Learning Hidden Unit Contributions for Unsupervised Acoustic Model Adaptation

Learning Hidden Unit Contributions for Unsupervised Acoustic Model Adaptation

Pawel Swietojanski, *Student Member, IEEE,* Jinyu Li, *Member, IEEE,* and Steve Renals, *Fellow, IEEE*

***Abstract*—This work presents a broad study on the adaptation of neural network acoustic models by means of learning hidden unit contributions (LHUC) – a method that linearly re-combines hidden units in a speaker- or environment-dependent manner using small amounts of unsupervised adaptation data. We also extend LHUC to a speaker adaptive training (SAT) framework that leads to a more adaptable DNN acoustic model, working both in a speaker-dependent and a speaker-independent manner, without the requirements to maintain auxiliary speaker-dependent feature extractors or to introduce significant speaker-dependent changes to the DNN structure. Through a series of experiments on four different speech recognition benchmarks (TED talks, Switchboard, AMI meetings, and Aurora4) comprising 270 test speakers, we show that LHUC in both its test-only and SAT variants results in consistent word error rate reductions ranging from 5% to 23% relative depending on the task and the degree of mismatch between training and test data. In addition, we have investigated the effect of the amount of adaptation data per speaker, the quality of unsupervised adaptation targets, the complementarity to other adaptation techniques, one-shot adaptation, and an extension to adapting DNNs trained in a sequence discriminative manner.**

I. Introduction and Summary

SPEECH recognition accuracies have improved substantially over the past several years through the use of (deep) neural network (DNN) acoustic models. Hinton et al [1] report word error rate (WER) reductions between 10–32% across a wide variety of tasks, compared with discriminatively trained Gaussian mixture model (GMM) based systems. These results use neural networks as part of both hybrid DNN/HMM (hidden Markov model) systems [1]–[5] in which the neural network provides a scaled likelihood estimate to replace the GMM, and as tandem or bottleneck feature systems [6], [7] in which the neural network is used as a discriminative feature extractor for a GMM-based system. For many tasks it has been observed that GMM-based systems (with tandem or bottleneck features) that have been adapted to the talker are more accurate than unadapted hybrid DNN/HMM systems [8]–[10], indicating that the adaptation of DNN acoustic models is an important topic that merits investigation.

P Swietojanski and S Renals are with the Centre for Speech Technology Research, University of Edinburgh, Edinburgh EH89AB, U.K., email: {p.swietojanski,s.renals}@ed.ac.uk

J Li is with Microsoft Corporation. One Microsoft Way, WA, USA, email: jinyu.li@microsoft.com

PS and SR were supported by EPSRC Programme Grant grant EP/I031022/1 *Natural Speech Technology* (NST) and the European Union under H2020 project *SUMMA*, grant agreement 688139. The NST research data collection may be accessed at http://datashare.is.ed.ac.uk/handle/10283/786. This research utilised the K40 GPGPU board donated by NVIDA Corporation.

Acoustic model adaptation [11] aims to normalise the mismatch between training and runtime data distributions that arises owing to the acoustic variability across speakers, as well as other distortions introduced by the channel or acoustic environment. In this paper we investigate unsupervised model-based adaptation of DNN acoustic models to speakers and to acoustic environments, using a recently introduced method called *Learning Hidden Unit Contributions* (`LHUC`) [12]–[14]. We present the `LHUC` approach both in the context of test-only adaptation, and an extension to speaker-adaptive training (SAT), referred to as `SAT-LHUC` [14]. We present an extensive experimental analysis using four standard corpora: TED talks [15], AMI [16], Switchboard [17] and Aurora4 [18]. These experiments include: adaptation of both cross-entropy and sequence trained DNN acoustic models (Sec. VI-A–VI-C); an analysis in terms of the quality of adaptation targets, quality of adaptation data and the amount of adaptation data (Sec. VI-D); complementarity with feature-space adaptation techniques based on maximum likelihood linear regression [19] (Sec. VI-E); and application to combined speaker and environment adaptation (Sec. VII).

