UNSUPERVISED ADAPTATION WITH DOMAIN SEPARATION NETWORKS FOR ROBUST SPEECH RECOGNITION

Zhong Meng[1,2*], *Zhuo Chen*[1], *Vadim Mazalov*[1], *Jinyu Li*[1], *Yifan Gong*[1]

[1] Microsoft AI and Research, Redmond, WA
[2] Georgia Institute of Technology, Atlanta, GA

ABSTRACT

Unsupervised domain adaptation of speech signal aims at adapting a well-trained source-domain acoustic model to the unlabeled data from target domain. This can be achieved by adversarial training of deep neural network (DNN) acoustic models to learn an intermediate deep representation that is both senone-discriminative and domain-invariant. Specifically, the DNN is trained to jointly optimize the primary task of senone classification and the secondary task of domain classification with adversarial objective functions. In this work, instead of only focusing on learning a domain-invariant feature (i.e. the shared component between domains), we also characterize the difference between the source and target domain distributions by explicitly modeling the private component of each domain through a private component extractor DNN. The private component is trained to be orthogonal with the shared component and thus implicitly increases the degree of domain-invariance of the shared component. A reconstructor DNN is used to reconstruct the original speech feature from the private and shared components as a regularization. This domain separation framework is applied to the unsupervised environment adaptation task and achieved 11.08% relative WER reduction from the gradient reversal layer training, a representative adversarial training method, for automatic speech recognition on CHiME-3 dataset.

***Index Terms*—** robust speech recognition, deep neural networks, domain adaptation, adversarial training, multi-task training

1. INTRODUCTION

In recent years, advances in deep learning have led to remarkable performance boost in automatic speech recognition (ASR) [1, 2, 3, 4, 5, 6]. However, ASR systems still suffer from large performance degradation when acoustic mismatch exists between the training and test conditions [7, 8]. Many factors contribute to the mismatch, such as variation in environment noises, channels and speaker characteristics. Domain adaptation is an effective way to address this limitation, in which the acoustic model parameters or input features are adjusted to compensate for the mismatch.

One difficulty with domain adaptation is that available data from the target domain is usually limited, in which case the acoustic model can be easily overfitted. To address this issue, regularization-based approaches are proposed in [9, 10, 11, 12] to regularize the neuron output distributions or the model parameters. In [13, 14], transformation-based approaches are introduced to reduce the number of learnable parameters. In [15, 16, 17], the trainable parameters are further reduced by singular value decomposition of weight matrices of a neural network. Although these methods utilize the limited data from the target domain, they still require labelling for the adaptation data and can only be used in supervised adaptation.

Unsupervised domain adaptation is necessary when human labelling of the target domain data is unavailable. It has become an important topic with the rapid increase of the amount of untranscribed speech data for which the human annotation is expensive. Pawel et al. proposed to learn the contribution of hidden units by additional amplitude parameters [18] and differential pooling [19]. Recently, Wang et al. proposed to adjust the linear transformation learned by batch normalized acoustic model in [20]. Although these methods lead to increased performance in the ASR task when no labels are available for the adaptation data, they still rely on the senone (tri-phone state) alignments against the unlabeled adaptation data through first pass decoding. The first pass decoding result is unreliable when the mismatch between the training and test conditions is significant. It is also time-consuming and can be hardly applied to huge amount of adaptation data. There are even situations when decoding adaptation data is not allowed because of the privacy agreement signed with the speakers. These methods depending on the first pass decoding of the unlabeled adaptation data is sometimes called "semi-supervised" adaptation in literature.

