*INTERSPEECH 2019*
September 15–19, 2019, Graz, Austria

# Unsupervised Adaptation with Adversarial Dropout Regularization for Robust Speech Recognition

*Pengcheng Guo, Sining Sun, Lei Xie*[*]

Shaanxi Provincial Key Laboratory of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China

guopengcheng1220@gmail.com, snsun@nwpu-aslp.org, lxie@nwpu.edu.cn

## Abstract

Recent adversarial methods proposed for unsupervised domain adaptation of acoustic models try to fool a specific domain discriminator and learn both senone-discriminative and domain-invariant hidden feature representations. However, a drawback of these approaches is that the feature generator simply aligns different features into the same distribution without considering the class boundaries of the target domain data. Thus, ambiguous target domain features can be generated near the decision boundaries, decreasing speech recognition performance. In this study, we propose to use Adversarial Dropout Regularization (ADR) in acoustic modeling to overcome the foregoing issue. Specifically, we optimize the senone classifier to make its decision boundaries lie in the class boundaries of unlabeled target data. Then, the feature generator learns to create features far away from the decision boundaries, which are more discriminative. We apply the ADR approach on the CHiME-3 corpus and the proposed method yields up to 12.9% relative WER reductions compared with the baseline trained on source domain data only and further improvement over the widely used gradient reversal layer method.

**Index Terms**: robust speech recognition, adversarial training, domain adaptation

## 1. Introduction

In recent years, the performance of automatic speech recognition (ASR) has been improved dramatically due to the advances of deep learning [1, 2, 3]. However, ASR systems can be susceptible to performance degradation when substantial mismatch exists between the training and test conditions [4]. These mismatches, including but not limited to speaker variations, channel distortions and acoustic noises, can be narrowed by adding more data for a better coverage, but can not be fully avoided.

*Domain adaptation* is a popular way to alleviate this mismatch and to achieve a more robust acoustic model (AM). There has been plenty of work focused on using a small size of labeled target domain data to adjust AM parameters or input features to compensate the mismatch. When fine-tuning the original model using a small set of data, it may easily raise over-fitting problem. To alleviate the problem, regularization approaches are proposed to bias the output distributions or the network parameters to the desired direction [5, 6]. Or we can only adapt a small set of target domain specific parameters by linear transformations [7, 8] or singular value decomposition (SVD) [9, 10].

In practice, labeled data from target domain is not always available or it is too expensive to collect. Hence, unsupervised domain adaptation becomes an alternative, by definition, which aims to transfer the AM from label-rich source domain to unlabeled target domain. Swietojanski et al. proposed to learn speaker specific hidden unit contributions through adding amplitude parameters to the network [11]. In [12], teacher-student (T/S) training strategy was introduced to achieve unsupervised domain adaptation. Specifically, posteriors generated by the source domain teacher model in lieu of the transcribed labels were used to train the target domain student model. Hsu et al. used a data augmentation approach via a variational autoencoder (VAE) that learns latent representations and then transforms the representations of the source domain data to be more similar to the target domain [13].

Recently, *domain adversarial training* (DAT) has become an increasingly popular idea for unsupervised domain adaptation because of its great success in estimating generative models [14]. In typical DAT approaches [15, 16, 17, 18], the deep neural network (DNN) is transformed to three sub-networks with different learning purposes: a classifier $C$ (predicting senones) works in parallel with a domain discriminator $D$ (predicting the input whether from source or target domain) and a feature generator $G$ is shared at the bottom. $G$ and $C$ are trained to minimize the class prediction error and maximize the domain prediction error respectively, while $D$ is updated to minimize the domain prediction error. In a form of multi-task learning, a gradient reverse layer (GRL) is inserted to learn both class-discriminative and domian-invariant intermediate representations. Following this idea, there has been several variants recently [19, 20, 21].

These DAT approaches are all motivated by the theoretical assumption that minimizing the divergence between domains can lower the upper bound of the error on the target domain [22]. Therefore, they separate the neural network into different components and attempt to reduce the divergence via adversarial learning. However, according to [23], a drawback of these approaches is that $D$ simply labels the generated features as source domain or not, without considering the boundaries between different classes. More specifically, although $G$ trained under this adversarial criterion can force the features from the two domains yielding similar distribution, it can not guarantee the features' discriminative attributes to the classification task, especially in the target domain. Thus $G$ may create features close to the class boundaries, which are more likely to be misclassified by $C$. In [23], Satio et al. made the observation that if $C$ can detect non-discriminative samples near the decision boundaries and $G$ tries to avoid these areas of the feature space, the generated features will be more discriminative. To achieve this, they removed the domain discriminator $D$ and introduced a new approach for aligning feature distributions called adversarial dropout regularization (ADR).

