SpeakerGAN: Recognizing Speakers in New Languages with Generative Adversarial Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Verifying a person's identity based on their voice is a challenging, real-world problem in biometric security. A crucial requirement of such speaker verification systems is to be domain robust. Performance should not degrade even if speakers are talking in languages not seen during training. To this end, we present a flexible and interpretable framework for learning domain invariant speaker embeddings using Generative Adversarial Networks. We combine adversarial training with an angular margin loss function, which encourages the speaker embedding model to be discriminative by directly optimizing for cosine similarity between classes. We are able to beat a strong baseline system using a cosine distance classifier and a simple score-averaging strategy. Our results also show that models with adversarial adaptation perform significantly better than unadapted models. In an attempt to better understand this behavior, we quantitatively measure the degree of invariance induced by our proposed methods using Maximum Mean Discrepancy and Fréchet distances. Our analysis shows that our proposed adversarial speaker embedding models significantly reduce the distance between source and target data distributions, while performing similarly on the former and better on the latter.

1 Introduction

Text-Independent Speaker Verification remains a challenging problem in the domain of biometric security. Armed with the machinery of deep learning, verification systems can now be deployed in the wild, and are still capable of delivering robust performance. In the verification community, situations wherein the test data is significantly different from the data available during system training are referred to as - In the Wild. For instance, the NIST-SRE 2016 evaluation data contains Cantonese and Tagalog speakers (in-domain, target data), while most of the speakers in our training set are talking in English (out-of-domain, source data). This distribution shift or mismatch between training and test data is an obstacle in several areas of pattern recognition and machine learning [1], and leads to a degradation in system performance. The development biometric verification system that perform reliably in such conditions is critical for this technology be used safely and securely on a day-to-day basis.

Deep neural networks (DNN) have revolutionized several areas of speech processing, and as such, are ideal candidates for learning discriminative speaker representations or embeddings [20, 10, 23, 3]. Indeed, neural speaker embeddings have surpassed the performance of i-vectors [20, 5], especially on real world, in the wild data [17, 14]. Arguably the most popular approach for learning speaker embeddings is to optimize the parameters of a DNN by minimizing the cross-entropy loss over speakers in the training data. Cross-entropy is natural choice for identifying speakers, however it does not directly address the verification task. As a consequence of not being optimized 'end-to-end', the performance of cross-entropy speaker embeddings (X-vectors) is heavily dependent on a

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

powerful classifier to perform verification. This dependence on a classifier motivates the research and development of end-to-end systems. We also believe that such systems can also benefit in downstream tasks that make use of speaker embeddings, such as speech recognition and synthesis. Speaker verification is a challenging problem, and modern verification datasets like NIST-SRE 2016, add to this challenge by introducing a mismatch between the distributions of the training and test data. This phenomena is referred to as domain or covariate shift. In the case of NIST-SRE 2016, the test data consists of Cantonese and Tagalog speakers, whereas the vast majority of training speakers are talking in English. NIST also provide a small amount of unlabelled, *in-domain, target* data, that can be used to compensate for the domain shift. Most the domain adaptation techniques that have been proposed for speaker verification have been proposed on top of i-vectors or x-vectors.

In this work we present a framework for learning domain invariant speaker embeddings using Generative Adversarial Networks (GAN). We drawn inspiration from research in computer vision, where GAN based unsupervised domain adaptation methods have been extremely successful [6, 21, 18, 19], and adapt these ideas for feature learning in a verification setting. The basic idea is cast domain adaptation/invariance as an adversarial game - generate features or embeddings such that a discriminator cannot tell if they come from the source or target domain. Unlike traditional GANs that work in high-dimensional spaces (e.g. natural images,speech), domain adaptation GANs operate in low-dimensional embedding space. We extend our recent work [2, 4] and propose a novel objective for updating the generator network. We find that optimizing GAN models with this objective proves to be unstable, and propose to stabilize it by augmenting the discriminator with an auxiliary loss function. This strategy also helped stabilize training for the conventional generator objective but was not strictly needed.
Additionally, we analyze the transformed source and target data distributions in order to gain further insight regarding the performance of our method. We measure distances between these distributions using Maximum Mean Discrepancy and Fréchet distances. From our analysis we see that a good performance in terms of distributional distance corresponds to good verification performance. Our speaker verification experiments show that the proposed adversarial speaker embedding framework delivers robust performance, significantly outperforming a strong i-vector baseline. Furthermore, by averaging the scores of our different GAN models, we are able to achieve state-of-the-art results.