The goal of our study is to achieve *purely* unsupervised domain adaptation *without* any exposure to the labels or the decoding results of the adaptation data in the target domain. In [21] we show that the source-domain model can be effectively adapted without any transcription by using teacher-student (T/S) learning [22], in which the posterior probabil-

*Zhong Meng performed the work while he was a research intern at Microsoft AI and Research, Redmond, WA.

ities generated by the source-domain model can be used in lieu of labels to train the target-domain model. However, T/S learning relies on the availability of parallel unlabeled data which can be usually simulated. However, if parallel data is not available, we cannot use T/S learning for model adaptation. In this study, we are exploring the solution to domain adaptation without parallel data and without transcription. Recently, adversarial training has become a very hot topic in deep learning because of its great success in estimating generative models [23]. It was first applied to the area of unsupervised domain adaptation by Ganin et al. in [24] in a form of multi-task learning. In their work, the unsupervised adaptation is achieved by learning deep intermediate representations that are both discriminative for the main task (image classification) on the source domain and invariant with respect to mismatch between source and target domains. The domain invariance is achieved by the adversarial training of the domain classification objective functions. This can be easily implemented by augmenting any feed-forward models with a few standard layers and a *gradient reversal layer (GRL)*. This GRL approach has been applied to acoustic models for unsupervised adaptation in [25] and for increasing noise robustness in [26, 27]. Improved ASR performance is achieved in both scenarios.

However, the GRL method focuses only on learning a domain-invariant representation, ignoring the unique characteristics of each domain, which could also be informative. Inspired by this, Bousmailis et al. [28] proposed the *domain separation networks (DSNs)* to separate the deep representation of each training sample into two parts: one private component that is unique to its domain and one shared component that is invariant to the domain shift. In this work, we propose to apply DSN for unsupervised domain adaptation on a DNN-hidden Markov model (HMM) acoustic model, aiming to increase the noise robustness in speech recognition. In the proposed framework, the shared component is learned to be both senone-discriminative and domain-invariant through adversarial multi-task training of a shared component extractor and a domain classifier. The private component is trained to be orthogonal with the shared component to implicitly increase the degree of domain-invariance of the shared component. A reconstructor DNN is used to reconstruct the original speech feature from the private and shared components, serving for regularization. The proposed method achieves 11.08% relative WER improvement over the GRL training approach for robust ASR on the CHiME-3 dataset.

2. DOMAIN SEPARATION NETWORKS

In the *purely* unsupervised domain adaptation task, we only have access to a sequence of speech frames $X^s = \{x_1^s, \dots, x_{N_s}^s\}$ from the source domain distribution, a sequence of senone labels $Y^s = \{y_1^s, \dots, y_{N_s}^s\}$ aligned with source data X^s and a sequence of speech frames $X^t = \{x_1^t, \dots, x_{N_t}^t\}$ from a target domain distribution. Senone labels or other types of transcription are *not* available for the target speech sequence X^t.

When applying domain separation networks (DSNs) to the unsupervised adaptation task, our goal is to learn the shared (or common) component extractor DNN M_c that maps an input speech frame x^s from source domain or x^t from target domain to a *domain-invariant* shared component f_c^s or f_c^t respectively. At the same time, learn a senone classifier DNN M_y that maps the shared component f_c^s from the source domain to the correct senone label y^s.

To achieve this, we first perform adversarial training of the domain classifier DNN M_d that maps the shared component f_c^s or f_c^t to its domain label d^s or d^t, while simultaneously minimizing the senone classificaton loss of M_y given shared component f_c^s from the source domain to ensure the *senone-dicriminativeness* of f_c^s.

For the source or the target domain, we extract the source or the target private component f_p^s or f_p^t that is unique to the source or the target domain through a source or a target private component extractor M_p^s or M_p^t. The shared and private components of the same domain are trained to be orthogonal to each other to further enhance the degree of domain-invariance of the shared components. The extracted shared and private components of each speech frame are concatenated and fed as the input of a reconstructor M_r to reconstruct the input speech frame x^s or x^t.

The architecture of DSN is shown in Fig. 1, in which all the sub-networks are jointly optimized using SGD. The optimized shared component extractor M^c and senone classifier M_y form the adapted acoustic model for subsequent robust speech recognition.

2.1. Deep Neural Networks Acoustic Model

The shared component extractor M_c and senone predictor of the DSN are initialized from an DNN-HMM acoustic model. The DNN-HMM acoustic model is trained with labeled speech data (X^s, Y^s) from the source domain. The senone-level alignment Y_s is generated by a well-trained GMM-HMM system.