In this study, we propose to use ADR to increase the ro-

[*] Corresponding author.

http://dx.doi.org/10.21437/Interspeech.2019-2544

bustness of AM under the unsupervised domain adaptation scenario. The theoretical assumption of ADR is that if the decision boundaries of $C$ are close to class boundaries of unlabeled target domain data, a small perturbation for $C$ will lead to a substantial change of the output distribution. ADR aims to learn a $G$ which can create representations far from the decision boundaries to avoid the class ambiguous problem. Specially, we slightly move the decision boundaries of $C$ by using dropout and optimize $C$ to increase the change of posterior probabilities. This step finds a $C$ whose decision boundaries lie in the class boundaries of unlabeled target domain data. Then, we update $G$ to decrease the change by pulling the generated features far away from the decision boundaries. Through this two-stage iterative adversarial learning strategy, $G$ learns to create more discriminative features, especially for the target data. The benefit of ADR is that we align feature distributions with feedback from $C$ instead of using $D$, which is more relevant to the classification task. On the CHiME-3 evaluation set, the proposed method achieves 12.9% relative WER reduction compared with the baseline model and it also outperforms the GRL based DAT approach.

## 2. Adversarial Dropout Regularization

In unsupervised domain adaptation task, we only have access to a source sample $\mathbf{x_s}$ and its corresponding label $y_s$ drawn from labeled source domain data $\{\mathbf{X_s}, \mathbf{Y_s}\}$, as well as a target sample $\mathbf{x_t}$ drawn from unlabeled target domain data $\{\mathbf{X_t}\}$. In order to achieve good recognition performance on target domain, the adapted AM should not only consider the mismatch between two domain distributions, but also take the senone boundaries into account, especially for unlabeled target domain data. The problem is, it is not clear how this can be accomplished without labels of target domain data. Support vector machine (SVM) is known as a maximum-margin classifier [24, 25], which aims to make the decision boundaries robust. Different with it, we train the classifier as a *minimum-margin* classifier to find not robust decision boundaries lying in the senone boundaries, as shown in Fig. 1 (Left). Next, the feature generator learns to create features far away from the decision boundaries and make decision boundaries more robust, as illustrated in Fig. 1 (Right).

In particular, we first train a feature generator network $G$, which takes $\mathbf{x_s}$ or $\mathbf{x_t}$ as the inputs, and a classifier network $C$ which aims to predict senone labels of source samples and detect non-discriminative target samples close to the decision boundaries. $C$ takes features from $G$ and maps them into $K$ classes, predicting a $K$-dimensional posterior senone probability. When $C$ acts as senone classifier, $G$ is optimized to obtain discriminative features for source domain data and $C$ is trained to classify them correctly. When $C$ acts as a minimum-margin classifier for target domain data, we slightly move the senone decision boundaries of $C$ through dropout and measure the change of posterior probabilities. In fact, the posterior change is inversely proportional to the distance from the boundaries. Then, we fix the parameters of $G$ (fix the data distribution) and optimize $C$ (change the decision boundary) to maximize the discrepancy between two probabilities, which makes $C$ be more sensitive to target domain features near the decision boundaries. Finally, $G$ is trained to minimize the discrepancy under a fixed sensitive $C$. Through this adversarial training, $G$ will learn to create discriminative target domain features far away from the decision boundaries. In this following, we will show how to use ADR in unsupervised domain adaptation for acoustic modeling in detail.