II. Review of Neural Network Acoustic Adaptation

Approaches to the adaptation of neural network acoustic models can be considered as operating either in the feature space, or in the model space, or as a hybrid approach in which speaker-, utterance-, or environment-dependent auxiliary features are appended to the standard acoustic features.

The dominant technique for estimating *feature space transforms* is constrained (feature-space) MLLR, referred to as CMLLR or fMLLR [19]. fMLLR is an adaptation method developed for GMM-based acoustic models, in which an affine transform of the input acoustic features is estimated by maximising the log-likelihood that the model generates the adaptation data based on first pass alignments. To use fMLLR with a DNN-based system, it is first necessary to train a complete GMM-based system, which is then used to estimate a single input transform per speaker. The transformed feature vectors are then used to train a DNN in a speaker adaptive manner and another set of transforms is estimated (using the GMM) during evaluation for unseen speakers. This technique has been shown to be effective in reducing WER across several different data sets, in both hybrid and tandem approaches [1], [4], [8], [9], [20]–[23]. Similar techniques have also been developed to operate directly on neural networks. The linear input network (LIN) [24], [25] defines an additional

speaker-dependent layer between the input features and the first hidden layer, and thus has a similar effect to fMLLR. This technique has been further developed to include the use of a tied variant of LIN in which each of the input frames is constrained to have the same linear transform [4], [26]. LIN and tied-LIN have been mostly used in test-only adaptation schemes; to make use of fMLLR transforms one needs to perform SAT training, which can usually better compensate against variability in acoustic space.

An alternative speaker-adaptive training approach – *auxiliary features* – augments the acoustic feature vectors with additional speaker-specific features computed for each speaker at both training and test stages. There has been considerable recent work exploring the use of i-vectors [27] for this purpose. I-vectors, which can be regarded as basis vectors which span a subspace of speaker variability, were first used for adaptation in a GMM framework by Karafiat et al [28]. Saon et al [29] used i-vectors to augment the input features of DNN-based acoustic models, and showed that appending 100-dimensional i-vectors for each speaker resulted in a 10% relative reduction in WER on Switchboard (and a 6% reduction when the input features had been transformed using fMLLR). Gupta et al [30] obtained similar results, and Karanasou et al [31] presented an approach in which the i-vectors were factorised into speaker and environment parts. Miao et al [32] proposed to transform i-vectors using an auxiliary DNN which produced speaker-specific transforms of the original feature vectors, similar to fMLLR. Other examples of auxiliary features include the use of speaker-specific bottleneck features obtained from a speaker separation DNN used in a distant speech recognition task [33], the use of out-of-domain tandem features [23], and speaker codes [34]–[36] in which a specific set of units for each speaker is optimised. Speaker codes require speaker adaptive (re-)training, owing to the additional connection weights between codes and hidden units.

Model-based adaptation relies on a direct update of DNN parameters. Liao [37] investigated supervised and unsupervised adaptation of different weight subsets using a few minutes of adaptation data. On a large net (60M weights), up to 5% relative improvement was observed for unsupervised adaptation when all weights were adapted. Yu et al [38] have explored the use of regularisation for adapting the weights of a DNN, using the Kullback-Liebler (KL) divergence between the speaker-independent (SI) and speaker-dependent (SD) output distributions, resulting in a 3% relative improvement on Switchboard. This approach was also recently used to adapt all parameters of sequence-trained models [39]. One can also reduce the number of speaker-specific parameters through a different forms of factorisation [40], [41]. Ochiai et al [42] have also explored regularised speaker adaptive training with a speaker-dependent layer.