Each output unit of the DNN acoustic model corresponds to one of the senones in the set $\mathcal{Q}$. The output unit for senone $q \in \mathcal{Q}$ is the posterior probability $p(q|x_n^s)$ obtained by a softmax function.

2.2. Shared Component Extraction with Adversarial Training

The well-trained acoustic model DNN in Section 2.1 can be decomposed into two parts: a share component extractor M_c with parameters θ_c and a senone classifier M_y with parameters θ_y. An input speech frame from source domain x^s is first mapped by the M_c to a K-dimensional shared component

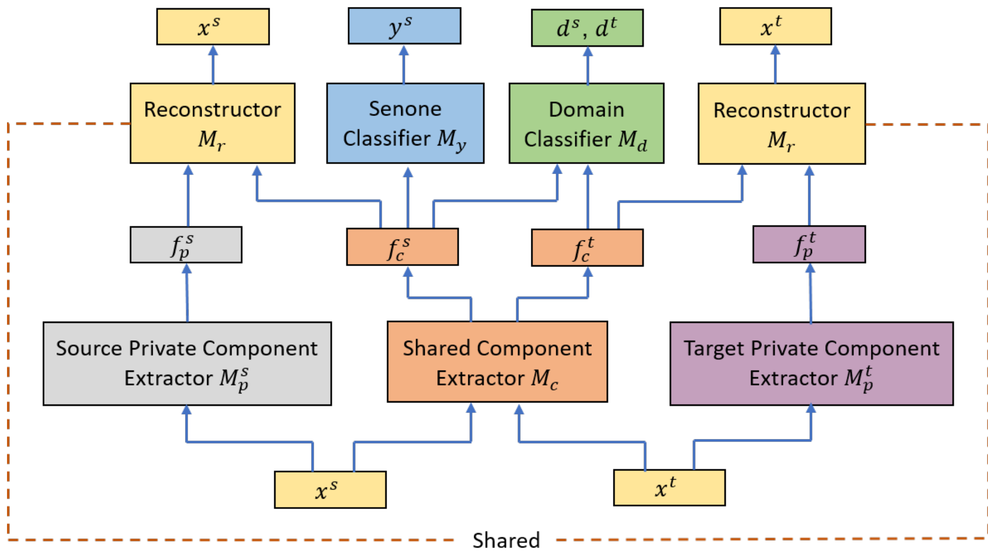

Fig. 1. The architecture of domain separation networks.

$f_c^s \in \mathcal{R}^K$. f_c^s is then mapped to the senone label posteriors by a senone classifier M_y with parameters θ_y as follows.

$$M_y(f_c^s) = M_y(M_c(x_i^s)) = p(\hat{y}_n^s = q | x_i^s; \theta_c, \theta_y) \quad (1)$$

where $\hat{y}_i^s$ denotes the predicted senone label for source frame x_i^s and $q \in \mathcal{Q}$.

The domain classifier DNN M_d with parameters θ_d takes the shared component from source domain f_c^s or target domain f_c^t as the input to predict the two-dimensional domain label posteriors as follows (the 1st and 2nd output units stand for the source and target domains respectively).

$$M_d(M_c(x_i^s)) = p(\hat{d}_i^s = a | x_i^s; \theta_c, \theta_d), \ a \in \{1, 2\} \quad (2)$$

$$M_d(M_c(x_j^t)) = p(\hat{d}_j^t = a | x_j^t; \theta_c, \theta_d), \ a \in \{1, 2\} \quad (3)$$

where $\hat{d}_i^s$ and $\hat{d}_j^t$ denote the predicted domain labels for the source frame x_i^s and the target frame x_j^t respectively.