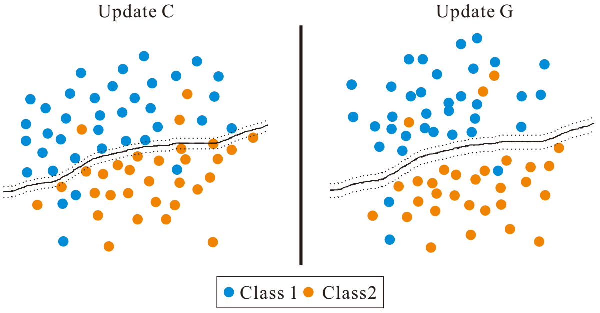

Figure 1: *Data distributions after updating C and updating G. The solid line means senone decision boundary while the dotted line means a slight movement.* ***Left:*** *We update C to find a decision boundary lies in the class boundaries of unlabeled target data and detect non-discriminative features.* ***Right:*** *We update G to generate discriminative features away from decision boundaries.*

### 2.1. Move Decision Boundaries via Dropout

Dropout was firstly proposed in [26, 27], which prevents the neural networks from over-fitting and increases their robustness through randomly dropping out nodes during training. Consider the standard utilization of dropout. For every sample within a mini-batch, each node of the network is removed with some probability, effectively selecting a different classifier for every sample. We harness this idea as a slightly movement of decision boundaries. We first forward input features $G(\mathbf{x_t})$ to $C$ twice, dropping different nodes and obtain two different classifiers $C_1$ and $C_2$, as shown in Fig 2. The output posterior probabilities of them are denoted as $p_1(\mathbf{y}|\mathbf{x_t})$ and $p_2(\mathbf{y}|\mathbf{x_t})$. Then, we measure the discrepancy of two probabilities using the absolute distance, which is calculated as

$$Dis(p_1, p_2) = \|p_1 - p_2\|_2, \tag{1}$$

or the symmetric kullback leibler (KL) divergence [28], which is calculated as

$$Dis(p_1, p_2) = \frac{1}{2}(KL(p_1|p_2), KL(p_2|p_1)), \tag{2}$$

where $KL(p_1|p_2)$ means the KL divergence between $p_1$ and $p_2$. As mentioned above, $C$ is optimized to increase the discrepancy for target domain data.

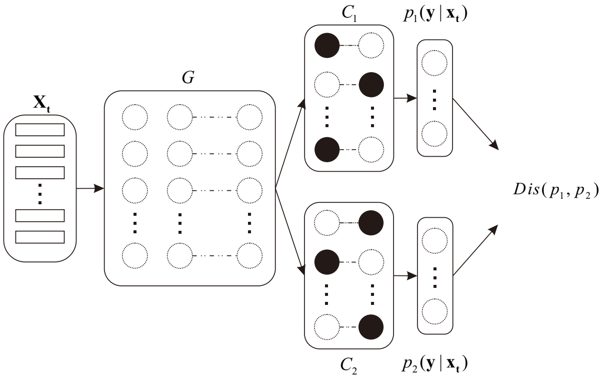
$\mathbf{X}_t$
$G$
$C_1$
$p_1(\mathbf{y}|\mathbf{x}_t)$
$Dis(p_1, p_2)$
$C_2$
$p_2(\mathbf{y}|\mathbf{x}_t)$

Figure 2: *Using dropout to select two different classifiers.*

### 2.2. Training procedure

In order to obtain more discriminative features for both source and target domain frames, the whole training procedure of unsupervised domain adaptation with ADR consists of three steps:

**Step 1**, use labeled source domain data to optimize $G$ and $C$ following a standard ASR training setup. Given source samples $\mathbf{X_s}$ and their labels $\mathbf{Y_s}$, the objective function can be defined as

$$\min_{G,C} L_{cls} = -\mathbb{E}_{(\mathbf{x_s},y_s)\sim(\mathbf{X_s},\mathbf{Y_s})} \sum_{k=1}^{K} y_k \log C(G(\mathbf{x_s}))_k, \quad (3)$$

where $C(G(\mathbf{x_s}))_k$ refers to the posterior class probability of frame $\mathbf{x_s}$ belongs to senone $k$.

