AN INVESTIGATION INTO LEARNING EFFECTIVE SPEAKER SUBSPACES FOR ROBUST UNSUPERVISED DNN ADAPTATION

Lahiru Samarakoon[◇⋆] *Khe Chai Sim*[†] *Brian Mak*[⋆]

[◇] National University of Singapore
[⋆] Hong Kong University of Science and Technology
[†] Google, Inc.

ABSTRACT

Subspace methods are used for deep neural network (DNN)-based acoustic model adaptation. These methods first construct a subspace and then perform the speaker adaptation as a point in the subspace. This paper aims to investigate the effectiveness of subspace methods for robust unsupervised adaptation. For the analysis, we compare two state-of-the-art subspace methods, namely, the singular value decomposition (SVD)-based bottleneck adaptation and the factorized hidden layer (FHL) adaptation. Both of these methods perform speaker adaptation as a linear combination of rank-1 bases. The main difference between the subspace construction is that FHL adaptation constructs a speaker subspace separate from the phoneme classification space while SVD-based bottleneck adaptation shares the same subspace for both the phoneme classification and the speaker adaptation. So far, no direct comparisons between these two methods are reported. In this work, we compare these two methods for their robustness to unsupervised adaptation on Aurora 4, AMI IHM and AMI SDM tasks. Our findings show that the FHL adaptation outperforms the SVD-based bottleneck adaptation especially in challenging conditions where the adaptation data is limited, or the quality of the adaptation alignments are low.

Index Terms: Automatic Speech Recognition, DNN Adaptation. Subspace Methods.

1. INTRODUCTION

Deep neural network (DNN)-based acoustic modeling has significantly outperformed the conventional Gaussian mixture model (GMM)-based automatic speech recognition (ASR) systems. However, DNNs are still susceptible to performance degradations due to the mismatches between the training and testing conditions. Adaptation techniques reduce the mismatch by changing a well-trained model to match the testing conditions or by transforming the runtime features to match the model.

It is possible to broadly categorize DNN adaptation techniques into two classes: test-only adaptation (simply refers to as adaptation), and adaptive training. The adaptation methods start from a well-trained DNN model and use data from the testing condition to reduce the mismatch. The adaptive training uses both training and testing data to reduce the mismatch. Some adaptation methods use a condition dependent linear layer to augment a well-trained DNN model [1–6]. There are subspace or subset methods where the adaptation is performed to a subset of model parameters or on a pruned model [7–14]. Regularization based adaptation helps to perform the adaptation more conservatively [15, 16]. Recently developed cluster adaptive training (CAT) for DNNs [17, 18], feature normalization techniques like constrained maximum likelihood linear regression (CMLLR) [19], vocal tract length normalization (VTLN) [20], and speaker-aware training (SaT) [21–23] can be considered as adaptive training methods.

A good adaptation technique should be able to perform adaptation in an unsupervised fashion, which is more realistic. In addition, the method should prevent over-fitting to adaptation data, especially when the adaptation data is limited. Furthermore, it is desirable to have a small number of speaker-dependent (SD) parameters (per-speaker footprint) to reduce deployment costs. Subspace methods are proposed to meet these requirements. In this paper, we investigate the singular value decomposition (SVD)-based bottleneck adaptation [9, 10, 24] and the recently proposed factorized hidden layer (FHL) adaptation [25] on their robustness to unsupervised adaptation. We also compare their per-speaker footprint requirements. The evaluations are reported in three benchmark ASR tasks: Aurora 4 [26, 27], Augmented Multi-party Interaction (AMI) [28, 29] individual headset microphone (IHM) and AMI single distance microphone (SDM).

The rest of the paper is organized as follows. Section 2 briefly describes the methods being investigated. In Section 3, we give the details of our experimental setup. The results are reported in Section 4 and we conclude the paper in Section 5.

2. METHODS

In this section, we review the SVD-based adaptation and the FHL adaptation. At the end of this section, we highlight the key differences between the two methods.