Directly adapting the weights of a large DNN results in extremely large speaker-dependent parameter sets, and a computationally intensive adaptation process. Smaller subsets of the DNN weights may be modified, including output layer biases [43], the bias and slope of hidden units [44] or training the models with differentiable pooling operators [45], which are then adapted in SD fashion. Siniscalchi et al [46] also investigated the use of Hermite polynomial activation functions, whose parameters are estimated in a speaker adaptive fashion. One can also adapt the top layer in a Bayesian fashion resulting in a maximum a posteriori (MAP) approach [47], or address the sparsity of context-dependent tied-states when few adaptation data-points are available by using multi-task adaptation, using monophones to adapt the context-dependent output layer [48], [49]. A similar approach, but using a hierarchical output layer (tied-states followed by monophones) rather than multi-task adaptation, has also been proposed [50].

III. Learning Hidden Unit Contributions (LHUC)

A neural network may be viewed as a set of *adaptive basis functions*. Under certain assumptions on the family of target functions f^* (as well as on the model structure itself) the neural network can act as an universal approximator [51]–[53]. That is, given some vector of input random variables $\mathbf{x} \in \mathbb{R}^d$ there exists a neural network $f_n(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ of the form

$$f_n(\mathbf{x}) = \sum_{k=1}^{n} r_k \psi(\mathbf{w}_k^\top \mathbf{x} + b_k) \tag{1}$$

which can approximate f^* with an arbitrarily small error ϵ with respect to a distance measure such as mean square error (provided n is sufficiently large):

$$||f^*(\mathbf{x}) - f_n(\mathbf{x})||_2 \leq \epsilon. \tag{2}$$

In (1) $\psi : \mathbb{R} \rightarrow \mathbb{R}$ is an element-wise non-linear operation applied after an affine transformation which forms an adaptive basis function parametrised by a set of biases $b_k \in \mathbb{R}$ and a weight vectors $\mathbf{w}_k \in \mathbb{R}^{d_\mathbf{x}}$. The target approximation may then be constructed as a linear combination of the basis functions, each weighted by $r_k \in \mathbb{R}$. The formulation can be extended to m-dimensional mappings $f_n^m(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ simply by splicing the models in (1) m times. The properties also hold true when considering deeper (nested) models [51] (Corollaries 2.6 and 2.7).

DNN training results in the hidden units learning a joint representation of the target function and becoming specialised and complementary to each other. Generalisation corresponds to the learned combination of basis functions continuing to approximate the target function when applied to unseen test data. This interpretation motivates the idea of using LHUC– Learning Hidden Unit Contributions – for test-set adaptation. In LHUC the network's basis functions, previously estimated using a large amount of training data, are kept fixed. Adaptation involves modifying the combination of hidden units in order to minimise the adaptation loss based on the adaptation data. Fig. 1 illustrates this approach for a regression problem, where the adaptation is performed by linear re-combination of basis functions changing only the r parameters from eq. (1).

The key idea of LHUC is to explicitly parametrise the amplitudes of each hidden unit (either in fully-connected and convolutional layers after max-pooling), using a speaker-dependent amplitude function. Let $h_j^{l,s}$ denote the j-th hidden unit activation (basis) in layer l, and let $r_j^{l,s} \in \mathbb{R}$ denote the s-th speaker-dependent amplitude function:

$$h_j^{l,s} = \xi(r_j^{l,s}) \circ \psi_j \left(\mathbf{w}_j^{l\top} \mathbf{x} + b_j^l \right). \tag{3}$$