**Step 2**, make $C$ be sensitive to non-discriminative target features near the decision boundaries. Here, two classifiers are sampled from $C$ for each target sample using dropout. Then, $C$'s parameters are updated to maximize the discrepancy between two posterior probabilities as denoted by Eq. (1) or (2). Since $C$ should classify source samples correctly, we add Eq. (3) as a constraint item to the objective function in this step. Thus, the objective function becomes

$$\min_{C} L_{adv_C} = L_{cls} - \mathbb{E}_{(\mathbf{x_t})\sim(\mathbf{X_t})} Dis(p_1, p_2). \quad (4)$$

**Step 3**, train $G$ to generate feature representations far away from the decision boundaries. The objective function is

$$\min_{G} L_{adv_G} = \mathbb{E}_{(\mathbf{x_t})\sim(\mathbf{X_t})} Dis(p_1, p_2). \quad (5)$$

Algorithm 1 outlines the acoustic model training integrated with ADR. For each iteration, we feed both source and target domain data into the neural network. The parameters of $G$ and $C$ are optimized by the 3-step optimization criterion. Besides, we find it will be beneficial to repeat Step 3 several times in one training iteration. This results in the $G$ being maintained near its optimal solution and creating more discriminative features far away from decision boundaries.

## 3. Experiments

In this work, we investigate the unsupervised domain adaptation of DNN-HMM acoustic model with ADR for robust speech recognition on CHiME-3 data set.

### 3.1. CHiME-3 Dataset

The CHiME-3 task is a speech recognition challenge for single microphone or multi-microphone tablet device recordings in everyday scenarios under noisy environments [29]. There are 4 noisy recording environments in the dataset: street (STR), pedestrian area (PED), cafe (CAF) and bus (BUS). The noisy training set includes 1600 utterances recorded in 4 real noisy environments from 4 speakers, and additional 7138 noisy utterances simulated from a part of the Wall Street Journal (WSJ0) corpus by adding noises from the 4 noisy environments. The development set consists of 410 utterances in each of the 4 noisy environments with both real and simulated environments, for a total of 3280 utterances. There are 2640 utterances in the evaluation set, with 330 utterances in each of the same 8 conditions. The training, development and evaluation sets are all recorded in 6 different channels.

In the experiments, we collect 7138 clean utterances from WSJ0 together with 1600 close-talking utterances from CHiME-3 to form the clean training set. Then, we extract 8738 noisy utterances from the 5th channel of CHiME3 noisy training set and define them as unlabeled adaptation set. Hence, the source domain is clean condition while the target domain refers to noisy condition. The WSJ 5K words 3-gram language model is used for decoding.

**Algorithm 1** Training acoustic models using ADR

```
Initialize model parameters $\boldsymbol{\theta}$ and let $iter$ equals to 0
Given hyper parameters
    • learning rate $\alpha$ for updating parameters of $G$ and $C$
    • dropout rate $p$ for $C$
    • number of repeated times $N$ of Eq. 5
Loading pre-trained model $G$ and $C$ or not
while not converge do
    Get a mini-batch labeled source samples $S = \{\mathbf{x_s^m}, \mathbf{y_s^m}\}_{m=1}^{M}$ and a mini-batch unlabeled target samples $T = \{\mathbf{x_t^m}\}_{m=1}^{M}$
    Forward the network using source samples $S$
    Update $G$ and $C$ using Eq. 3
    Forward the network using source samples $S$ and compute the cross entropy loss using Eq. 3
    Forward the network twice using target samples $T$ and obtain two posterior probabilities. Compute the discrepancy between them as in Eq. (1) or Eq. (2)
    Update $C$ using Eq. 4
    for ($i = 0$; $i <$ N; $i$++) do
        Update $G$ using Eq. 5
    end for
    $iter$ = $iter$ + 1
end while
return $\boldsymbol{\theta}$
```

### 3.2. System Description

We first train a DNN-HMM acoustic model with clean training set as the baseline system. The DNN model has 6 hidden layers with 2048 hidden units for each layer. The output layer has 2000 output units corresponding to 2000 senone labels. Then, both training set and adaptation set are used to achieve the GRL based adversarial domain adaption as described in [16]. The feature generator of GRL model has 4 hidden layers with 2048 hidden units, while the senone classifier has 2 hidden layers with 2048 hidden units and an output layer with 2000 output units. The domain discriminator has 2 hidden layers with 512 hidden units and an output layers with 2 output units representing source and target domain. A GRL layer is inserted between the generator and discriminator in order to obtain domain-invariant features. Finally, the same configurations in GRL model are adopted for adversarial dropout regularization (ADR) training, except the removal of the domain discriminator. No senone alignments of the noisy adaptation set is used for both GRL model and ADR model.