2.1. SVD-Based Adaptation

A DNN can be viewed as a model that learns a feature representation as well as a classifier. Each hidden layer learns a more abstract representation ($\mathbf{h}^l$) from the lower layer's representation ($\mathbf{h}^{l-1}$):

$$\mathbf{h}^l = \sigma(\mathbf{W}^l\mathbf{h}^{l-1} + \mathbf{b}^l) \tag{1}$$

where $\mathbf{W}^l$ is the weight matrix, $\mathbf{b}^l$ is the bias vector and σ is the sigmoid activation function.

2.1.1. Training

In SVD-based adaptation [9, 10, 24] a well-trained DNN model is restructured to approximate $\mathbf{W}^l$ with two low-rank matrices using SVD as given in Equation 2:

$$\mathbf{W}^l \approx \mathbf{A}^l\mathbf{C}^{l\top} \tag{2}$$

where $\mathbf{W}^l \in \mathbb{R}^{m\times n}$, $\mathbf{A}^l \in \mathbb{R}^{m\times k}$, $\mathbf{C}^l \in \mathbb{R}^{k\times n}$ and k is the number of retained singular values ($k \ll min(m,n)$). Then, the DNN is retrained to recover from the loss of accuracy.

2.1.2. Adaptation

During the adaptation step, an SD linear transform is estimated between $\mathbf{A}^l$ and $\mathbf{C}^{l\top}$ as given below:

$$\mathbf{W}_s^l \approx \mathbf{A}^l \mathbf{K}_s^l \mathbf{C}^{l\top} \tag{3}$$

where $\mathbf{W}_s^l$ is the SD transformation matrix and $\mathbf{K}_s^l \in \mathbb{R}^{k\times k}$ is the adaptation matrix. Before the adaptation $\mathbf{K}_s^l$ is intialized with the identity matrix. It is possible to perform the adaptation in two ways: namely, SVD full or SVD diagonal. In SVD full adaptation, $\mathbf{K}_s^l$ is a full matrix and can be formulated as a linear interpolation of rank-1 bases as below:

$$\mathbf{W}_s^l \approx \sum_{i=1}^{k}\sum_{j=1}^{k} \mathbf{K}_s^l(i,j)\mathbf{a}_i^l \mathbf{c}_j^{l\top} \tag{4}$$

where $\mathbf{a}_i^l$, $\mathbf{c}_i^l$ are i-th column vectors for $\mathbf{A}^l$, $\mathbf{C}^l$ respectively. In SVD diagonal adaptation, $\mathbf{K}_s^l$ is a diagonal matrix and is formulated as a linear interpolation of rank-1 bases as below:

$$\mathbf{W}_s^l \approx \sum_{i=1}^{k} \mathbf{K}_s^l(i,i)\mathbf{a}_i^l \mathbf{c}_i^{l\top}. \tag{5}$$

2.2. Factorized Hidden Layer (FHL) Adaptation

In the standard SaT where only an SD bias is used, all the phonemes of a speaker are adapted with a fixed bias which is not optimal. Therefore, FHL includes an SD transformation in addition to the SD bias.

$$\mathbf{h}^l = \sigma(\mathbf{W}_s^l \mathbf{h}^{l-1} + \mathbf{b}_s^l) \tag{6}$$

where the SD transformation matrix, $\mathbf{W}_s^l$ is given by:

$$\mathbf{W}_s^l = \mathbf{W}^l + \sum_{i=1}^{|\mathbf{d}_s^l|} \mathbf{d}_s^l(i)\mathbf{B}_i^l \tag{7}$$

where $\{\mathbf{B}_1^l, \mathbf{B}_2^l .., \mathbf{B}_{|\mathbf{d}_s^l|}^l\}$ is the set of basis for the SD transformation and $\mathbf{d}_s^l \in \mathbb{R}^{|\mathbf{d}_s^l|}$ is the SD interpolation vector. Similarly, the SD bias vector, $\mathbf{b}_s^l$, for hidden layer l is given by:

$$\mathbf{b}_s^l = \mathbf{b}^l + \sum_{i=1}^{|\mathbf{v}_s^l|} \mathbf{v}_s^l(i)\mathbf{u}_i^l \tag{8}$$

where $\{\mathbf{u}_1^l, \mathbf{u}_2^l .., \mathbf{u}_{|\mathbf{v}_s^l|}^l\}$ is the set of basis for the SD bias and $\mathbf{v}_s^l \in \mathbb{R}^{|\mathbf{v}_s^l|}$ is the SD interpolation vector.

Furthermore, in [25] $\mathbf{B}_i^l$ weight bases are constrained to be rank-1 matrices. This allows us to formulate the SD transformation as:

$$\begin{aligned}\mathbf{W}_s^l &= \mathbf{W}^l + \sum_{i=1}^{|\mathbf{d}_s^l|} \mathbf{d}_s^l(i)\boldsymbol{\gamma}_i^l \boldsymbol{\psi}_i^{l\top} \\ &= \mathbf{W}^l + \boldsymbol{\Gamma}^l \mathbf{D}_s^l \boldsymbol{\Psi}^{l\top}\end{aligned} \tag{9}$$

where $\mathbf{B}_i^l = \boldsymbol{\gamma}_i^l \boldsymbol{\psi}_i^{l\top}$ and $\mathbf{D}_s^l \in \mathbb{R}^{|\mathbf{d}_s^l|\times|\mathbf{d}_s^l|}$ is a diagonal matrix ($\mathbf{D}_s^l = \mathrm{diag}(\mathbf{d}_s^l)$) and $\boldsymbol{\gamma}_i^l$, $\boldsymbol{\psi}_i^l$ are i-th column vectors for $\boldsymbol{\Gamma}^l$, $\boldsymbol{\Psi}^l$ respectively.

2.2.1. Training

The diagonality of $\mathbf{D}_s^l$ matrix allows us to initialize $\mathbf{d}_s^l$ with the speaker i-vector and train the system in the form given in equation 9. The trained model is referred as the "initialized" model.

2.2.2. Adaptation

The adaptation can be performed in 3 ways namely, full, constrained full and diagonal adaptation. For the full adaptation, for each FHL, the entire matrix $\mathbf{D}_s^l$ is updated. In constrained full matrix adaptation, $\mathbf{D}_s^l$ is updated while keeping the non-diagonal elements shared among all FHLs. During diagonal adaptation, only the diagonal elements of the matrix $\mathbf{D}_s^l$ are updated. More detailed descriptions of FHL adaptation method including the per-speaker footprint calculations can be found in [25].

2.3. Comparison

Both of these adaptation methods use a linear combination of rank-1 bases to perform speaker adaptation. Their main difference lies in the subspace construction. FHL adaptation constructs a speaker subspace separate from the phoneme classification space while SVD-based bottleneck adaptation shares the same subspace for both phoneme classification and speaker adaptation. In the SVD-based adaptation, rank-1 bases are interpolated with 1's for all the training speakers and an SD interpolation vector is estimated for test speakers. However, in FHL adaptation, speaker i-vector is used as the interpolation vector which is later fine-tuned for test speaker in an unsupervised fashion. Furthermore, the best performance for SVD-based adaptation is reported when the adaptation is performed in a single layer [10] while for the best performance, FHL subspaces are shared among layers [25].

3. EXPERIMENTAL SETUP

3.1. Aurora 4

The initial experiments are conducted on the Aurora 4 noisy speech recognition task. Aurora 4 contains multi-condition training set with 83 speakers for training and the development set with 10 speakers for validation. We report the results of the test set with 8 speakers.