2 Models

2.1 Feature Extractor (Generator)

The first step for learning discriminative speaker embeddings is to learn a mapping $F(X_s) \longrightarrow \mathbf{f}$, $\mathbf{f} \in R^D$ from a sequence of speech frames from speaker s to a D-dimensional feature vector **f**. $F(X)$ can be implemented using a variety of neural network architectures. We design our feature extractor using a residual network structure. We choose to model speech using 1-dimensional convolutional filters, owing to the fact that speech is translation invariant along the time-axis only. Following the residual blocks we use a combination of self-attention and dense layers in order to represent input audio of arbitrary size by a fixed-size vector, **f**. Unlike traditional approaches, our proposed feature extractor is updated with an adversarial loss in addition to the standard task loss.

2.2 Self-Attentive Speaker Statistics

Self-Attention models are an active area of research in the speaker verification community. Intuitively, such models allow the network to focus on fragments of speech that are more speaker discriminative. The attention layers computes a scalar weight corresponding to each time-step t:

$$e_t = \mathbb{v}^T f(\mathbf{W}h_t + \mathbf{b}) + k \tag{1}$$

These weights are then normalized, $\alpha_t = softmax(e_t)$, to give them a probabilistic interpretation. We use the attention model proposed in [25], which extends attention to the mean as well as standard deviation:

$$\hat{\mu} = \sum_t^T \alpha_t \mathbf{h}_t \tag{2}$$

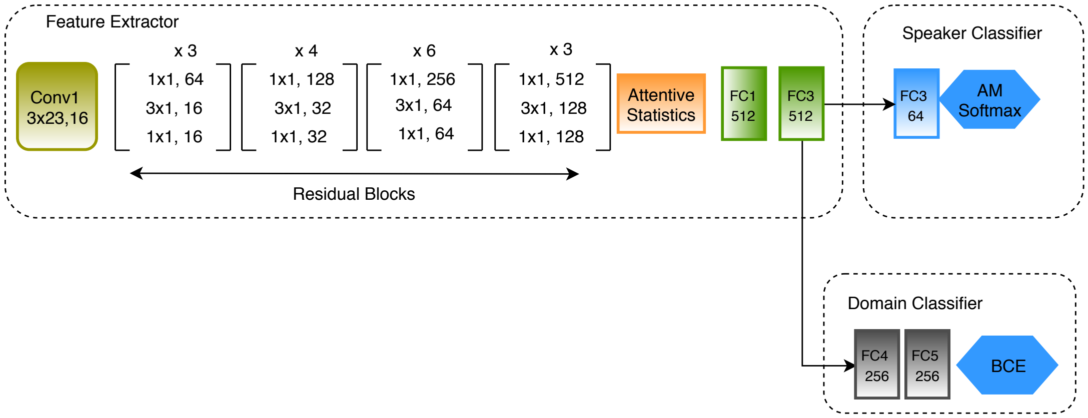

Figure 1: Domain Adversarial Neural Speaker Embedding Model.

$$\hat{\sigma} = \sum_{t}^{T} \alpha_t \mathbf{h}_t \odot \mathbf{h}_t - \hat{\mu} \odot \hat{\mu} \tag{3}$$

In this work we apply the use of self attention to convolutional feature maps, as indicated in Fig. 1. The last residual block outputs a tensor of size $n_B \times n_F \times T$, where n_B is the batch size, n_F is the number of filters and T is time. The input to the attention layer, h_t, is a n_F dimensional vector.

By using a self-attention model, we also equip our network with a more robust framework for processing inputs of arbitrary size than simple global averaging. This allows us simply forward propagate a recording through the network in order to extract speaker embeddings.