The 40-dimensional mel-filter bank (fbank) features together with 1st and 2nd order delta features for both the clean and noisy utterances are extracted as the input for all systems. Each frame is normalized by CMVN and spliced together with 5 left and 5 right context frames to form a 1320-dimensional feature. Standard recipes in Kaldi [30] are adopted for feature extraction, HMM-GMM training and alignment generation of clean training set. PyTorch [31] is used for training different

Table 1: *The WER (%) performance and relative improvement (Rel.) of baseline (trained with clean data only), GRL and ADR acoustic models for robust ASR on development and evaluation sets of CHiME-3 dataset.*

| System | Mode | Dev Set | | | | | | Eval Set | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Rel. | Avg. | BUS | CAF | PED | STR | Rel. | Avg. | BUS | CAF | PED | STR |
| **Baseline** | **Simu** | - | 35.29 | 31.83 | 43.50 | 31.33 | 34.51 | - | 46.24 | 47.01 | 49.79 | 46.32 | 41.82 |
| | **Real** | - | 40.76 | 56.11 | 40.46 | 30.24 | 36.22 | - | 65.82 | 88.20 | 67.50 | 64.22 | 43.37 |
| **GRL** | **Simu** | 12.8 | 30.79 | 29.17 | 38.33 | 26.56 | **29.09** | 11.4 | 40.98 | 41.05 | 44.83 | **40.62** | 37.41 |
| | **Real** | 6.4 | 38.17 | 52.49 | 40.19 | 26.69 | 33.30 | 6.7 | 61.38 | 83.66 | 65.20 | **56.89** | 39.78 |
| **ADR** | **Simu** | **16.3** | **29.53** | **27.73** | **36.65** | **24.09** | 29.65 | **12.9** | **40.26** | **38.91** | **43.87** | 41.17 | **37.11** |
| | **Real** | **13.2** | **35.37** | **50.51** | **35.35** | **22.98** | **32.64** | **9.3** | **59.71** | **81.88** | **60.46** | 57.32 | **39.19** |

models in this work with cross-entropy as objective function and Adam [32] as the optimizer. We set the initial learning rate to $2.0 \times 10^{-4}$ and use the same learning rate schedule as Kaldi nnet1. The values of hyper parameters for both tasks are tuned on the respective development set, and the best parameters are then applied to the evaluation sets.

### 3.3. Experimental Result

The motivation of our work is to make the non-discriminative target domain feature (near the senone decision boundaries) to be more discriminative (away from the senone decision boundaries). We suppose that if we slightly move the decision boundaries, the posterior probabilities of those features are likely to have the largest change, reducing the performance of classification. To confirm the assumption, we select two typical utterances from the real noisy condition development set (dt05_real_noisy) and observe the discrepancy of output labels predicted by a well-trained baseline model with slight movement of decision boundaries. One of them has a low WER (equals to 0.0%), which means its features are more discriminative, while another has a high WER (equals to 100%), which means its features are more ambiguous. In Fig. 3 (Top), two outputs are similar and the correlation coefficient is very high. It is clear that the discriminative features are far away from the senone boundaries and robust to the movement of decision boundaries. However, for high WER utterance in Fig. 3 (Bottom), great changes have taken place, which demonstrate the ambiguous features are sensitive to the slight boundaries movement and degrade the classification performance.

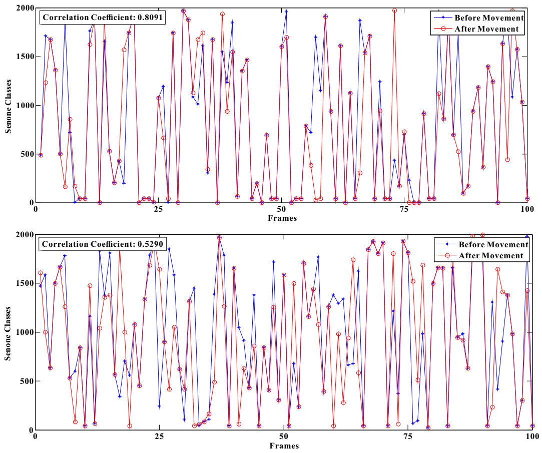

Figure 3: *The change of output labels predicted by the baseline acoustic model when slightly move the decision boundaries.* ***Top:*** *High WER utterance.* ***Bottom:*** *Low WER utterance.*

