CROSS-DOMAIN SPEECH RECOGNITION USING NONPARALLEL CORPORA WITH CYCLE-CONSISTENT ADVERSARIAL NETWORKS

Masato Mimura, Shinsuke Sakai, Tatsuya Kawahara

Kyoto University, School of Informatics,
Sakyo-ku, Kyoto 606-8501, Japan

ABSTRACT

Automatic speech recognition (ASR) systems often does not perform well when it is used in a different acoustic domain from the training time, such as utterances spoken in noisy environments or in different speaking styles. We propose a novel approach to cross-domain speech recognition based on acoustic feature mappings provided by a deep neural network, which is trained using nonparallel speech corpora from two different domains and using no phone labels. For training a target domain acoustic model, we generate "fake" target speech features from the labeleld source domain features using a mapping G_f. We can also generate "fake" source features for testing from the target features using the backward mapping G_b which has been learned simultaneously with G_f. The mappings G_f and G_b are trained as adversarial networks using a conventional adversarial loss and a cycle-consistency loss criterion that encourages the backward mapping to bring the translated feature back to the original as much as possible such that $G_b(G_f(x)) \approx$ x. In a highly challenging task of model adaptation only using domain speech features, our method achieved up to 16 % relative improvements in WER in the evaluation using the CHiME3 real test data. The backward mapping was also confirmed to be effective with a speaking style adaptation task.

Index Terms— acoustic model adaptation, unsupervised training, speech enhancement, generative adversarial networks, cycle consistency loss

1. INTRODUCTION

Deep learning-based hybrid acoustic models have drastically improved the performance of automatic speech recognition (ASR) [1]. It was recently reported that even a human-level recognition performance can be achievable when they are coupled with bidirectional LSTMs and very deep convolutional networks with residual connections [2][3]. However, these excellent results are only guaranteed in the fortunate cases where a large amount of training data matched to test data is available. This is why adaptation of acoustic models trained with a speech corpus in some domain to a new target domain still remains one of the most actively investigated research topics. Making manual transcriptions of speech is costly and sometimes raises privacy concerns. Therefore, if an effective way of unsupervised adaptation for acoustic models to a new domain were established, which only requires acoustic data in the new domain, it would make a great impact on the applicability os ASR in a variety of real life situations. With an well-established unsupervised adaptation method, for example, ASR products such as intelligent speakers and conversational robots can continue to improve their performances even after shipping using acoustic signals recorded in users' own operating environments.

There are a number of potential needs for domain mapping in ASR. One example is a noisy speech recognition task. ASR in noisy conditions was conventionally addressed by multi-condition training of acoustic models using simulated noisy corpora, which is artificially generated by convolving room impulse responces and adding noise to clean corpora. Another approach is acoustic feature enhancement in frontend using denoising autoencoders [4][5][6][7][8], where mappings from noisy features to enhanced features are learned using paired examples between clean and simulated noisy corpora. The problem with these methods is that the mixing process of speech and noise in real noisy conditions may have a highly nonlinear nature, and the simulated data generated using linear transformations described above has a very different characteristics from real noisy data, which can limit the performances of the methods based on simulated data. In fact, discrepancies between the recognition performances for simulated and real noisy test data have been reported in the literature [9][10]. Generating simulated data also requires a considerable cost for carefully recording room impulse responses and noise backgrounds in the target conditions which can sometimes lead to an infringement of privacy. On the other hand, annotating real noisy corpus is highly expensive, and paired examples between clean and real noisy data can not be generated in principle, as in most of other adaptation problems such as speaker or speaking style adaptation.

In this paper, we propose a novel approach to cross-domain speech recognition requiring no corresponding examples between source and target domains and no labels for the target domain corpus. In the proposed method, acoustic models are trained using "fake" target domain features translated from source domain features using a complex nonlinear mapping provided by a variant of generative adversarial networks (GANs) [11]. This network is trained without supervision using source and target domain corpora. The method does not require initial speech recognition results which are crucial in typical retraining approaches for unsupervised acoustic model adaptation. Two notable design choices are incorporated into the network including the cycle consistency loss criterion [12] in the training to guide the network to retain useful information for speech recognition in the translated features. In the experiment, we also demonstrate that "fake" source domain features generated by an inverse mapping from target to source domain can contribute to improve the speech recognition performance.