2.3 Classifier

The classifier block, $C(\mathbf{f}, \theta_y)$, is arguably the key component of the model, as it is responsible for learning speaker discriminative features.Recently, angular margin loss functions have been proposed as an alternative to contrastive loss functions for verification tasks [11, 24]. The Additive Margin softmax (AM-softmax) loss function is one such algorithm with an intuitive interpretation. The loss computes similarity between classes using cosine, and forces the similarity of the correct class to be greater than that of incorrect classes by a margin m.

$$\begin{aligned} L_{AMS} &= -\frac{1}{n}\sum_{i=1}^{n} \log \frac{e^{s.(cos\theta_{y_i} - m)}}{e^{s.(cos\theta_{y_i} - m)} + \sum_{j \neq y_i} e^{s.(cos\theta_j)}} \\ &= -\frac{1}{n}\sum_{i=1}^{n} \log \frac{e^{s.(W^T \boldsymbol{f_i} - m)}}{e^{s.(W^T \boldsymbol{f_i} - m)} + \sum_{j \neq y_i} e^{s.(W^T \boldsymbol{f_j})}} \end{aligned} \tag{4}$$

Where W^T and f_i are the normalized weight vector and speaker embedding respectively. The AM-softmax loss also adds a scale parameter s, which helps the model converge faster. We select $m = 0.6$ and $s = 30$ for all our experiments.

2.4 Domain Discriminator

The domain discriminator $D(.)$ is tasked with determining if embeddings come from the source or target domains, and is arguably the most important component of the model. In order to learn domain invariant features, we engage the domain discriminator in an adversarial game with the feaure extractor $E(.)$. The domain discriminator consists of two fully connected layers followed by the output layer.

3 Domain Adversarial Speaker Embeddings

A crucial requirement for learning speaker embeddings that are domain invariant is to find a balance between the task loss and the adversarial loss. The objective to learn a feature space wherein embeddings are speaker discriminative irrespective of the domain. Key to achieving this is the domain discriminator D, which is trained using the Binary Cross-Entropy loss (BCE).

$$\mathcal{L}_{adv_D}(\mathbf{X}_s, \mathbf{X}_t, E) = -E_{x_s \sim X_s}[\log(D(E(x_s))] - E_{x_t \sim X_t}[\log(1 - D(E(x_t))] \tag{5}$$

Where $\mathbf{X}_s$, $\mathbf{X}_t$ represent source and target data respectively. $E(.)$ is the feature extractor/generator. The adversarial game between $D(.)$ and $E(.)$ is given by:

$$\begin{aligned} &\min_D \ \mathcal{L}_{adv_D}(\mathbf{X}_s, \mathbf{X}_t, E) \\ &\min_E \ \mathcal{L}_{adv_E}(\mathbf{X}_s, \mathbf{X}_t, D) \end{aligned} \tag{6}$$

Equation (6) represents the most general form of the GAN game, and can be used to represent different adversarial frameworks depending on the choice of $\mathcal{L}_{adv E}$.

Gradient Reversal: We obtain the gradient reversal framework by setting $\mathcal{L}_{adv E} = -\mathcal{L}_{adv D}$. Gradient reversal optimizes the true minmax objective of the adversarial game [6]. However, this objective can become problematic, since the discriminator converges early during training and leads to vanishing gradients. We refer to the model trained with gradient reversal as Domain Adversarial Neural Speaker Embeddings (DANSE).

GAN: Rather than directly using the minimax loss, the standard way to train the generator is using the inverted label loss. The generator objective is given by:

$$\mathcal{L}_{adv_E}(\mathbf{X}_s, \mathbf{X}_t, D) = -\mathbb{E}_{x_s \sim X_s}[\log(D(E(x_t))] \tag{7}$$

This splits the optimization into two independent objectives, one for the generator and one for the discriminator. This loss has the same fixed-point properties as the minimax loss while providing stronger gradients to target mappings [21].