First, we extract the Mel-frequency cepstral coefficients (MFCCs) from the speech using a 25 ms window and a 10 ms frame shift. Then the linear discriminant analysis (LDA) features are obtained by first splicing 7 frames of 13-dimensional MFCCs and then projecting downwards to 40 dimensions using LDA. A single semi-tied covariance (STC) transformation [30] is applied on top of the LDA features. The GMM-hidden Markov model (HMM) system for generating the alignments for DNN training is trained on these 40 dimensional LDA+STC features. CMLLR features are extracted after applying an CMLLR transform on top of these LDA+STC features.

We train the DNN-HMM baselines on the LDA+STC and CMLLR features that span a context of 11 neighboring frames. Before being presented to the DNN, cepstral mean variance normalization (CMVN) is performed on the features globally. To train the network, layer-wise discriminative pre-training is used. The initial DNN has 7 sigmoid hidden layers with 2048 units per layer, and

Table 1. Aurora 4 : Word Error Rate (WER %) for DNN baselines trained on LDA+STC features.

Model	Test Set	#params
Full-rank Baseline	11.9	30 M
Low-rank Baseline	11.8	6.5 M

around 2000 senones as the outputs. All the DNNs are trained to optimize the cross-entropy criterion with a mini-batch size of 256. We use CNTK [31] to train the DNNs. The Kaldi [32] is used to build the GMM-HMM systems and for the i-vector extraction. The i-vectors are trained on top of the corresponding acoustic features. The UBM consist of 128 full Gaussians. All decodings are performed with the pruned 5K trigram language model of WSJ0. For all the models, alignments from the GMM-HMM system are used.

3.2. AMI

Next, we use the AMI corpus which contains about 100 hours of meetings conducted in English. The speech is recorded by multiple microphones, including one IHM and a uniform microphone circular array. In experiments, we use the IHM data and the speech from the first microphone in the array which is known as the SDM. We use the ASR split [33] of the corpus where 78 hours of the data are used for training while about 9 hours each are used for evaluation and development sets. We use 90% of the training set for training, and the rest is used as the validation set. The results are reported on the evaluation set.

For both IHM and SDM datasets, we follow the same steps mentioned in the Aurora 4 experimental setup to generate the CMLLR features. For both tasks, GMM-HMM systems that are used to obtain the training alignments are trained on CMLLR features. DNN baselines are also trained on CMLLR features and have 6 sigmoid hidden layers with 2048 units per layer, and around 4000 senones as the outputs. For decodings, we use the trigram language model as used in Kaldi, which is an interpolation of trigram language models trained on AMI and Fisher English transcripts.

4. RESULTS

Table 1 shows results for the baseline models trained on top of the LDA+STC features. The full-rank model is restructured using SVD to obtain the low-rank baseline. We keep the 80% of the total singular values for the weight matrix between the input and the first hidden layer, and only 40% of the total singular values are kept for the rest of the weight matrices. As can be seen, the number of parameters can be reduced from 30M to 6.5M without any loss in accuracy.

Table 2 presents the layer-wise results of SVD full and SVD diagonal adaptation experiments. For each layer, the speaker specific footprint is also mentioned. For SVD full adaptation, we observe significant performance degradations when layer 6 or layer 7 is adapted. A possible reason for this is that the upper layers are more relevant to the phoneme classification, therefore including the speaker information degrades the performance. As can be clearly seen the best performance (8.8%) is reported when SVD full adaptation is performed in layer 3 or layer 4. Therefore, in the rest of the experiments, the SVD full adaptation is performed in layer 3. The speaker specific footprint is in thousands for SVD full adaptation. It is possible to reduce the per-speaker footprint by performing SVD diagonal adaptation. Last two columns of Table 2 presents the results. We observe performance degradations for most of the layers except for layers 1, 4 and 5. The best performance (11.4%) is reported for the layer 1, which is significantly lower than the best performance (8.8%) of the SVD full adaptation.

Table 2. Aurora 4 : WER % for layer-wise SVD full and SVD diagonal adaptation for the low-rank baseline trained on LDA+STC features.