2. ACOUSTIC MODEL ADAPTATION WITH GENERATIVE ADVERSARIAL NETWORKS (GANS)

The problem we address in this paper is summarized as follows. We have two kinds of data which belong to two distinct domains, namely, "source" and "target" domains. For the source domain data,

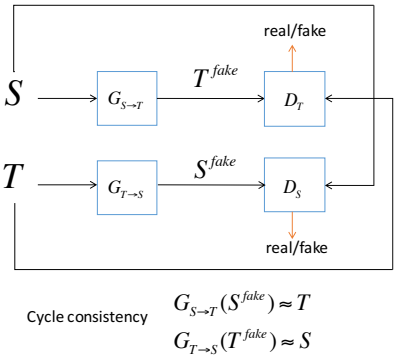


Fig. 1. Original cycle GAN architecture for image-to-image translation

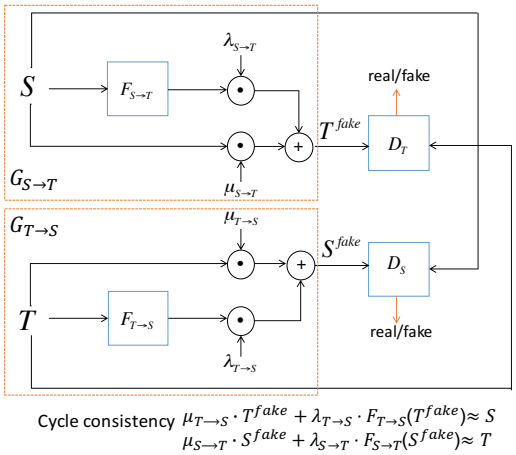


Fig. 2. Proposed cycle GAN-based architecture for acoustic feature transformation

we have manual transcriptions which can be used for training acoustic models, but we do not for the target domain. We also do not have paired examples between these two domains. Our objective is to improve ASR performance for test data which belongs to the target domain under these constraints.

This is a very common situation we encounter frequently. In the following parts of this section, we particularly take an "noise-robust ASR" example where the source domain is clean speech and the target domain is noisy speech in explaining our proposed method for understandability. Our approach is to train acoustic models using "fake" noisy features translated from clean features for which we have transcriptions. The most important issue here is how to generate a realistic "noisy" version of clean features in the absence of paired examples between two domains. This is a much more difficult setting than in conventional denoising autoencoder approaches ([13][14][4][5][6][7][8]).

We propose to use the concept of generative adversarial networks (GANs) [11] and cycle consistency adversarial networks (cycle GANs) [12], which recently yielded impressive results in the image processing area, for generating translated data. Moreover, we introduce an enhancement in the architecture of GANs in order to achieve desirable characteristics for ASR in translated features.

2.1. Generative adversarial networks for domain translation

GAN is a framework for estimating generative models via an adversarial process, in which two models G and D are simultaneously trained. G is a generative model that captures the data distribution, and D is a discriminative model that estimates the probability that a sample came from the training data rather than G. While the original GANs generate data from latent variables, we consider here a network $G_{S\to T}$ which transforms speech from a domain S (e.g. clean speech) to some other domain T (e.g. noisy speech). This generator network $G_{S\to T}$ is trained such that the distribution $p_{tgt}(\boldsymbol{t})$ of speech features t in T is indistinguishable from the distribution of "fake" target domain speech features $G_{S\to T}(\boldsymbol{s})$, where s is subject to the source domain distribution $p_{src}(\boldsymbol{s})$. In the GAN framework, this is achieved by optimizing the following minimax criterion:

$$G^*_{S\to T} = \arg\min_{G_{S\to T}} \max_{D_T} \mathcal{L}_{\mathrm{GAN}}(G_{S\to T}, D_T), \tag{1}$$

where an *adversarial objective* $\mathcal{L}_{\mathrm{GAN}}(G_{S\to T}, D_T)$ is defined as:

$$\mathcal{L}_{\mathrm{GAN}}(G_{S\to T}, D_T) = \mathbb{E}_{\boldsymbol{t}\sim p_{tgt}(\boldsymbol{t})}[\log D_T(\boldsymbol{t})] + \mathbb{E}_{\boldsymbol{s}\sim p_{src}(\boldsymbol{s})}[\log(1 - D_T(G_{S\to T}(\boldsymbol{s})))], \tag{2}$$

and D_T is a discriminator network with its output representing the probability that the input comes from T. By the optimization criterion (1), D_T is trained to maximize $D_T(t)$ for $t \in T$ and minimize $D_T(G_{S\to T}(s))$ for fake data $G_{S\to T}(s)$ generated from $s \in S$, while $G_{S\to T}$ is trained to maximize $D_T(G_{S\to T}(s))$.

2.2. Cycle consistency loss

GANs may provide us with a powerful way to translate data across domains without parallel examples, but they are too underconstrained for keeping discriminative information required in ASR. For example, information such as formant trajectories needs to be preserved after domain translation, but the adversarial loss (2) may not enough for it.