3.1 Updating the Generator with Source Embeddings

In a typical GAN setting, the generator is trained only using fake data (with inverted labels). This structure is also maintained in several adversarial domain adaptation algorithms. However, in the context of this work we believe that updating the generator using *both* source and target data can be beneficial. In this case, the generator loss simply inverts the discriminator loss of eq. (1):

$$\begin{aligned} &\mathcal{L}_{adv_E}(\mathbf{X}_s, \mathbf{X}_t, D) = \\ &- E_{x_s \sim X_s}[\log(D(E(x_t))] \\ &- E_{x_t \sim X_t}[\log(1 - D(E(x_s))] \end{aligned} \tag{8}$$

When using the proposed objective for training the generator, we are optimizing the true minimax loss like in the gradient reversal approach. Unfortunately, we found that optimizing this loss becomes unstable early during training. We found a simple approach to stabilize training for this model was to augment the discriminator with an auxiliary loss function.

3.2 Auxiliary Classifier GAN

The Auxiliary Classifier GAN (AuxGAN) model augments the standard GAN framework with an auxiliary loss to perform conditional image generation [16]. This approach aims to predict side information (such as class labels), as opposed to feed the same information to the generator and discriminator. In the context of this work, our goal is to use the prediction loss for regularization and representation learning.

$$\min_{D} \mathcal{L}_{adv_D}(\mathbf{X}_s, \mathbf{X}_t, E) + \mathcal{L}_{Aux}(\mathbf{X}_s, Y_s)$$
$$\min_{E} \mathcal{L}_{adv_E}(\mathbf{X}_s, \mathbf{X}_t, D) + \mathcal{L}_{Aux}(\mathbf{X}_s, Y_s) \tag{9}$$

Eq. (9) is a modified version of the AuxGAN objective. In particular, the original formulation also uses the auxiliary loss to train the generator as well (with fake data being assigned its own unique label). We found that the auxiliary loss was crucial for stabilizing $\mathcal{L}_{adv_E}(\mathbf{X}_s, \mathbf{X}_t, D)$ when using the formulation in eq. (8). In our experiments we found that the AuxGAN setup stabilizes model training even when we use eq. (7) as the generator objective, and leads to slightly better verification performance. In this setting only the discriminator is trained with the auxillary loss.

3.3 GAN Variants

Since their introduction, GANs have been one of the most researched topics in the deep learning community. Several variations of the original formulation have been proposed, each with different generative characteristics and stability issues. In this work we explore three GAN variants in addition to the standard GAN - Least-Squares GAN [13], Auxiliary Classifier GAN and Relativistic GAN [9]. We use the standard average GAN variant of the Relativistic GAN model. These models differ in the structure of the discriminator network. We show that each variant transforms the feature space in different way, will all the model showing mostly similar performance. Additionally we see that by fusing the performance of all GAN variants together through score averaging we achieve the best overall performance.

4 Experimental Setup

Training Data (Source): We used audio from previous NIST-SRE evaluations (2004-2010) and Switchboard Cellular audio for training the proposed DANSE model as well as the x-vector and i-vector baseline systems. We also augment our data with noise and reverberation, as in [20]. We add 128k noisy copies to the clean speech, ending up with $\tilde{2}$20k recordings in our training set. For DANSE model training we filter out speakers with less than 5 recordings, ending up with approximately 6000 speakers, whereas the x-vector and i-vector systems were trained using the Kaldi recipe. We note that the vast majority of our training data consists of English speakers, and is recorded over telephone/cellular channels.

Model: The *Embedding function/Generator*, $E(.)$, consists of a 3×23 input convolutional layer, 4 residual blocks [3,4,6,3], an attentive statistics layer and two fully connected layers (512,512). The *classifier*, $C(.)$, module consists of a fully connected layer (64) and the AM-softmax output layer. The former is the final domain invariant speaker embedding extracted during evaluation. Finally, the *domain discriminator* module consists of two fully connected layers (256,256) and a binary cross-entropy output layer. Exponential Linear Units (ELU) are used as non-linear activations for all layers of the network. Batch Normalization is used on all layers expect the attentive statistics layer.

Optimization: We start by pre-training the Embedding function using standard cross-entropy training. Pre-training is carried out using the RMSprop optimizer with a learning rate (lr) of 0.001. For training GAN based speaker embedding models we use different optimizers for training the three networks (Embedding function,Classifier, Discriminator). The classifier is optimized using RMSprop with lr=0.003, while the domain classifier and feature extractor are trained using SGD with lr=0.001. We were able to train all our GAN models using the same set of hyper-parameters. We used performance on held out validation set to determine when to stop training.