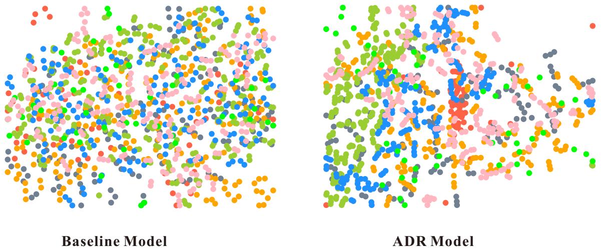

Figure 4: *Visualization of target features using T-SNE [33].* ***(a)*** *Features obtained by baseline model;* ***(b)*** *Features obtained by ADR model.*

Fig.4 visualized the target domain features obtained by $G$ with baseline model and ADR model. While the embedding of the baseline model does not separate classes well due to domain shift, we can see clearly improved separation with ADR.

Table 1 shows the WER performance of different acoustic models: baseline, GRL method trained and ADR method trained, for different environment evaluation sets. The baseline model achieves 46.24% and 65.82% WERs on the simulated and real noisy evaluation sets respectively. The GRL model achieves 40.98% and 61.38% WERs on those sets. The best WER performance for ADR model are 40.26% and 59.71% , which achieves 12.9% and 9.3% relative improvement over the baseline model and achieves 1.8% and 2.5 % further relative improvement over the GRL model. Besides, the proposed approach is also able to bring consistent improvements for all types of noises, whether in simulated or real environments.

## 4. Conclusions

In this study, we propose to use ADR to solve the domain mismatch problem for robust speech recognition. The proposed method consists of a classifier to detect non-discriminative features close to the class boundaries and a generator to avoid those ambiguous feature space. Experimental results on CHiME-3 dataset show the proposed method can improve the robustness of acoustic model achieving 12.9% relative WER reduction on CHiME-3 real test set compared with baseline model and consistent improvement over the GRL model. Inspired by this work, we will try to combine the class boundaries information with different domain private information to improve the robustness of acoustic model in future work.

## 5. Acknowledgements

This research work is supported by the National Natural Science Foundation of China (No.61571363).

## 6. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[2] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams *et al.*, "Recent advances in deep learning for speech research at microsoft," in *Proc. ICASSP*. IEEE, 2013, pp. 8604–8608.

[3] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 396–409, 2017.

[4] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[5] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. ICASSP*. IEEE, 2013, pp. 7893–7897.

[6] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 7947–7951.

[7] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Linear hidden transformations for adaptation of hybrid ann/hmm models," *Speech Communication*, vol. 49, no. 10-11, pp. 827–835, 2007.

[8] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*. IEEE, 2011, pp. 24–29.

[9] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *Proc. INTERSPEECH*. ISCA, 2013, pp. 2365–2369.

[10] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Proc. ICASSP*. IEEE, 2014, pp. 6359–6363.

[11] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.

[12] J. Li, M. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," in *Proc. INTERSPEECH*. ISCA, 2017, pp. 2386–2390.

[13] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," in *Proc. ASRU*. IEEE, 2017, pp. 16–23.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[15] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[16] S. Sun, B. Zhang, L. Xie, and Y. Zhang, "An unsupervised deep domain adaptation approach for robust speech recognition," *Neurocomputing*, vol. 257, pp. 79–87, 2017.

[17] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 4889–4893.

[18] J. Hou, P. Guo, S. Sun, F. K. Soong, W. Hu, and L. Xie, "Domain adversarial training for improving keyword spotting performance of esl speech," in *Proc. ICASSP*. IEEE, 2019, pp. 8122–8126.

[19] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. NIPS*, 2016, pp. 343–351.

[20] Z. Meng, Z. Chen, V. Mazalov, J. Li, and Y. Gong, "Unsupervised adaptation with domain separation networks for robust speech recognition," in *Proc. ASRU*. IEEE, 2017, pp. 214–221.

[21] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. CVPR*. IEEE, 2017, pp. 7167–7176.

[22] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[23] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Adversarial dropout regularization," in *Proc. ICLR*, 2018.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[25] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. ICML*, 1999, pp. 200–209.

[26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A kullback-leibler divergence based kernel for svm classification in multimedia applications," in *Proc. NIPS*, 2004, pp. 1385–1392.

[29] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'chime' speech separation and recognition challenge: dataset, task and baselines," in *Proc. ASRU*. IEEE, 2015, pp. 504–511.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Proc. NIPS*, 2017.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.