Therefore, we choose to train not only a source to target mapping $G_{S\to T}$, but also a target to source mapping $G_{T\to S}$ in a consistent way by introducing a constraint that these mappings should be "cycle consistent" [12], namely, the data translated by $G_{S\to T}$ is mapped back by $G_{T\to S}$ to a source domain feature as close as possible to the original feature, and vice versa. Thus, we expect that the information for reconstructing the input data in either domain is kept in the domain-transformed data. We can incentivize this behavior using a *cycle consistency loss*:

$$\mathcal{L}_{\mathrm{cyc}}(G_{S\to T}, G_{T\to S}) = \mathbb{E}_{\boldsymbol{s}\sim p_{src}(\boldsymbol{s})}[\|G_{T\to S}(G_{S\to T}(\boldsymbol{s})) - \boldsymbol{s}\|_1] + \mathbb{E}_{\boldsymbol{t}\sim p_{tgt}(\boldsymbol{t})}[\|G_{S\to T}(G_{T\to S}(\boldsymbol{t})) - \boldsymbol{t}\|_1]. \tag{3}$$

By paring this cycle consistency loss with the standard adversarial loss, we encourage $G_{S\to T}(G_{T\to S}(\boldsymbol{t})) \approx \boldsymbol{t}$ and $G_{T\to S}(G_{S\to T}(\boldsymbol{s})) \approx \boldsymbol{s}$. The structure of GANs with the cycle consistency loss is depicted in Fig. 1.

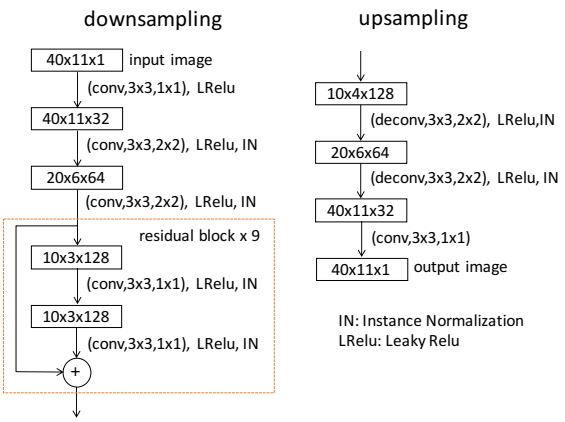


Fig. 3. Network architecture for $F_{S\to T}$ and $F_{T\to S}$

The full objective for training $G_{S\to T}$, $G_{T\to S}$, D_S and D_T is:

$$\begin{aligned}\mathcal{L}(G_{S\to T}, G_{T\to S}, D_S, D_T) &= \mathcal{L}_{\text{GAN}}(G_{S\to T}, D_T) \\ &+\mathcal{L}_{\text{GAN}}(G_{T\to S}, D_S) \\ &+\alpha\mathcal{L}_{\text{cyc}}(G_{S\to T}, G_{T\to S}). \end{aligned} \quad (4)$$

2.3. Modified network architecture for acoustic feature transformation

We also consider to add more structure to our network architecture for further improving the quality of the transformed data.

We build two distinct paths in each of $G_{S\to T}$ and $G_{T\to S}$, which will be summed together to generate transformed data (Fig. 2). The first path is basically an identity mapping, which explicitly guarantees that the detailed structure in the input data will be preserved after the domain translation. The second has a generative network $F_{S\to T}$ or $F_{T\to S}$ which has a capacity to learn a complex nonlinear mapping like generators in standard GANs. With the existence of the first identity mapping path, the generative networks in the second path are enforced to learn devotedly the "difference" between two domains. Submapping-wise scaling factors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are introduced for adjusting the intensity of each component before summation.

Now the mappings $G_{S\to T}$ and $G_{T\to S}$ are reformulated as:

$$G_{S\to T}(\boldsymbol{s}) = \boldsymbol{\lambda}_{S\to T} \odot F_{S\to T}(\boldsymbol{s}) + \boldsymbol{\mu}_{S\to T} \odot \boldsymbol{s}, \quad (5)$$

$$G_{T\to S}(\boldsymbol{t}) = \boldsymbol{\lambda}_{T\to S} \odot F_{T\to S}(\boldsymbol{t}) + \boldsymbol{\mu}_{T\to S} \odot \boldsymbol{t}, \quad (6)$$

where $\odot$ means element-wise multiplication.

3. IMPLEMENTATION

3.1. Network architecture

Following the description in the implementation part of [12], the architecture for the generative networks $F_{S\to T}$ and $F_{T\to S}$ (Fig. 3) is adapted from Johnson et al. [15]. A feature map of size 40x11x1 consisting of eleven frames of 40-channel log Mel-scale filterbank (lmfb) outputs is used as input to the networks. Each generative network has two subnetworks for downsampling and upsamping. The downsampling subnetwork consists of three convolutional layers and nine residual blocks [16]. Each residual network is composed of two stride-1 convolutions and a residual connection which bypasses them. On the other hand, the upsampling part has two deconvolutional layers followed by one stride-1 convolution. The filter and stride size in each convolution layer are depicted in Fig. 3. For example, (conv,3x3,1x1) means a convolution layer with a filter of size 3x3 and a stride of size 1x1. Note that we chose a smaller size for convolutional filters than in [12], because the size of our input images composed of lmfb features is much smaller than those in typical image processing applications. We used instance normalization [17] in layers specified in Fig. 3 before applying nonlinearities. Leaky ReLU nonlinearities with a slope of 0.2 are used in all layers with the exception of the output layer, which uses an identity function.