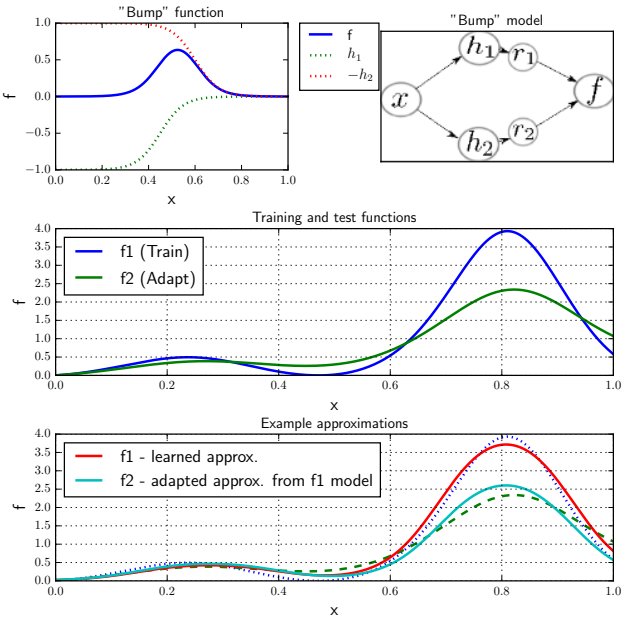


Fig. 1. Example illustration of how LHUC performs adaptation (best viewed in color). Top: A "bump" model (eq. 1) with two hidden units can approximate "bump" functions. Middle: To learn function f_2 given training data f_1 (middle), we splice two "bump" functions together (4 hidden units, one input/output) to learn an approximation of function f_1. Bottom: LHUC adaptation of the model optimised to f_1 and adapted to f_2 using LHUC scaling parameters. Image reproduced from [14].

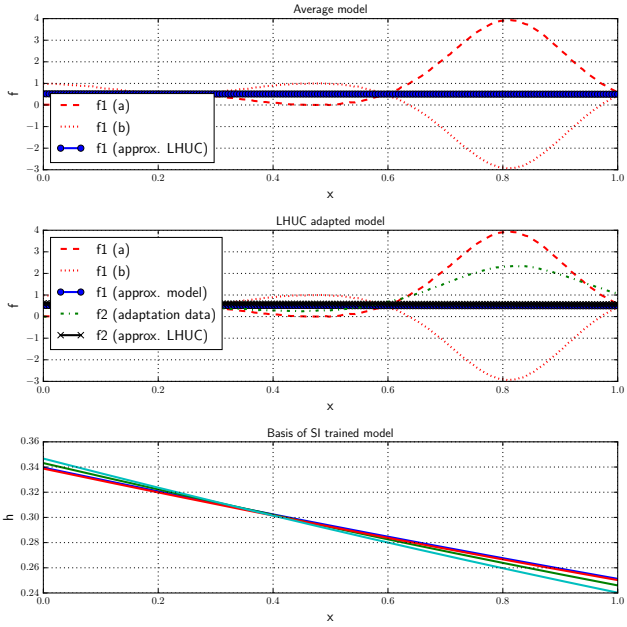


Fig. 2. A 4-hidden-unit model trained to $f1(a)$ and $f1(b)$) for an SI approach (top) and an adapted representation (middle) keeping the resulting basis functions fixed (bottom). (Best viewed in color.)

The amplitude is modelled using a function $\xi : \mathbb{R} \rightarrow \mathbb{R}^+$ – typically a sigmoid with range $(0, 2)$ [13], but an identity function could be used [54]. $\mathbf{w}_j^l$ is the jth column of the corresponding weight matrix $\mathbf{W}^l$, b_j^l denotes the bias, ψ is the hidden unit activation function (unless stated otherwise, this is assumed to be sigmoid), and $\circ$ denotes a Hadamard product[1]. ξ constrains the range of the hidden unit amplitude scaling (compare with Fig. 1) hence directly affecting the adaptation transform capacity – this may be desirable when adapting with potentially noisy unsupervised targets (see Sec. VI-A). LHUC adaptation progresses by setting the speaker-specific amplitude parameters $r_j^{l,s}$ using gradient descent with targets provided by the adaptation data.

The idea of directly learning hidden unit amplitudes was proposed in the context of an adaptive learning rate schedule by Trentin [55], and was later applied to supervised speaker adaptation by Abdel-Hamid and Jiang [12]. The approach was extended to unsupervised adaptation, non-sigmoid non-linearities, and large vocabulary speech recognition by Swietojanski and Renals [13]. Other adaptive transfer function methods for speaker adaptation have also been proposed [44], [46], as have "basis" techniques [56]–[58]. However, the basis in the latter works involved re-tuning parallel models on pre-defined clusters (gender, speaker, environment) in a supervised manner; the adaptation then relied on learning linear combination coefficients for those sub-models on adaptation data.