Data Sampling: We use an extremely simple approach for sampling data during training. We sample random chunks of audio (3-8 seconds) from each recording in the training set. We sample each recording 10 times to define an epoch. For each mini-batch of source data, we randomly sample (with repetition) a mini-batch from the unlabelled adaptation data for GAN training. The training set contains recordings from 6000 speakers (we filter out speakers with less than 5 recordings) and a total of 217,620 recordings. The adaptation data contatins 2272 unleabelled recordings.

Speaker Verification: At test time we discard the domain discriminator branch of the model, as it is not needed for extracting embeddings. Extraction is done by performing a forward pass on the full recording, and using the 64-dimensional $FC3$ layer as our speaker embeddings. Verification trials are scored using cosine distance. Verification performance is reported in terms of Equal Error Rate (EER).

5 Results

NIST-SRE 2016: Unlike previous years, The 2016 edition of the NIST-SRE introduced a challenging new dataset containing Cantonese and Tagalog speakers. We use the Kaldi recipes for our baseline i-vector and x-vector systems. We note that the x-vector baseline may be considered as state-of-the-art performance on this dataset.

Adaptation Data (Target): 2722 unlabelled, target data recordings are provided to adapt verification systems.

Table 1: Performance of Baseline Systems (EER).

Model	*Classifier*	*Cantonese*	*Tagalog*	*Pooled*
i-vector	PLDA	9.51	17.61	13.65
x-vector	COSINE	36.44	41.07	38.69
x-vector	**LDA/PLDA**	**7.03**	**15.41**	**11.15**
x-vector	PLDA	18.46	7.99	12.21

Table 2: Performance of Different GAN systems in terms of EER(%). **GradRev:** Gradient Reversal **SGAN:** standard, **AuxGAN:** auxiliary classifier, **LSGAN:** least squares, **RelGAN:** reletavistic, **FuseGAN:** score averaging.

Model	*Classifier*	*Cantonese*	*Tagalog*	*Pooled*
GradRev	COSINE	8.84	18.21	13.36
SGAN	COSINE	8.32	17.51	12.65
AuxGAN	COSINE	7.60	16.04	11.93
LSGAN	COSINE	7.92	15.63	11.74
RelGAN	COSINE	8.01	16.22	12.21
FuseGAN	**COSINE**	**6.93**	**14.77**	**10.88**

Tables 1 & 2. compare the performance of the different speaker representations on the NIST-SRE16 task. Among the baseline systems the x-vector model produces the best results, however requires LDA based dimensionality reduction and the PLDA classifier to produce its best result. We see that all of the GAN based models outperform gradient reversal by a large margin, but none of the individual models are able to match the best x-vector system. Interestingly, we find that we are able to best this system by simply averaging the scores of our different GAN models. The FuseGAN results do not include the scores from the standard GAN model, although this does not affect the final performance significantly.

6 Analysis

One particularly interesting result from our experiments is the improvement we see through a simple score averaging procedure. Our hypothesis is that the different discriminator objectives encourage the generator to cover different modes of the target data distribution. This finding is consistent with GAN