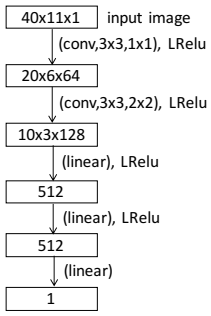


Fig. 4. Network architecture for D_S and D_T

The architecture for the discriminators D_S and D_T is depicted in Fig. 4. The network has two convolutional layers, followed by three fully-connected layers. While Leaky ReLU nonlinearities are also used in the discriminators, we did not apply instance normalization here as suggested in [18].

3.2. Training procedure

We used Wasserstein GANs (WGANs) [19] for building our generative networks instead of standard GANs to stabilize our model training procedure and avoid training problems inherent in GANs such as model collapse. Gradient penalties recently proposed in [18] are also used in training WGANs instead of the weight clipping technique [19] to enforce the Lipshitz constraint. Accordingly, the full objective (4) is modified as:

$$\begin{aligned}\mathcal{L}(G_{S\to T}, G_{T\to S}, D_S, D_T) &= \mathcal{L}_{\text{WGAN}}(G_{S\to T}, D_T) \\ &+\mathcal{L}_{\text{WGAN}}(G_{T\to S}, D_S) \\ &+\alpha\mathcal{L}_{\text{cyc}}(G_{S\to T}, G_{T\to S}), \end{aligned} \quad (7)$$

where $\mathcal{L}_{\text{WGAN}}(G_{S\to T}, D_T)$ is:

$$\begin{aligned}\mathcal{L}_{\text{WGAN}}(G_{S\to T}, D_T) &= \mathbb{E}_{\boldsymbol{t}\sim p_{tgt}(\boldsymbol{t})}[D_T(\boldsymbol{t})] \\ &-\mathbb{E}_{\boldsymbol{s}\sim p_{src}(\boldsymbol{s})}[D_T(G_{S\to T}(\boldsymbol{s}))] \\ &-\beta\mathbb{E}_{\hat{\boldsymbol{t}}\sim p_{tgt}(\hat{\boldsymbol{t}})}[(\|\Delta_{\hat{\boldsymbol{t}}} D_T(\hat{\boldsymbol{t}})\|_2 - 1)^2] \\ &-\beta\mathbb{E}_{\hat{\boldsymbol{s}}\sim p_{src}(\hat{\boldsymbol{s}})}[(\|\Delta_{\hat{\boldsymbol{s}}} D_S(\hat{\boldsymbol{s}})\|_2 - 1)^2], \end{aligned} \quad (8)$$

where, for example, $\mathbb{E}_{\hat{\boldsymbol{t}}\sim p_{tgt}(\hat{\boldsymbol{t}})}[(\|\Delta_{\hat{\boldsymbol{t}}} D_T(\hat{\boldsymbol{t}})\|_2 - 1)^2]$ is the gradient penalty for critic[1] D_T. We defined $\hat{\boldsymbol{t}}$ as $\hat{\boldsymbol{t}} = a\boldsymbol{t} + (1-a)G_{S\to T}(\boldsymbol{s})$ using a random variable $a \sim U(0,1)$, $\boldsymbol{s} \sim p_{src}(\boldsymbol{s})$ and $\boldsymbol{t} \sim p_{tgt}(\boldsymbol{t})$, as suggested in [18]. More detailed explanations on WGANs and

[1]When we use WGANs, we call the networks D_S and D_T "critics", because they actually don't discriminate anything.

gradient penalties, which are out of scope of this paper, are found in [19] and [18].

Following the recipes in [19] and [18], we update critics D_S and D_T n_{critic} times before updating $G_{S\to T}$ and $G_{T\to S}$ for each minibatch iteration. Note that while the critics are trained to minimize $-\mathcal{L}(G_{S\to T}, G_{T\to S}, D_S, D_T)$, the generators are trained to minimize $\mathcal{L}(G_{S\to T}, G_{T\to S}, D_S, D_T)$. We set n_{critic} to be 4 in all our experiments. We used the Adam optimizer [20] with a minibatch size of 256 for each of source and target domain data. All network parameters are initialized with random values with the exception of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, which were initialized with 1, and trained with a learning rate of 0.0001 for 20 epochs. We set α in (7) to be 10, and β in (8) to be 10.