In order to adapt the source domain acoustic model (i.e., M_c and M_y) to the *unlabeled* data from target domain, we want to make the distribution of the source domain shared component $P(f_c^s) = P(M_c(x^s))$ as close to that of the target domain $P(f_c^t) = P(M_c(x^t))$ as possible. In other words, we want to make the shared component domain-invariant. This can be realized by adversarial training, in which we adjust the parameters θ_c of shared component extractor to *maximize* the loss of the domain classifier $\mathcal{L}_{\text{domain}}^c(\theta_c)$ below while adjusting the parameters θ_d to *minimize* the loss of the domain classifier $\mathcal{L}_{\text{domain}}^d(\theta_d)$ below.

$$\begin{aligned}\mathcal{L}_{\text{domain}}^d(\theta_d) = &-\sum_i^{N_s} \log p(\hat{d}_i^s = 1 | x_i^s; \theta_d) \\ &- \sum_j^{N_t} \log p(\hat{d}_j^t = 2 | x_j^t; \theta_d) \end{aligned} \quad (4)$$

$$\begin{aligned}\mathcal{L}_{\text{domain}}^c(\theta_c) = &-\sum_i^{N_s} \log p(\hat{d}_i^s = 1 | x_i^s; \theta_c) \\ &- \sum_j^{N_t} \log p(\hat{d}_j^t = 2 | x_j^t; \theta_c) \end{aligned} \quad (5)$$

This minimax competition will first increase the capability of both the shared component extractor and the domain classifier and will eventually converge to the point where the shared component extractor generates extremely confusing representations that domain classifier is unable to distinguish (i.e., domain-invariant).

Simultaneously, we minimize the loss of the senone classifier below to ensure the domain-invariant shared component f_c^s is also discriminative to senones.

$$\mathcal{L}_{\text{senone}}(\theta_c, \theta_y) = -\sum_i^{N_s} \log p(y_i^s | x_i^s; \theta_y, \theta_c) \quad (6)$$

Since the adversarial training of the domain classifier M_d and shared component extractor M_c has made the distribution of the target domain shared-component f_c^t as close to that of f_c^s as possible, the f_c^t is also senone-discriminative and will lead to minimized senone classification error given optimized

M_y. Because of the domain-invariant property, good adaptation performance can be achieved when the target domain data goes through the network.

2.3. Private Components Extraction

To further increase the degree of domain-invariance of the shared components, we explicitly model the private component that is unique to each domain by a private component extractor DNN M_p parameterized by θ_p. M_p^s and M_p^t map the source frame x^s and the target frame x^t to hidden representations $f_p^s = M_p^s(x^s)$ and $f_p^t = M_p^t(x^t)$ which are the private components of the source and target domains respectively. The private component for each domain is trained to be orthogonal to the shared component by minimizing the difference loss below.

$$\mathcal{L}_{\text{diff}}(\theta_c, \theta_p^s, \theta_p^t) = \left|\left|\sum_i^{N_s} M_c(x_i^s) M_p^s(x_i^s)^\top\right|\right|_F^2 + \left|\left|\sum_j^{N_t} M_c(x_j^t) M_p^t(x_j^t)^\top\right|\right|_F^2 \tag{7}$$

where $||\cdot||_F^2$ is the squared Frobenius norm. All the vectors are assumed to be column-wise.

As a regularization term, the predicted shared and private components are then concatenated and fed into a reconstructor DNN M_r with parameters θ_r to recover the input speech frames x^s and x^t from both source and target domains respectively. The reconstructor is trained to minimize the mean square error based reconstruction loss as follows:

$$\mathcal{L}_{\text{recon}}(\theta_c, \theta_p^s, \theta_p^t, \theta_r) = \sum_i^{N_s} ||\hat{x}_i^s - x_i^s||_2^2 + \sum_j^{N_t} ||\hat{x}_j^t - x_j^t||_2^2 \tag{8}$$

$$\hat{x}_i^s = M_r([M_c(x_i^s), M_p^s(x_i^s)]) \tag{9}$$

$$\hat{x}_j^t = M_r([M_c(x_j^t), M_p^t(x_j^t)]) \tag{10}$$

where $[\cdot,\cdot]$ denotes concatenation of two vectors.