Layer	SVD full		SVD diagonal	
	WER %	Footprint ($k * k$)	WER %	Footprint (k)
1	11.2	30976 (176*176)	11.4	176
2	8.9	53824 (232*232)	23.7	232
3	8.8	53824 (232*232)	13.4	232
4	8.8	53824 (232*232)	11.5	232
5	9.0	57600 (240*240)	11.6	240
6	85.5	61504 (248*248)	89.4	248
7	57.0	78400 (280*280)	98.2	280

Table 3. Aurora 4 : WER % for various adaptation methods for both full-rank and low-rank models trained on LDA+STC features.

Method	Full-rank	Low-rank	Footprint
Baseline	11.9	11.8	-
Baseline + LHUC	10.0	9.7	14336
Baseline + SVD full	-	8.8	53824
4 FHLs initialized	10.6	10.7	100
4 FHLs + diagonal	9.0	9.0	500
4 FHLs + constraind full	8.4	8.3	10400
4 FHLs + full	8.3	8.2	40100

Table 3 summarizes the results for Aurora 4 LDA+STC features for both full-rank and low-rank models. The learning hidden unit contributions (LHUC) adaptation improves the performance significantly over the baselines. The performance of the SVD full adaptation (8.8%) is significantly better than the that of LHUC adaptation (9.7%). Since Aurora 4 has 70.3 minutes per speaker for adaptation, the large number of adaptation parameters estimated in SVD full adaptation results in more gains. We present the results for FHL adaptation on both full-rank and low-rank models because it facilitates direct comparisons with SVD-based adaptation. As in [25], we use a model with 4 FHLs where an SD bias is connected to the first layer and SD transformations are connected to the 4 lowest layers. For both full-rank and low-rank models, FHL adaptation with 4 FHLs reports similar gains. The performance (9.0%) of diagonal adaptation with 4 FHLs is slightly worse than the performance (8.8%) of SVD full adaptation. However, we can get a 99.8% of per-speaker footprint reduction when 4 FHLs with diagonal adaptation is used in comparison to SVD full adaptation. Both constrained full and full adaptations with 4 FHLs performed better than the SVD full adaptation while keeping a smaller per-speaker footprint. We get the best performance when full adaptation is performed on the 4 FHLs model for both full-rank and low-rank models.

Next, we present the results for Aurora 4 models trained on CMLLR features in Table 4. The FHL adaptation experiments on the

Table 4. Aurora 4 : WER % for various adaptation methods for both full-rank and low-rank models trained on CMLLR features.

Method	Full-rank	Low-rank	Footprint
Baseline	9.5	9.5	-
Baseline + SVD full	-	7.5	53824
4 FHLs initialized	8.8	9.1	100
4 FHLs + diagonal	7.9	8.2	500
4 FHLs + constrained full	7.3	7.5	10400
4 FHLs + full	7.2	7.3	40100

Table 5. AMI IHM : WER % for various adaptation methods for both full-rank and low-rank models trained on CMLLR features.

Method	Full-rank	Low-rank	Footprint
Baseline	26.3	26.6	-
Baseline + SVD full	-	25.2	53824
4 FHLs initialized	25.7	26.0	100
4 FHLs + diagonal	24.4	24.7	500
4 FHLs + constrained full	24.7	24.7	10400
4 FHLs + full	24.8	24.7	40100

full-rank model perform slightly better than the low-rank experiments. Similar to the LDA+STC models, the best performance of is reported when the full adaptation is performed on the models with 4 FHLs.

In Table 5, we investigate the various unsupervised adaptation techniques mentioned in this paper on the AMI IHM dataset. We observe a slight performance degradation when the full-rank baseline is restructured using SVD to get the low-rank baseline model. This performance degradation is also reflected in the 4 FHLs initialized model as well as the diagonally adapted 4 FHLs model. All other adaptation methods perform better than the SVD full adaptation. One reason can be that on the AMI evaluation set the data per speaker is smaller (32.2 minutes) compared to the of Aurora 4 (70.3 minutes). For both full-rank and low-rank experiments, the best performance is reported for 4 FHLs with the diagonal adaptation which is a 1.9% absolute improvement over the respective baselines.