4. EXPERIMENTAL EVALUATION

We evaluated the proposed methods through two domain adaptation tasks, namely, noise-robust speech recognition and speaking style adaptation.

4.1. Noise-robust speech recognition

First, we evaluate the proposed method on a noisy speech recognition task, specifically the 1-channel track of the fourth CHiME Challenge [10], where the source domain is clean speech and target domain is noisy speech. The "source" clean training set consists of 7,138 utterances from WSJ0 corpus. The "target" noisy training set consists of 1,600 real noisy utterances and 7,138 simulated noisy utterances generated by artificially mixing the clean training set with noise backgrounds. There are four different types of noisy environments, namely, bus, street, cafe, and pedestrian area [10]. A 440-dimensional feature vector consisting of 11 frames of 40-channel lmfb outputs is used as input to domain translation networks, as described in Section 3.1. The acoustic feature vectors in each set are normalized to have a zero mean and unit variance, and shuffled at frame level. For simplifying our training procedure, we used the same amount of shuffled data for both domains. We trained a CNN-HMM acoustic model [21] using the clean training set described above. It has two stride-1 convolutional layers, three fully-connected layers with 2k rectified linear units (ReLUs) [22] and a softmax output layer with 2k nodes. Each convolutional layer is followed by a stride-2 max pooling layer. The first convolutional layer has 180 filters of size 5x11, and the second one has 180 filters of size 5x1. The same 440-dimensional lmfb-based feature vector is used as input to the acoustic model as used for the domain translation networks, and we can directly input the translated features to the acoustic model. For decoding, we used the Kaldi WFST decoder [23]. The language model is the standard WSJ 5k trigram LM. We used the real noisy evaluation set ("et05_real_noisy") consisting of 1,320 utterances for evaluating the methods.

We present some examples of comparative domain translation results with the proposed approaches. Fig. 5 depicts an lmfb spectrogram of noisy speech and its domain transformation results using three different methods[2]. Fig. 5 (b) is the "fake" clean speech translated from the original noisy speech (a), using $G_{T\to S}$ trained only with the Wasserstein adversarial loss. Note that when the cycle consistency loss is not used, $G_{S\to T}$ and $G_{T\to S}$ are trained independently, because they cannot affect each other. Obviously, (b) is totally different from the original utterance (a), and it hardly looks like

[2]Although a domain translation output vector consists of 11 frames of 40-channel lmfb features, we show here only the sequence of the center frames.

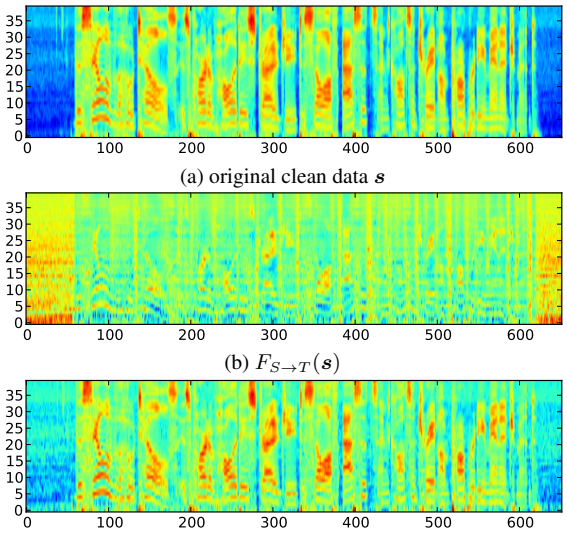
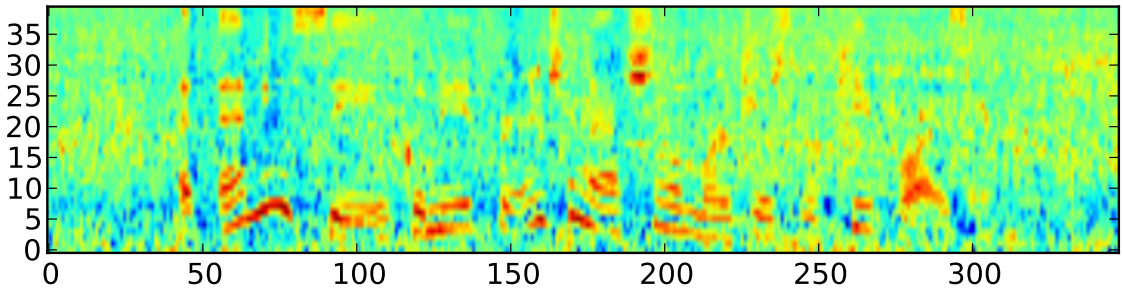


Fig. 6. An example of clean utterance and the translated "fake" noisy data with the proposed method. The vertical axis designates lmfb channel numbers and the horizontal axis designates the time frames with 10ms shift.