The total loss of DSN is formulated as follows and is jointly optimized with respect to the parameters.

$$\mathcal{L}_{\text{total}}(\theta_y, \theta_c, \theta_d, \theta_p^s, \theta_p^t, \theta_r) = \mathcal{L}_{\text{senone}}(\theta_c, \theta_y) + \mathcal{L}_{\text{domain}}^d(\theta_d) - \alpha \mathcal{L}_{\text{domain}}^c(\theta_c) + \beta \mathcal{L}_{\text{diff}}(\theta_c, \theta_p^s, \theta_p^t) + \gamma \mathcal{L}_{\text{recon}}(\theta_c, \theta_p^s, \theta_p^t, \theta_r) \tag{11}$$

$$\min_{\theta_y, \theta_c, \theta_d, \theta_p^s, \theta_p^t, \theta_r} \mathcal{L}_{\text{total}}(\theta_y, \theta_c, \theta_d, \theta_p^s, \theta_p^t, \theta_r) \tag{12}$$

All the parameters of DSN are jointly optimized through backprogation with stochastic gradient descent (SGD) as follows:

$$\theta_c \leftarrow \theta_c - \mu \left[\frac{\partial \mathcal{L}_{\text{senone}}}{\partial \theta_c} - \alpha \frac{\partial \mathcal{L}_{\text{domain}}^c}{\partial \theta_c} + \beta \frac{\partial \mathcal{L}_{\text{diff}}}{\partial \theta_c} + \gamma \frac{\partial \mathcal{L}_{\text{recon}}}{\partial \theta_c}\right] \tag{13}$$

$$\theta_d \leftarrow \theta_d - \mu \frac{\partial \mathcal{L}_{\text{domain}}^d}{\partial \theta_d}, \quad \theta_y \leftarrow \theta_y - \mu \frac{\partial \mathcal{L}_{\text{senone}}}{\partial \theta_y} \tag{14}$$

$$\theta_p^s \leftarrow \theta_p^s - \mu \left[\beta \frac{\partial \mathcal{L}_{\text{diff}}}{\partial \theta_p^s} + \gamma \frac{\partial \mathcal{L}_{\text{recon}}}{\partial \theta_p^s}\right] \tag{15}$$

$$\theta_p^t \leftarrow \theta_p^t - \mu \left[\beta \frac{\partial \mathcal{L}_{\text{diff}}}{\partial \theta_p^t} + \gamma \frac{\partial \mathcal{L}_{\text{recon}}}{\partial \theta_p^t}\right] \tag{16}$$

$$\theta_r \leftarrow \theta_r - \mu \frac{\partial \mathcal{L}_{\text{recon}}}{\partial \theta_r} \tag{17}$$

Note that the negative coefficient $-\alpha$ in Eq. (13) induces reversed gradient that maximizes the domain classification loss in Eq. (5) and makes the shared components domain-invariant. Without the reversal gradient, SGD would make representations different across domains in order to minimize Eq. (4). For easy implementation, GRL is introduced in [24], which acts as an identity transform in the forward pass and multiplies the gradient by $-\alpha$ during the backward pass.

The optimized shared component extractor M_c and senone classifier M_y form the adapted acoustic model for robust speech recognition.

3. EXPERIMENTS

In this work, we perform the *pure* unsupervised environment adaptation of the DNN-HMM acoustic model with domain separation networks for robust speech recognition on CHiME-3 dataset.