Table 6 shows results on the AMI SDM dataset. The low-rank baseline improves the performance slightly by 0.5% over the full-rank baseline. In addition, we only observe a 0.6% slight performance improvement from the SVD full adaptation over the low-rank baseline. We believe this is because lot of adaptation parameters are needed to estimate and the hypotheses used for unsupervised adaptation is of low quality. The smaller gain of other adaptation methods also supports this claim. For both, full-rank and low-rank experiments, the best performance is reported for 4 FHLs with the diagonal adaptation which is a 1.6% absolute improvement over the respective baselines.

As can be clearly seen most of the performance gain of FHL comes after the two-pass adaptation. Therefore, to evaluate the robustness of SVD full adaptation, in comparison to the two-pass adaptation of FHL, we perform SVD full adaptation on low-rank 4 FHLs initialized models for all the datasets used in this paper. Table 7 presents the results. The best performance of 8.0% is reported when

Table 6. AMI SDM : WER % for various adaptation methods for both full-rank and low-rank models trained on CMLLR features.

Method	Full-rank	Low-rank	Footprint
Baseline	53.2	52.7	-
Baseline + SVD full	-	52.1	53824
4 FHLs initialized	52.9	52.6	100
4 FHLs + diagonal	51.6	51.1	500
4 FHLs + constrained full	51.8	52.2	10400
4 FHLs + full	51.8	52.2	40100

Table 7. WER % after the SVD full adaptation of 4 FHLs Initialized models trained on all datasets.

Dataset	Features	WER %
Aurora 4	LDA+STC	8.0
Aurora 4	CMLLR	7.2
AMI IHM	CMLLR	25.0
AMI SDM	CMLLR	52.1

the low-rank model with 4 FHLs is adapted using the SVD full adaptation for Aurora 4 LDA+STC features which is 0.2% absolute improvement over the 4 FHLs + full adaptation (Table 3). Similarly, for Aurora 4 CMLLR experiments, 4 FHLs initialized + SVD full reports 0.1% absolute improvement over the 4 FHLs + full adaptation (Table 4). This is due to that fact that Aurora 4 has a large amount of speech (70.3 minutes) per-speaker for adaptation, and the large number of adaptation parameters estimated in SVD full adaptation results in more gains. However, for more challenging AMI IHM and AMI SDM tasks, 4 FHLs + diagonal adaptation outperforms the 4 FHLs + SVD full adaptation. Furthermore, 4 FHLs + SVD full adaptation increases the per-speaker footprint significantly compared to all other test-only adaptation methods.

5. CONCLUSION

In this paper, we have investigated the effectiveness of subspace methods for robust unsupervised adaptation. We compared two state-of-the-art subspace methods. Namely, the singular value decomposition (SVD)-based adaptation and the factorized hidden layer (FHL) adaptation. The FHL adaptation constructs a speaker subspace separate from the phoneme classification space while SVD-based adaptation shares the same subspace for both phoneme classification and speaker adaptation. We conducted experiments in three benchmark ASR tasks: Aurora 4, AMI IHM and AMI SDM. First, we investigated the SVD based bottleneck adaptation and the FHL adaptation method individually. Then, we also investigated the combination of SVD full adaptation with the FHL adaptation. Our findings show that when a significant amount of adaptation data is available, the SVD full adaptation can be combined with the FHL adaptation to get the best performance. However, in more challenging conditions where the adaptation data is limited or the adaptation alignments are of low quality, having a separate subspace for adaptation is more robust and also provides smaller per-speaker footprint requirements.

6. REFERENCES

[1] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Eurospeech*. ISCA, 1995, pp. 2183–2186.