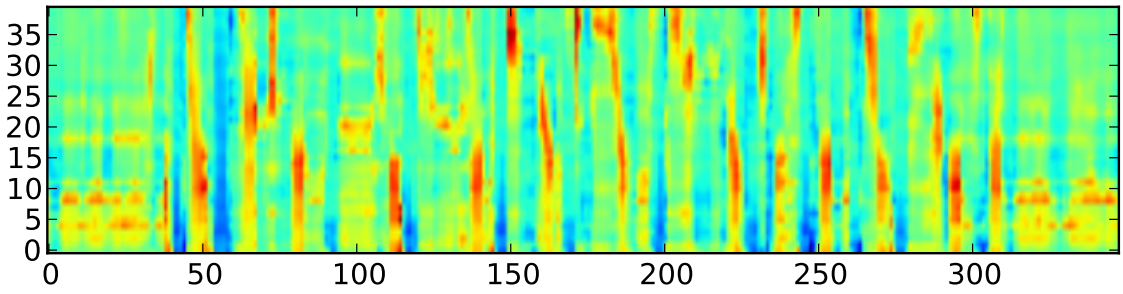
human speech since it does not have formant trajectories. Compared with (b), the translated utterance with a GAN trained using the cycle consistency loss (c) has much more detailed structures as expected. It looks "cleaner" than (a) because it has more blue regions, and it is much more like human speech than (b). However, the apparent phonetic structure in the original data seems to be totally lost in (c), which is fatal for ASR. (d) is the translated data with a GAN which has our proposed network architecture and was trained using the cycle consistency loss. Noise is effectively suppressed and speech is enhanced here, and more importantly, a consonant-vowel phonetic structure is clearly seen in the translated spectrogram.

In Fig. 6, we show a "fake" noisy data translated from a clean utterance using $G_{S\to T}$. We separately present the outputs of two subnetworks (cf. Fig. 2) in order to show the mixing process. (a) is the output of the identity mapping component, which is of course identical to the input s. (b) is the output of the generative network component $F_{S\to T}(\boldsymbol{s})$, and (c) is the resulting noisy version of the input utterance generated by simply summing (a) and (b) together. In this example, we chose the network trained with fixed weights, $\lambda = \mu = 1$, for the purpose of demonstration. We can see that our method can make up a quite realistic noisy utterance which has similar characteristics to a real noisy data such as (a) in Fig. 5.

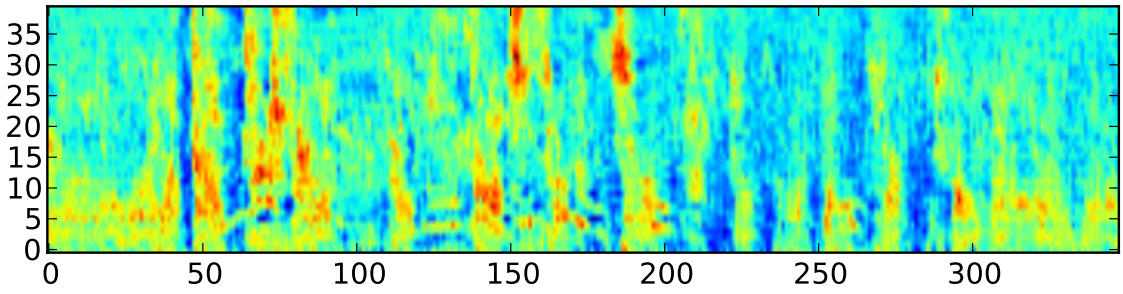
We show the speech recognition results by our proposed methods in Table 1. First, we evaluated the performance of speech enhancement using the noisy to clean mapping $G_{T\to S}$. Speech recognition was performed using the acoustic features of the real noisy test set transformed using $G_{T\to S}$ and the baseline acoustic model trained using the clean data. By comparing the results for the original noisy data (row (1)) and the enhanced data (row (3)), we see that the proposed method effectively enhanced the acoustic features and yielded an improvement of 3.7 points in WER. Moreover, by introducing the submapping-wise scaling factors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, we had a further improvement of 0.78 points (row (4)). We also show the result for a GAN