3.1. CHiME-3 Dataset

The CHiME-3 dataset is released with the 3rd CHiME speech Separation and Recognition Challenge [29], which incorporates the Wall Street Journal corpus sentences spoken in challenging noisy environments, recorded using a 6-channel tablet based microphone array. CHiME-3 dataset consists of both real and simulated data. The real speech data was recorded in four real noisy environments (on buses (BUS), in cafés (CAF), in pedestrian areas (PED), and at street junctions (STR)). To generate the simulated data, the clean speech is first convoluted with the estimated impulse response of the environment and then mixed with the background noise separately recorded in that environment [30]. The noisy training data consists of 1600 real noisy utterances from 4 speakers, and 7138 simulated noisy utterances from 83 speakers in the WSJ0 SI-84 training set recorded in 4 noisy environments. There are 3280 utterances in the development set including 410 real and 410 simulated utterances for each of the 4 environments. There are 2640 utterances in the test set including

330 real and 330 simulated utterances for each of the 4 environments. The speakers in training set, development set and the test set are mutually different (i.e., 12 different speakers in the CHiME-3 dataset). The training, development and test data sets are all recorded in 6 different channels.

8738 clean utterances corresponding to the 8738 noisy training utterances in the CHiME-3 dataset are selected from the WSJ0 SI-85 training set to form the clean training data in our experiments. WSJ 5K word 3-gram language model is used for decoding.

3.2. Baseline System

In the baseline system, we first train a DNN-HMM acoustic model with clean speech and then adapt the clean acoustic model to noisy data using GRL unsupervised adaptation in [24]. Hence, the source domain is with clean speech while the target domain is with noisy speech.

The 29-dimensional log Mel filterbank features together with 1st and 2nd order delta features (totally 87-dimensional) for both the clean and noisy utterances are extracted by following the process in [31]. Each frame is spliced together with 5 left and 5 right context frames to form a 957-dimensional feature. The spliced features are fed as the input of the feed-forward DNN after global mean and variance normalization. The DNN has 7 hidden layers with 2048 hidden units for each layer. The output layer of the DNN has 3012 output units corresponding to 3012 senone labels. Senone-level forced alignment of the clean data is generated using a GMM-HMM system. The DNN is first trained with 8738 clean training utterances in CHiME-3 and the alignment to minimize the cross entropy loss and then tested with simulation and real development data of CHiME-3.

The DNN well-trained with clean data is then adapted to the 8738 noisy utterances from Channel 5 using GRL method. No senone alignment of the noisy adaptation data is used for the unsupervised adaptation. The feature extractor is initialized with the first 4 hidden layers of the clean DNN and the senone classifier is initialized with the last 3 hidden layers plus the output layers of the clean DNN. The domain classifier is a feedforward DNN with two hidden layers and each hidden layer has 512 hidden units. The output layer of the domain classifier has 2 output units representing source and target domains. The 2048 hidden units of the 4^{th} hidden layer of the DNN acoustic model is fed as the input to the domain classifier. A GRL is inserted in between the deep representation and the domain classifier for easy implementation. The GRL adapted system is tested on real and simulation noisy development data in CHiME-3 dataset.

3.3. Domain Separation Networks for Unsupervised Adaptation

We adapt the clean DNN acoustic model trained in Section 3.2 to the 8738 noisy utterances using DSN. No senone alignment of the noisy adaptation data is used for the unsupervised adaptation.

The DSN is implemented with CNTK 2.0 Toolkit [32]. The shared component extractor M_c is initialized with the first N_h hidden layers of the clean DNN and the senone classifier M_y is initialized with the last $(7 - N_h)$ hidden layers plus the output layer of the clean DNN. N_h indicates the position of shared component in the DNN acoustic model and ranges from 3 to 7 in our experiments. The domain classifier M_d of the DSN has exactly the same architecture as that of the GRL.

The private component extractors M_p^s and M_p^t for the clean and noisy domains are both feedforward DNNs with 3 hidden layers and each hidden layer has 512 hidden units. The output layers of both M_p^s and M_p^t have 2048 output units. The reconstructor M_r is a feedforward DNN with 3 hidden layers and each hidden layer has 512 hidden units. The output layer of the M_r has 957 output units with no non-linear activation functions to reconstruct the spliced input features.

The activation functions for the hidden units of M_c is sigmoid. The activation functions for hidden units of M_p^s, M_p^t, M_d and M_r are rectified linear units (ReLU). The activation functions for the output units of M_c and M_d are softmax. The activation functions for the output units of M_p^s, M_p^t are sigmoid. All the sub-networks except for M_y and M_c are randomly initialized. The learning rate is fixed at $5 \times 10^{\;5}$ throughout the experiments. The adapted DSN is tested on real and simulation development data in CHiME-3 Dataset.