[1]Although the equations are given in scalar form, we have used Hadamard product notation to emphasise the operation that would be performed once expanded to full-rank matrices.

IV. Speaker Adaptive Training LHUC (SAT-LHUC)

When LHUC is applied as a test-only adaptation it assumes that the set of speaker-independent basis functions estimated on the training data provides a good starting point for further tuning to the underlying data distribution of the adaptation data (Fig. 1). However, one can derive a counter-example where this assumption fails: the top plot of Fig. 2 shows example training data uniformly drawn from two competing distributions $f1(a)$ and $f1(b)$ where the linear recombination of the resulting basis in the average model (Fig 2 bottom), provides a poor approximation of adaptation data.

This motivates combining LHUC with speaker adaptive training (SAT) [59] in which the hidden units are trained to capture both good average representations and speaker-specific representations, by estimating speaker-specific hidden unit amplitudes for each training speaker. This is visualised in Fig. 3 where, given the prior knowledge of which data-point comes from which distribution, we estimate a set of parallel LHUC transforms (one per distribution) as well as one extra transform which is responsible for modelling average properties. The top of Fig. 3 shows the same experiment as in Fig 2 but with three LHUC transforms – one can see that the 4-hidden-unit MLP in this scenario was able to capture each of the underlying distributions as well as the average aspect well, given the LHUC transform. At the same time, the resulting basis functions (Fig 3, bottom) are a better starting point for the adaptation (Fig. 3, middle).

The examples presented in Figs. 2 and 3 could be solved by breaking the symmetry through rebalancing the number of training data-points for each function, resulting in less trivial and hence more adaptable basis functions in the average model. However, as we will show experimentally later, similar

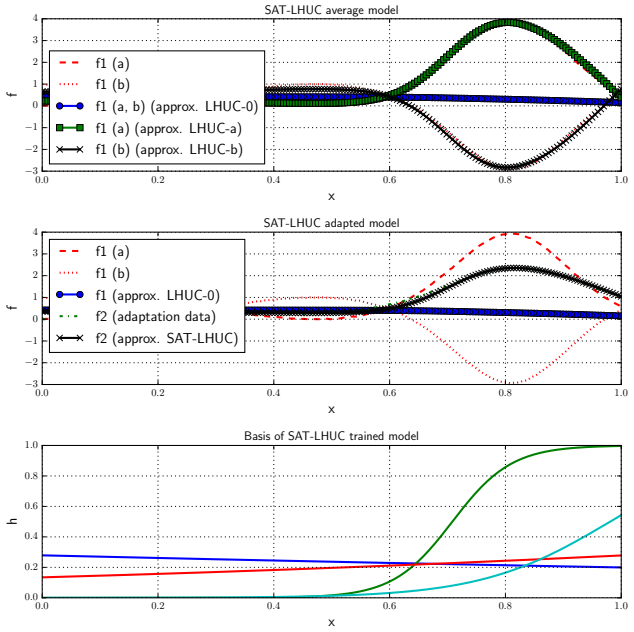


Fig. 3. Learned solutions using three different SAT−LHUC transforms and shared basis functions: LHUC-0 learns to provide a good average fit to both distributions $f1(a)$ and $f1(b)$) at the same time, while LHUC-a and LHUC-b are tasked to fit either $f1(a)$ or $f1(b)$, respectively. The bottom plot shows the resulting basis functions (activations of 4 hidden units) of the SAT−LHUC training approach - one can observe SAT−LHUC provides a richer set of basis function which can fit the data well on average, and can also capture some underlying characteristics necessary to reconstruct target training data – using different LHUC transforms, this property is also visualised in the middle plot. (Best viewed in color.)

effects are also present in high-dimensional speech data, and SAT−LHUC training allows more tunable canonical acoustic models to be built, that can be better tailored to particular speakers through adaptation.