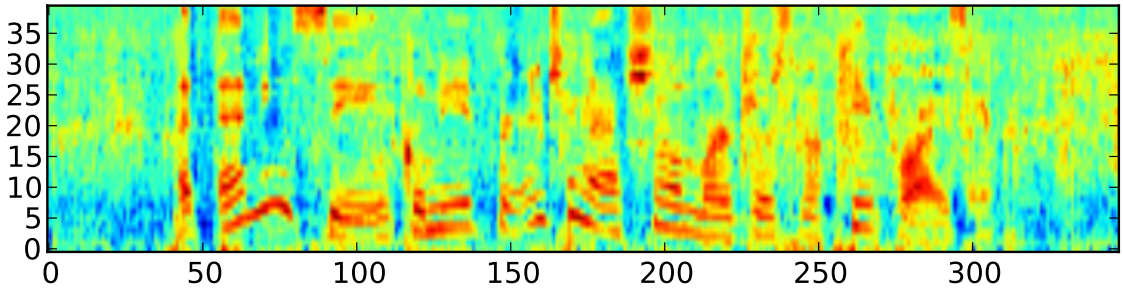
(a) original noisy data



(b) translated ”fake” clean data with standard GAN



(c) translated ”fake” clean data with standard GAN with cycle loss

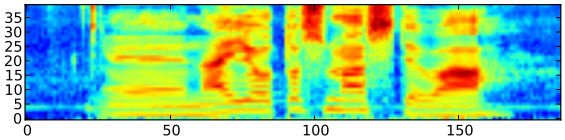


(d) translated ”fake” clean data with proposed method

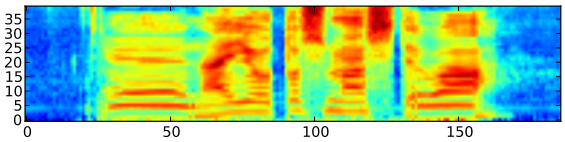
Fig. 5. An example of noisy utterance and the translated ”fake” clean data with various methods. The vertical axis designates lmfb channel numbers and the horizontal axis designates the time frames with 10ms shift.

Table 1. Performance of proposed methods (WER(%))

acoustic model	feature	cycle loss	λ and μ	WER	ID
no adapt.	no adapt.	-	-	41.08	(1)
no adapt.	adapt. with $G_{T\to S}$	no	1, 1	55.45	(2)
		yes	1, 1	37.34	(3)
		yes	trained	36.56	(4)
adapt. with $G_{S\to T}$	no adapt.	yes	1, 1	35.98	(5)
		yes	trained	34.31	(6)



(a) original CSJ-SPS data



(b) translated "fake" JNAS data

Fig. 7. An example of CSJ-SPS (simulated public speech) utterance and the translated "fake" JNAS (read speech) data with the proposed method. The vertical axis designates lmfb channel numbers and the horizontal axis designates the time frames with 10ms shift.

Table 2. Performance of proposed methods applied to speaking style adaptation (WER(%))

source	target	feature	WER
JNAS	CSJ-SPS	no adapt.	26.47
		adapt. with $G_{T\to S}$	25.93
CSJ-APS	CSJ-SPS	no adapt.	17.15
		adapt. with $G_{T\to S}$	16.60

with our proposed architecture trained without the cyle consistency loss in row (2), from which we understand that the constraint that the mappings are cycle-consistent is essential for the desired quality in the translated features. Next, we evaluated the ASR performances of the adapted acoustic models. These models were trained using the "fake" noisy data translated with $G_{S\to T}$. From the results in row (5) and (6), we understand that the model adaptation approach is more effective than the feature enhancement approach (row (3) and (4)), and training $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ is also beneficial for adapting acoustic models. The best model achieved WER of 34.35 % which is a 16 % relative improvement from the baseline. We see this outcome quite promising as a result of our first trial of unsupervised domain mapping, even though it is not yet comparable to the WER of 21.15 % which is achieved by the traditional training approach in which we make direct use of labeled simulated noisy data for acoustic model training.

4.2. Speaking style adaptation

We also applied the proposed method to speaking style adaptation of acoustic models to improve the ASR performance for test data with a different speaking style from the training data.

We used three corpora with different speaking styles, namely, JNAS (Japanese Newspaper Article Sentences), and APS (Academic Public Speaking) and SPS (Simulated Public Speaking) subcorpora from the CSJ (Corpus of Spontaneous Japanese). JNAS is a read speech corpus, and utterances in CSJ-APS and CSJ-SPS have spontaneous speaking styles. CSJ-APS consists of live recordings of academic presentations in public, and speeches in CSJ-SPS were presented in front of a small audience and in a relatively relaxed atmosphere. Speaking style transformation experiments were conducted using JNAS and CSJ-APS as source domains and CSJ-SPS as a target domain. We used the same amount of data (20 hours) for each corpus in training the GANs.

Fig. 7 depicts a lmfb spectrogram of a CSJ-SPS utterance transformed to JNAS style. While a number of differences between the original and transformed data can be observed, the most evident one is the enhanced formant structures in the region from the 20th to around the 50th frame, which corresponds to a filler word consisting of one long vowel "e:". It is a natural consequence considering that a more articulately pronounced vowel is a characteristic of read speech.