System	Data	BUS	CAF	PED	STR	Avg.
Clean	Real	36.25	31.78	22.76	27.18	29.44
	Simu	26.89	37.74	24.38	26.76	28.94
GRL	Real	35.93	28.24	19.58	25.16	27.16
	Simu	26.14	34.68	22.01	25.83	27.16
DSN	Real	32.62	23.48	17.29	23.46	**24.15**
	Simu	23.38	30.39	19.51	22.01	**23.82**

Table 1. The WER (%) performance of unadapted acoustic model, GRL and DSN adapted DNN acoustic models for robust ASR on real and simulated development set of CHiME-3.

3.4. Result Analysis

Table 1 shows the WER performance of clean, GRL adapted and DSN adapted DNN acoustic models for ASR. The clean DNN achieves 29.44% and 28.25% WERs on the real and simulated development data respectively. The GRL adapted acoustic model achieves 27.16% and 27.16% WERs on the real and simulated development data. The best WER performance for DSN adapted acoustic model are 24.15% and 23.82% on real and simulated development data, which achieve 11.08% and 12.30% relative improvement over the GRL baseline system and achieve 17.97 % and 17.69% relative improvement over the unadapted acoustic model. The

N_h	Reversal Gradient Coefficient α									
	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	Avg.
3	27.2	26.24	25.76	26.51	26.12	26.92	26.65	26.91	27.41	26.64
4	26.56	26.08	25.75	25.99	25.88	26.76	27.0	27.13	27.74	26.54
5	26.53	25.9	26.07	25.88	25.72	26.17	27.36	26.67	27.37	26.41
6	25.77	25.17	25.06	24.94	24.6	25.19	25.53	25.42	25.93	25.29
7	25.99	25.5	24.73	24.43	25.08	24.53	25.07	24.15	24.29	**24.86**

Table 2. The ASR WERs (%) for the DSN adapted acoustic models with respect to N_h reversal gradient coefficient α on the real development set of CHiME-3.

best WERs are achieved when $N_h = 7$ and $\alpha = 8.0$. By comparing the GRL and DSN performance at $N_h = 4$, we observe that the introduction of private components and reconstructor lead to 5.1% relative improvements in WER.

We investigate the impact of shared component position N_h and the reversal gradient coefficient α on the WER performance as in Table 2. We observe that the WER decreases with the growth of N_h, which is reasonable as the higher hidden representation of a well-trained DNN acoustic model is inherently more senone-discriminative and domain-invariant than the lower layers and can serve as a better initialization for the DSN unsupervised adaptation.

4. CONCLUSIONS

In this work, we investigate the domain adaptation of the DNN acoustic model by using domain separation networks. Different from the conventional supervised, semi-supervised and T/S adaptation approaches, DSN is capable of adapting the acoustic model to the adaptation data without any exposure to its transcription, decoded lattices or unlabeled parallel data from the source domain. The shared component between source and target domains extracted by DSN through adversarial multi-task training is both domain-invariant and senone-discriminative. The extraction of private component that is unique to each domain significantly improves the degree of domain-invariance and the ASR performance.

When evaluated on the CHiME-3 dataset for environment adaption task, the DSN achieves 11.08% and 17.97% relative WER improvement over the GRL baseline system and the unadapted acoustic model. The WER decreases when higher hidden representations of the DNN acoustic model are used as the initial shared component. The WER first decreases and then increases with the growth of the reversal gradient coefficient.

5. REFERENCES

[1] Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. INTERSPEECH*, 2011.

[2] Tara N Sainath, Brian Kingsbury, Bhuvana Ramabhadran, Petr Fousek, Petr Novak, and Abdel-rahman Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2011, pp. 30–35.

[3] Navdeep Jaitly, Patrick Nguyen, Andrew Senior, and Vincent Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. INTERSPEECH*, 2012.