Test-only adaptation for SAT−LHUC remains the same as for LHUC– the set of speaker-dependent LHUC parameters $\theta^s_{LHUC} = \{r^{l,s}_j\}$ is inserted for each test speaker and their values optimised from unsupervised adaptation data. We also use a set of LHUC transforms θ^s_{LHUC}, where $s = 1 \dots S$, for the training speakers which are jointly optimised with the speaker-independent parameters $\theta_{SI} = \{\mathbf{W}^l, \mathbf{b}^l\}$. There is an additional speaker-independent LHUC transform, denoted by θ^0_{LHUC}, which allows the model to be used in speaker-independent fashion, for example, to produce first pass adaptation targets. This joint learning process of hidden units with speaker-dependent LHUC scalers is important, as it results in a more tunable canonical acoustic model that can be better adjusted to unseen speakers at test time, as we have illustrated in Fig. 3 and demonstrated on adaptation tasks in the following sections.

To perform SAT training with LHUC, we use the negative log likelihood and maximise the posterior probability of obtaining the correct context-dependent tied-state c_t given observation vector $\mathbf{x}_t$ at time t:

$$\mathcal{L}_{SAT}(\theta_{SI}, \theta_{SD}) = -\sum_{t \in D} \log P(c_t|\mathbf{x}^s_t; \theta_{SI}; \theta^{m_t}_{LHUC}) \quad (4)$$

where s denotes the sth speaker, $m_t \in \{0, s\}$ selects the SI

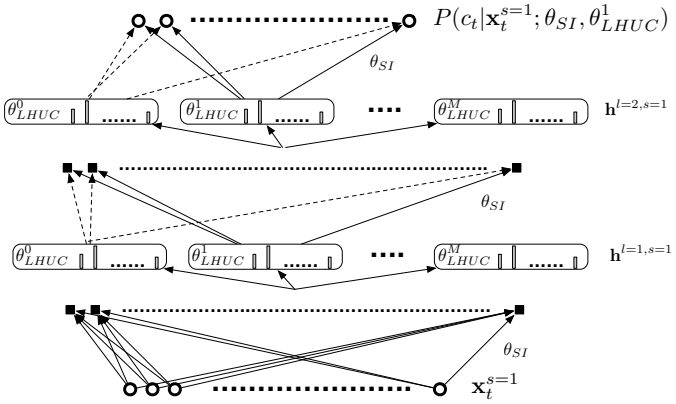


Fig. 4. Schematic of SAT-LHUC training, with a data point from speaker $s = 1$. Dashed line indicates an alternative route through the SI LHUC transform.

or SD LHUC transforms from $\theta_{SD} \in \{\theta^0_{LHUC}, \dots, \theta^S_{LHUC}\}$ based on a Bernoulli distribution:

$$k_t \sim \text{Bernoulli}(\gamma) \quad (5)$$

$$m_t = \begin{cases} s & \text{if } k_t = 0 \\ 0 & \text{if } k_t = 1 \end{cases} \quad (6)$$

where γ is a hyper-parameter specifying the probability the given example is treated as SI. The SI/SD split (determined by equations (5) and (6)) can be performed at speaker, utterance or frame level. We further investigate this aspect in section VI-B. The SAT−LHUC model structure is depicted in Fig 4; notice the alternative routes of forward and backward passes for different speakers.