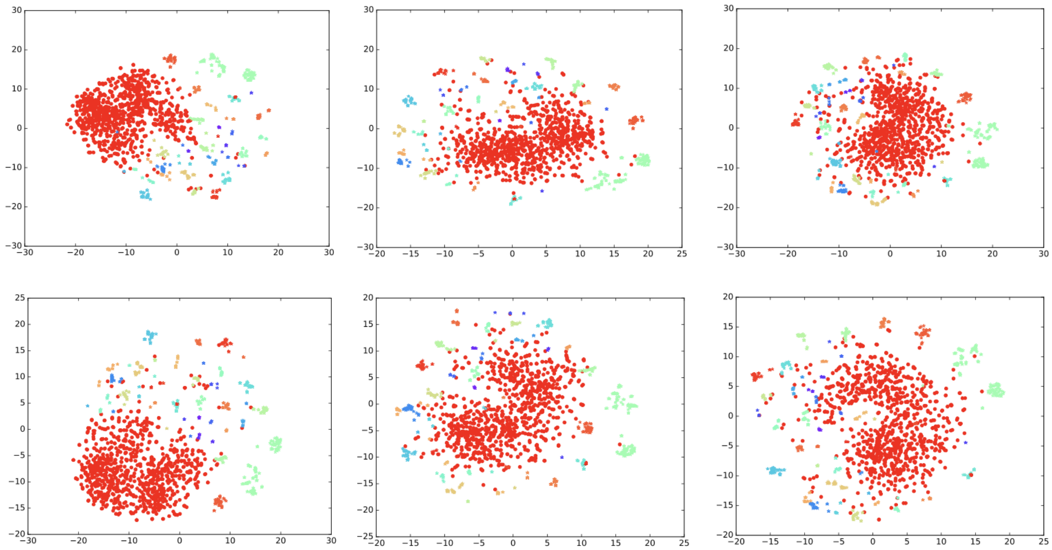

Figure 2: t-SNE visualization of embedding space. Large red cluster represents target data. **top row:** No Adaptation, standard GAN, auxGAN **bottom:** Grad Reversal, Relativistic, auxGAN (ours) (left to right).

approaches that train multiple discriminators [15], although we do not train them simultaneously. In Fig. 2 we visualize the embedding spaces learned by our models using t-SNE [12]. While Gradient Reversal primarily appears to rotate the feature space, the transformations induced by the GAN models is more pronounced. Crucially, we see that that the source domain speaker clusters appear to remain intact. This indicates that our models retain discriminative properties in the source domain, a fact we verify experimentally.

Maximum Mean Discrepancy (MMD): is based on the idea that two distributions are identical if and only if all their moments are identical [7]. A divergence can be defined if we can measure how "different" the moments of the two distributions are. MMD is a method of efficiently doing this via the kernel trick:

$$\begin{aligned} &MMD(p(z)||q(z)) = \\ &\mathbb{E}_{p(z),p(z')}[k(z,z^{'})] + \mathbb{E}_{q(z),q(z')}[k(z,z^{'})] - 2\mathbb{E}_{p(z),q(z')}[k(z,z^{'})] \end{aligned} \tag{10}$$

In order to quantitatively evaluate our models in terms of domain adaptation, we measure the Maximum Mean Discrepancy distance between a selection of source data and the unlabelled target data. MMD is a standard distribution distance metric and has been applied in the context of domain adaptation [22].

Fréchet Distance: The Fréchet Inception Distance (fid) is a popular approach for evaluating GANs, and has been shown to correlate well with human judgement of visual quality [8]. Instead of an Inception network, we extract embeddings from our gan models from the source and target data. The Fréchet Distance between between the Gaussian (m_s,C_s) obtained from the source data distribution p_s and the Gaussian (m_t,C_t) from the target data is given by:

$$d^2((\mathbf{m}_s, \mathbf{C}_s, (\mathbf{m}_t, \mathbf{C}_t)) = ||\mathbf{m}_s - \mathbf{m}_t||_2^2 + Tr(\mathbf{C}_s + \mathbf{C}_t - 2(\mathbf{C}_s\mathbf{C}_t)^{1/2}) \tag{11}$$

Source Domain Speaker Verification: We use the same source data used to compute the MMD and Fréchet Distance to construct a trial list for verification. The list consists of 2500 recordings and we score them all versus all. There are a total of 101,666 target and 6,145,834 non-target trials.