[4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, et al., "Recent advances in deep learning for speech research at microsoft," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8604–8608.

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

[7] 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, April 2014.

[8] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition: A Bridge to Practical Applications*, Academic Press, 2015.

[9] 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 *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 7893–7897.

[10] Zhen Huang, Sabato Marco Siniscalchi, I-Fan Chen, Jinyu Li, Jiadong Wu, and Chin-Hui Lee, "Maximum a posteriori adaptation of network parameters in deep models," in *Interspeech*, 2015.

[11] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 7947–7951.

[12] Zhen Huang, Jinyu Li, Sabato Marco Siniscalchi, I-Fan Chen, Ji Wu, and Chin-Hui Lee, "Rapid adaptation for deep neural networks through multi-task learning.," in *Interspeech*, 2015, pp. 3625–3629.

[13] Roberto Gemello, Franco Mana, Stefano Scanzio, Pietro Laface, and Renato De Mori, "Linear hidden transformations for adaptation of hybrid ann/hmm models," *Speech Communication*, vol. 49, no. 10, pp. 827 – 835, 2007, Intrinsic Speech Variations.

[14] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, Dec 2011, pp. 24–29.

[15] Jian Xue, Jinyu Li, and Yifan Gong, "Restructuring of deep neural network acoustic models with singular value decomposition.," in *Interspeech*, 2013, pp. 2365–2369.

[16] 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 *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6359–6363.

[17] Y. Zhao, J. Li, and Y. Gong, "Low-rank plus diagonal adaptation for deep neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5005–5009.

[18] 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, Aug 2016.

[19] P. Swietojanski and S. Renals, "Differentiable pooling for unsupervised speaker adaptation," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4305–4309.

[20] Z. Q. Wang and D. Wang, "Unsupervised speaker adaptation of batch normalized acoustic models for robust asr," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4890–4894.

[21] Jinyu Li, Michael L Seltzer, Xi Wang, Rui Zhao, and Yifan Gong, "Large-scale domain adaptation via teacher-student learning," in *INTERSPEECH*, 2017.

[22] Jinyu Li, Rui Zhao, Jui-Ting Huang, and Yifan Gong, "Learning small-size DNN with output-distribution-based criteria.," in *INTERSPEECH*, 2014, pp. 1910–1914.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672–2680. Curran Associates, Inc., 2014.

[24] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning*, Francis Bach and David Blei, Eds., Lille, France, 07–09 Jul 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 1180–1189, PMLR.

[25] Sining Sun, Binbin Zhang, Lei Xie, and Yanning Zhang, "An unsupervised deep domain adaptation approach for robust speech recognition," *Neurocomputing*, vol. 257, pp. 79 – 87, 2017, Machine Learning and Signal Processing for Big Multimedia Analysis.

[26] Yusuke Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition.," in *INTERSPEECH*, 2016, pp. 2369–2372.

[27] Dmitriy Serdyuk, Kartik Audhkhasi, Philémon Brakel, Bhuvana Ramabhadran, Samuel Thomas, and Yoshua Bengio, "Invariant representations for noisy speech recognition," in *NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*, 2016.

[28] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan, "Domain separation networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 343–351. Curran Associates, Inc., 2016.

[29] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 504–511.

[30] T. Hori, Z. Chen, H. Erdogan, J. R. Hershey, J. Le Roux, V. Mitra, and S. Watanabe, "The merl/sri system for the 3rd chime challenge using beamforming, robust feature extraction, and advanced speech recognition," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 475–481.

[31] Jinyu Li, Dong Yu, Jui-Ting Huang, and Yifan Gong, "Improving wideband speech recognition using mixed-bandwidth training data in cd-dnn-hmm," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 131–136.

[32] Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Zhiheng Huang, Brian Guenter, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Huaming Wang, et al., "An introduction to computational networks and the computational network toolkit," *Microsoft Technical Report MSR-TR-2014–112*, 2014.