[2] B. Li and K. Sim, "Comparison of discriminative input and output transformation for speaker adaptation in the hybrid NN/HMM systems," in *INTERSPEECH*. ISCA, 2010, pp. 526–529.

[3] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," in *INTERSPEECH*. ISCA, 1995.

[4] J. Trmal, J. Zelinka, and L. Müller, "Adaptation of a feedforward artificial neural network using a linear transform," in *Text, Speech and Dialogue*. Springer, 2010, pp. 423–430.

[5] Y. Xiao, Z. Zhang, S. Cai, J. Pan, and Y. Yan, "A initial attempt on task-specific adaptation for deep neural network-based large vocabulary continuous speech recognition," in *INTERSPEECH*. ISCA, 2012.

[6] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training," in *ICASSP*. IEEE, 2006, pp. 1189–1192.

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

[8] S. Xue, H. Jiang, and L. Dai, "Speaker adaptation of hybrid NN/HMM model for speech recognition based on singular value decomposition," in *ISCSLP*. IEEE, 2014, pp. 1–5.

[9] 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 *ICASSP*. IEEE, 2014, pp. 6359–6363.

[10] K. Kumar, C. Liu, K. Yao, and Y. Gong, "Intermediate-layer DNN adaptation for offline and session-based iterative speaker adaptation," in *INTERSPEECH*. ISCA, 2015.

[11] S. Dupont and L. Cheboub, "Fast speaker adaptation of artificial neural networks for automatic speech recognition," in *ICASSP*. IEEE, 2000, pp. 1795–1798.

[12] J. Stadermann and G. Rigoll, "Two-stage speaker adaptation of hybrid tied-posterior acoustic models," in *ICASSP*. IEEE, 2005, pp. 977–980.

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

[14] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *SLT*. IEEE, 2014, pp. 171–176.

[15] 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 *ICASSP*. IEEE, 2013, pp. 7893–7897.

[16] Y. Huang and Y. Gong, "Regularized sequence-level deep neural network model adaptation," in *INTERSPEECH*. ISCA, 2015.

[17] T. Tian, Q. Yanmin, Y. Maofan, Z. Yimeng, and K. Yu, "Cluster adaptive training for deep neural network," in *ICASSP*. IEEE, 2015, pp. 4325–4329.

[18] C. Wu and M. J. F. Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *ICASSP*. IEEE, 2015, pp. 4315–4319.

[19] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[20] L. Lee and R. Rose, "Speaker normalization using efficient frequency warping procedures," in *ICASSP*, vol. 1. IEEE, 1996, pp. 353–356.

[21] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription," in *ICASSP*. IEEE, 2014, pp. 6334–6338.

[22] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*. IEEE, 2013, pp. 55–59.

[23] L. Samarakoon and K. Sim, "Learning factorized transforms for speaker normalization," in *ASRU*. IEEE, 2015.

[24] Y. Zhao, J. Li, and Y. Gong, "Low-rank plus diagonal adaptation for deep neural networks," in *ICASSP*. IEEE, 2016, pp. 5005–5009.

[25] L. Samarakoon and K. Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2241–2250, 2016.

[26] N. Parihar, J. Picone, D. Pearce, and H. Hirsch, "Performance analysis of the Aurora large vocabulary baseline system," in *EUSIPCO*. IEEE, 2004, pp. 553–556.

[27] S. Yeung and M. Siu, "Improved performance of Aurora-4 using HTK and unsupervised MLLR adaptation," in *International Conference on Spoken Language Processing*, 2004.

[28] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.

[29] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI meeting corpus," *Language Resources and Evaluation*, vol. 41, no. 2, pp. 181–190, 2007.

[30] M. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[31] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang *et al.*, "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR, Microsoft Research, 2014, https://github.com/Microsoft/CNTK, Tech. Rep., 2014.

[32] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The Kaldi speech recognition toolkit," in *ASRU*. IEEE, 2011.

[33] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *ASRU*. IEEE, 2013, pp. 285–290.