95.91	10.20	70.56
Single-Branch	78.87	26.70	24.47
Shared-Ds	87.27	18.72	46.75
No-Adv-Stage-1	92.47	14.81	46.97
No-Adv-At-All	77.69	27.41	15.92
Direct	78.35	26.70	20.96

Table 6. Results of the ablation study.

strate how the OVRDL (stage 1) and AAL (stage 2) mechanisms contribute to the performance. Specifically, we compare the performance of the following four model variants.

Single-Branch. Only the first branch of networks in stage 1 is left. Stage 2 is therefore removed because it works with a multi-branch stage 1.

Shared-Ds. $D_{1j} = \ldots = D_{mj}$, for all $j \in [m+1]$.

No-Adv-Stage-1. The networks $\{D_{jj'}\}_{j \neq j'}$ in stage 1 are removed.

No-Adv-At-All. Based on No-Adv-Stage-1, in stage 2, losses are back-propagated to all the networks as normal. It is the ABS-Net [18] method.

The results are presented in Table 6. It is notable that Single-Branch's performances drastically decrease comparing with other methods containing adversarial learning. The performances of Single-Branch are similar to those of the related approaches listed in Table 2. Such phenomenon suggests that building a single-branch model to handle domain-differences is not sufficient and that it is worthwhile to build a multi-branch model to learn all the attributes to improve the representation ability of P. Such a multi-branch structure is the main difference between our method and the related approaches, which we believe is of the main structural contributions of our framework. Besides, Shared-Ds' performances also decrease considerably. It is worth mentioning that, for Shared-Ds, the performances of both stages are nearly the same. These phenomena demonstrate that restricting the attribute vectors in the same space harms their learning of independent representations. Comparing No-Adv-Stage-1 with Shared-Ds, no adversarial learning in stage 1 is better than "shared" adversarial learning, which also demonstrates the importance of independent representations. These phenomena demonstrate the high effectiveness of our proposed OVRDL mechanism.

On the other hand, compared stage 1+2 with stage 2, the improvement gained from stage 2 is significant. For No-Adv-Stage-1, it is worth mentioning that it can only achieve aAUC of 75% without stage 2. Moreover, No-Adv-At-All significantly underperforms compared with No-Adv-Stage-1 by only changing the back-propagation mechanism of stage 2. These phenomena demonstrate the high effectiveness of our proposed AAL mechanism in stage 2.

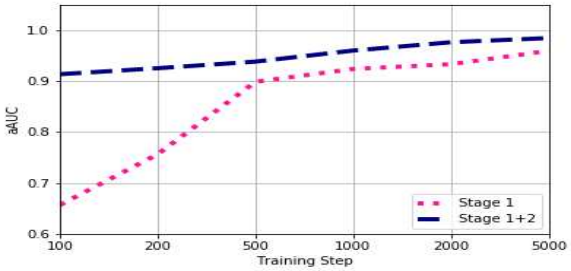

Figure 4. Improvements of the AAL mechanism in stage 2 during different training phases of stage 1.

We further investigate the effectiveness of the AAL mechanism during different training phases of stage 1. Fig. 4 shows that the AAL mechanism can contribute 25% absolute performances at the beginning of the training. At the middle and later phases of training, the improvements are limited, because AAL aims to eliminate biased factors in the features further, but such factors are nearly cleansed to the optimum by stage 1.

5. Conclusion

In this paper, we investigate data biases and a generalized cross-domain recognition problem in the field of authentication where domains do not share classes. We recognize the class for unseen ⟨class, domain⟩ combinations of data. We propose a two-stage disentangle learning method to tackle the problem. The stage 1 builds a one-versus-rest disentangle learning mechanism to disentangle the class and each type of domain-difference. The stage 2 conducts a data augmentation and uses a proposed additive adversarial learning to improve the disentanglement of stage 1 further. We also discuss how to avoid the dilemma due to disentangling causally related types of domain-difference. The experiments demonstrate that our method significantly outperforms existing state-of-the-art methods. We also conduct an ablation study to demonstrate the effectiveness of the critical components of our method. Some interesting future directions of research include developing transfer learning algorithms flexible to train to increase the number of types of domain-difference.