We present speech recognition experiment results using the adapted acoustic features in Table 2. The acoustic models were trained using source domain data. The WERs were improved by 0.5 points by adapting acoustic features to the source domains. The most remarkable point is that these improvements were obtained using only acoustic signals of the target domain without any supervision labels. From these results, we understand that our method can be applied to speaking style adaptation as well as noise-robust acoustic model training. Note that we did not conduct acoustic model adaptation with $G_{S\to T}$ in the speaking style translation experiments due to the limited time.

5. COLNCLUSION

We proposed a novel approach to noise-robust acoustic training with GANs which are trained with a cycle consistent loss and have a specially designed architecture for retaining discriminative information in translated data. We demonstrated the effectiveness of the proposed method in noisy speech recognition and speaking style adaptation.

This is our initial attempt to apply deep generative networks for speech recognition. We are interested in extensions of the proposed methods such as introduction of recurrent structures for incorporating longer context information. Another promising direction is to use class information in training GANs [24] to enhance discriminability in translated data.

6. REFERENCES

[1] G.E.Hinton, L.Deng, D.Yu, G.Dahl, A.Mohamed, N.Jaitly, A.Senior, V.Vanhoucke, P.Nguyen, T.Sainath, and B.Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," in *arXiv preprint arXiv:1610.0525*, 2016.

[3] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, Bergul Roomi, and Phil Hall, "English conversational telephone speech recognition by humans and machines," in *arXiv preprint arXiv:1703.02136*, 2017.

[4] Xue Feng, Yaodong Zhang, and James Glass, "Speech Feature Denoising and Dereverberation via Deep Autoencoders for Noisy Reverberant Speech Recognition," in *Proc. ICASSP*, 2014, pp. 1778–1782.

[5] Felix Weninger, Shinji Watanabe, Yuuki Tachioka, and Björn Schuller, "Deep Reccurent De-noising Auto-encoder and Blind De-reverberation for Reverberated Speech Recognition," in *Proc. ICASSP*, 2014, pp. 4656–4660.

[6] X.Lu, Y.Tsao, S.Matsuda, and C.Hori, "Speech enhancement based on deep denoising autoencoder," in *INTERSPEECH*, 2013, pp. 436–440.

[7] Jun Du, Qing Wang, Tian Gao, Yong Xu, Lirong Dai, and Chin-Hui Lee, "Robust speech recognition with speech enhanced deep neural networks," in *INTERSPEECH*, 2014, pp. 616–620.

[8] M.Mimura, S.Sakai, and T.Kawahara, "Reverberant speech recognition combining deep neural networks and deep autoencoders augmented with a phone-class feature," *EURASIP journal on Advances in Signal Processing*, 2015.

[9] K.Kinoshita, M.Delcroix, T.Yoshioka, T.Nakatani, E.Habets, R.Haeb-Umbach, V.Leutnant, A.Sehr, W.Kellermann, R.Maas, S.Gannot, and B.Raj, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA-13)*, 2013.

[10] J.Barker, R.Marxer, E.Vincent, and S.Watanabe, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *Proc. ASRU*, 2015.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *in Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.

[12] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *arXiv preprint arXiv:1703.10593*, 2017.

[13] Li Deng, Mike Seltzer, Dong Yu, Alex Acero, Abdel rahman Mohamed, and Geoffrey Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *INTERSPEECH*, 2010, pp. 1692–1695.

[14] T.Ishii, H.Komiyama, T.Shinozaki, Y.Horiuchi, and S.Kuroiwa, "Reverberant speech recognition based on denoising autoencoder," in *INTERSPEECH*, 2013, pp. 3512–3516.

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*. 2016, pp. 694–711, Springer.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization.," in *arXiv preprint arXiv:1607.08022*, 2016.

[18] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vinvent Dumoulin, and Aaron Courville, "Improved training of Wasserstein GANs," in *arXiv preprint arXiv:1704.00028*, 2017.

[19] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein GAN," in *arXiv preprint arXiv:1701.07875*, 2017.

[20] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *arXiv preprint arXiv:1412.6980*, 2014.

[21] Ossama Abdel-Hamid, Abdel rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *IEEE Trans. Audio, Speech & Language Process.*, vol. 22, no. 10, pp. 1533–1545, 2015.

[22] Vinod Nair and Geoffrey E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of ICML*, 2010, pp. 807–814.

[23] D.Povey, A.Ghoshal, G.Boulianne, L.Burget, O.Glembek, N.Goel, M.Hannemann, P.Motlicek, Y.Qian, P.Schwarz, J.Silovsky, G.Stemmer, and K.Vesely, "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[24] Augustus Odena, Christopher Olah, and Jonathon Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *arXiv preprint arXiv:1610.09585*, 2016.