Denote by $\partial\mathcal{L}_{SAT}/\partial h^{l,s}_j$ the error back-propagated to the jth unit at the lth layer (eq. (3)). To back propagate through the transform one needs to element-wise multiply it by the transform itself, as follows:

$$\frac{\partial\mathcal{L}_{SAT}}{\partial\psi^l_j} = \frac{\partial\mathcal{L}_{SAT}}{\partial h^{l,s}_j} \circ \xi(r^{l,s}_j)\,. \quad (7)$$

To obtain the gradient with respect to $r^{l,s}_j$:

$$\frac{\partial\mathcal{L}_{SAT}}{\partial r^{l,s}_j} = \frac{\partial\mathcal{L}_{SAT}}{\partial h^{l,s}_j} \circ \frac{\partial\xi(r^{l,s}_j)}{\partial r^{l,s}_j} \circ \psi^l_j\,. \quad (8)$$

When performing mini-batch SAT training one needs to explicitly take account of the fact that different data-points may flow through different transforms: hence the resulting gradient for $r^{l,s}_j$ for the sth speaker is the sum of the partial gradients belonging to speaker s:

$$\frac{\partial\mathcal{L}_{SAT}}{\partial r^{l,s}_j} = \sum_{t, m_t = s} \frac{\partial\mathcal{L}_{SAT}}{\partial h^{l,s}_j} \circ \frac{\partial\xi(r^{l,s}_j)}{\partial r^{l,s}_j} \circ \psi^l_j\,, \quad (9)$$

or 0 in case no data-points for sth speaker in the given mini-batch were selected. All adaptation methods studied in this paper require first-pass decoding to obtain adaptation targets to either estimate fMLLR transforms for unseen test speakers or to perform DNN speaker-dependent parameter update.

TABLE I
CORPUS STATISTICS RELATED TO SAT AND ADAPTATION. IN PARENTHESES WE GIVE THE ACTUAL NUMBER OF SPEAKERS.

	Training		Test	
Corpora	#Clusters	Time (h)	#Clusters	Time (h)
Aurora4	83 (83)	15	8 (8)	8.8
AMI	547 (155)	80	135 (36)	17.5
TED	788 (788)	143	39 (39)	9.0
SWBD	4804 (4000)	283	80 (80)	3.6

V. EXPERIMENTAL SETUPS

We experimentally investigated LHUC and SAT-LHUC using four different corpora: the TED talks corpus [15] following the IWSLT evaluation protocol (www.iwslt.org); the Switchboard corpus of conversational telephone speech [17] (ldc.upenn.edu); the AMI meetings corpus [16], [60] (corpus.amiproject.org); and the Aurora4 corpus of read speech with artificially corrupted acoustic environments [18] (catalog.elra.info). Unless explicitly stated otherwise, the models share similar structure across the tasks – DNNs with 6 hidden layers (2,048 units in each) using a sigmoid non-linearity. The output logistic regression layer models the distribution of context-dependent clustered tied states [5]. The features are presented in 11 (± 5) frame long context windows. All the adaptation experiments, unless explicitly stated otherwise, were performed unsupervised.

Below, we briefly describe each of the above corpora and its specific experimental configurations. The collective summary of adaptation-related statistics for each corpora is given in Table I. Note that we adapt to the headset or the side of a conversation, rather than the actual speaker (unless stated otherwise). As a result, the actual number of clusters (or estimated transforms) during training may differ from the number of physical speakers in the data.

TED: We carried out experiments using a corpus of publicity available TED talks (www.ted.com) following the IWSLT ASR evaluation protocol [61] (iwslt.org). The training data consisted of 143 hours of speech (813 talks) and the systems follow our previously described recipe [9]. In this work however, compared to our previous works [9], [13], [45], our systems employ more accurate language models developed for our IWSLT–2014 systems [62]: in particular, the final reported results use a 4-gram language model estimated from 751 million words. The baseline TED acoustic models are trained on unadapted PLP features with first and second order time derivatives. We present results on four predefined IWSLT test sets: dev2010, tst2010, tst2011 and tst2013 containing 8, 11, 8 and 28 ten-minute talks respectively. We use tst2010 and/or tst2013 to perform more detailed analyses. A collective summary of results on all TED testsets is reported in Sec. VI-F.