Acknowledgment

We would like to thank the TuringShield team of Tencent for supporting our research, Lu *et al*. [18] for sharing their codes of ABS-Net and C-MNIST data set, and Dr. Bing Bai from the Cloud and Smart Industries Group at Tencent for his insightful suggestions.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 6

[2] A. Bedagkar-Gala and S. K. Shah. A survey of approaches and trends in person re-identification. *Image and Vision Computing*, 32(4):270–286, 2014. 1

[3] A. Beutel, J. Chen, Z. Zhao, and E. H. Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017. 2, 6, 7

[4] D. Bhattacharyya, R. Ranjan, F. Alisherov, M. Choi, et al. Biometric authentication: A review. *International Journal of u-and e-Service, Science and Technology*, 2(3):13–28, 2009. 1

[5] D. Bouchacourt, R. Tomioka, and S. Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. *arXiv preprint arXiv:1705.08841*, 2017. 3, 6, 7

[6] J. Cao, O. Katzir, P. Jiang, D. Lischinski, D. Cohen-Or, C. Tu, and Y. Li. Dida: Disentangled synthesis for domain adaptation. *arXiv preprint arXiv:1805.08019*, 2018. 3

[7] A. Chouldechova and A. Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018. 2

[8] G. Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017. 1

[9] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014. 2, 6, 7

[10] N. Goel, M. Yaghini, and B. Faltings. Non-discriminatory machine learning through convex fairness criteria. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA*, 2018. 2

[11] M. Hardt, E. Price, N. Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016. 4

[12] C. Heinze-Deml and N. Meinshausen. Grouping-by-id: Guarding against adversarial domain shifts. 2018. 3

[13] A. Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. 4

[14] M. Kocaoglu, C. Snyder, A. G. Dimakis, and S. Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017. 3

[15] Y.-C. Liu, Y.-Y. Yeh, T.-C. Fu, S.-D. Wang, W.-C. Chiu, and Y.-C. F. Wang. Detach and adapt: Learning cross-domain disentangled deep representation. *arXiv preprint arXiv:1705.01314*, 2017. 3, 6, 7

[16] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 6, 7

[17] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2

[18] J. Lu, J. Li, Z. Yan, F. Mei, and C. Zhang. Attribute-based synthetic network (abs-net): Learning more from pseudo feature representations. *Pattern Recognition*, 80:129–142, 2018. 3, 5, 6, 7, 8

[19] Y. Luo, Y. Wen, L. Duan, and D. Tao. Transfer metric learning: Algorithms, applications and outlooks. *arXiv preprint arXiv:1810.03944*, 2018. 1

[20] D. Madras, E. Creager, T. Pitassi, and R. Zemel. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*, 2018. 3, 6, 7

[21] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE, 2017. 4

[22] A. Mir, S. Rubab, and Z. Jhat. Biometrics verification: a literature survey. *International Journal of Computing and ICT Research*, 5(2):67–80, 2011. 1

[23] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011. 2

[24] S. J. Pan, Q. Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 1

[25] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015. 1

[26] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney. Fairness gan. *arXiv preprint arXiv:1805.09910*, 2018. 3

[27] W. Shen and R. Liu. Learning residual images for face attribute manipulation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1225–1233. IEEE, 2017. 3

[28] S. Sun, H. Shi, and Y. Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92, 2015. 1

[29] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 2

[30] V. K. Verma, G. Arora, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6, 7

[31] C. Wadsworth, F. Vera, and C. Piech. Achieving fairness through adversarial learning: an application to recidivism prediction. *arXiv preprint arXiv:1807.00199*, 2018. 2

[32] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 1

[33] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016. 1, 2

[34] T. Xiao, J. Hong, and J. Ma. Dna-gan: Learning disentangled representations from multi-attribute images. *arXiv preprint arXiv:1711.05415*, 2017. 3

[35] T. Xiao, J. Hong, and J. Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes.

In *The European Conference on Computer Vision (ECCV)*, September 2018. 3, 6, 7

[36] D. Xu, S. Yuan, L. Zhang, and X. Wu. Fairgan: Fairness-aware generative adversarial networks. *arXiv preprint arXiv:1805.11202*, 2018. 3

[37] N. Yager and A. Amin. Fingerprint verification based on minutiae features: a review. *Pattern Analysis and Applications*, 7(1):94–113, 2004. 1

[38] H. Yu, M. Hu, and S. Chen. Multi-target unsupervised domain adaptation without exactly shared categories. *arXiv preprint arXiv:1809.00852*, 2018. 3

[39] B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. *arXiv preprint arXiv:1801.07593*, 2018. 2

[40] G. Zhang, M. Kan, S. Shan, and X. Chen. Generative adversarial network with spatial attention for face attribute editing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 417–432, 2018. 3

[41] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003. 1

[42] Y. Zhao. Network inference from temporal-dependent grouped observations. *arXiv preprint arXiv:1808.08478*, 2018. 3

[43] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016. 1