From Fig. 3 we see that MMD and the Fréchet distance display similar trends. Surprisingly we see that Gradient Reversal only has a small effect on either metric, while the GAN models all have much lower MMD and Fréchet distances. We note that the model using the novel generator objective shows

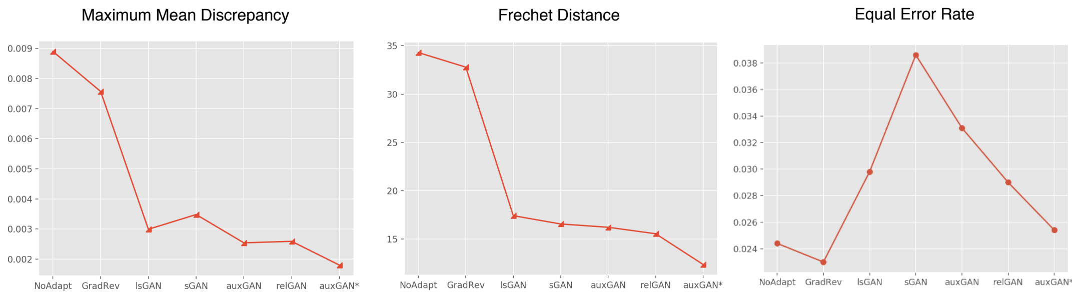

Figure 3: Comparing Models in terms of MMD, Frćhet distances and source domain verification. **NoAdapt:** No Adaptation, **GradRev:** Gradient Reversal, **lsGAN:** Least Squares, **sGAN:** standard, **auxGAN:** auxiliary classifier, **relGAN:** Relativistic, **auxGAN***:proposed objective.

the lowest scores on both metrics. The results on source domain speaker verification also indicate that our models remain discriminative in the source domain as well, with only a small degradation as compared to the unadapted model. The vanilla GAN performs worst on the verification task, and this relative performance also translates to the target domain. Interestingly, the Gradient Reversal model shows the best performance on this experiment albeit by a small margin.

7 Conclusion

In this work we we presented a novel framework for learning domain-invariant speaker embeddings using GANs. By combining a powerful deep feature extractor, an end-to-end loss function and most importantly, adversarial training we are able to learn extremely compact speaker embeddings that deliver robust verification performance on challenging evaluation data. We showed that the proposed methods do reduce the domain mismatch between source and target data in terms of MMD and Fréchet distance. Furthermore, we see that our methods adapt while maintaining their speaker discriminative nature in the source domain as well. In future work we will experiment with other GAN variants in an attempt to further improve performance. Given the success of our simple fusion approach, we believe that exploring models with multiple discriminators could be an interesting research direction.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[2] Gautam Bhattacharya, Jahangir Alam, and Patrick Kenny. Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training. In *Acoustics Speech and Signal Processing (ICASSP), 2019 IEEE International Conference on (Submitted)*. IEEE, 2019.

[3] Gautam Bhattacharya, Md Jahangir Alam, Vishwa Gupta, and Patrick Kenny. Deeply fused speaker embeddings for text-independent speaker verification. *Proc. Interspeech 2018*, pages 3588–3592, 2018.

[4] Gautam Bhattacharya, Joao Montiero, Jahangir Alam, and Patrick Kenny. Generative adversarial speaker embedding networks for domain-robust end-to-end speaker verification. In *Acoustics Speech and Signal Processing (ICASSP), 2019 IEEE International Conference on (Submitted)*. IEEE, 2019.

[5] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[7] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

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

[10] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. Deep speaker: an end-to-end neural speaker embedding system. *arXiv preprint arXiv:1705.02304*, 2017.

[11] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 1, 2017.

[12] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[13] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE, 2017.

[14] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson. The speakers in the wild (sitw) speaker recognition database. In *Interspeech*, pages 818–822, 2016.

[15] Behnam Neyshabur, Srinadh Bhojanapalli, and Ayan Chakrabarti. Stabilizing gan training with multiple random projections. *arXiv preprint arXiv:1705.07831*, 2017.

[16] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

[17] Seyed Omid Sadjadi, Timothée Kheyrkhah, Audrey Tong, Craig Greenberg, Elliot Singer Reynolds, Lisa Mason, and Jaime Hernandez-Cordero. The 2016 nist speaker recognition evaluation. In *Proc. Interspeech*, pages 1353–1357, 2017.

[18] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *ArXiv e-prints, abs/1704.01705*, 2017.

[19] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*, 2018.

[20] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. *ICASSP, Calgary*, 2018.

[21] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.

[22] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[23] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[24] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[25] Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey. Self-attentive speaker embeddings for text-independent speaker verification. *Proc. Interspeech 2018*, pages 3573–3577, 2018.