AMI: We follow the Kaldi GMM recipe described in [63] and use acoustics from either Individual Headset Microphone (IHM) or Single Distant Microphone (SDM). In addition to cepstral features, we also trained a separate set of models using 40 mel-filter-bank (FBANK) features for which fMLLR transforms cannot be easily obtained (though not impossible [64]), and for which LHUC offers an interesting adaptation alternative. We also evaluated the effectiveness of LHUC and SAT-LHUC applied to convolutional networks [8], [65], [66], trained as described in [67] but with 300 convolutional filters. We decoded with a pruned 3-gram language model estimated from 800k words of AMI training transcripts interpolated with an LM trained on Fisher conversational telephone speech transcripts (1M words).

Switchboard: We use the Kaldi GMM recipe [68], [69], using Switchboard–1 Release 2 (LDC97S62). Our baseline unadapted acoustic models were trained on LDA/MLLT features. The results are reported on the full Hub5 00 set (LDC2002S09) to which we will refer as eval2000. The eval2000 contains two types of data, Switchboard (SWBD) – which is better matched to the training data – and CallHome English (CHE). Our reported results use 3-gram LMs estimated from Switchboard and Fisher data.

Aurora4: The Aurora 4 task is a small scale, medium vocabulary noise and channel ASR robustness task based on the Wall Street Journal corpus [18]. We train our ASR models using the multi-condition training set. One half of the training utterances were recorded using a primary Sennheiser microphone, and the other half was collected using one of 18 other secondary microphones. The multi-condition set contains noisy utterances corrupted with one of six different noise types (airport, babble, car, restaurant, street traffic and train station) at 10-20 dB SNR. The standard Aurora 4 test set (eval92) consists of 330 utterances, which are used in 14 test conditions (4620 utterances in total). The same six noise types used during training are used to create noisy test utterances with SNRs ranging from 5-15dB SNR, resulting in a total of 14 test sets. These test sets are commonly grouped into 4 subsets – clean (group A, 1 test case), noisy (group B, 6 test cases), clean with channel distortion (group C, 1 test case) and noisy with channel distortion (group D, 6 test cases). We decode with the standard task's 5k words bigram LM.

VI. RESULTS

A. LHUC hyperparameters

Our initial study concerned the hyper-parameters used with LHUC adaptation. First, we used the TED talks to investigate how the word error rate (WER) is affected by adapting different layers in the model using LHUC transforms. The results, graphed in Fig. 5 (a), indicated that adapting only the bottom layer brings the largest drop in WER; however, adapting more layers further improves the accuracy for both LHUC and SAT-LHUC approaches (adapting the other way round – starting from the top layer – is much less effective [13]). Since obtaining the gradients for the r parameters at each layer is inexpensive compared to the overall back-propagation, and we want to adapt at least the bottom layer, we apply LHUC to each layer for the rest of this work.

Fig. 5 (b) shows WERs for the number of adaptation iterations. The results indicate that one sweep over the adaptation data (in this case tst2010) is sufficient and, more importantly, the model does not overfit when adapting with more iterations (despite the adaptation objective consistently

TABLE II
WER(%) FOR DIFFERENT RE-PARAMETRISATION FUNCTIONS FOR LHUC TRANSFORMS ON TED tst2010. UNADAPTED BASELINE WER IS 15.0%.

r	$2/(1+\exp(-r))$	$\exp(r)$	$\max(0,r)$
12.8	12.8	12.7	12.7

improving – Fig. 5 (c)). This suggests that it is not necessary to carefully regularise the model – for example, by Kullback-Leibler divergence training [38] which is usually required when adapting the weights of one or more layers in a network.

Finally, we explored how the form of the LHUC re-parametrisation function ξ affects the WER and frame error rate (FER) (Fig. 5 (c) and Table II). For test-only adaptation only a small WER difference (0.1% absolute) is observed, regardless of the large difference in frame accuracies. This supports our previous observation that LHUC is robust against over-fitting. For SAT-LHUC training, a less constrained parametrisation was found to give better WERs for the SI model. Based on our control experiments, during SAT-LHUC training, setting ξ to be the identity function (linear r) gave similar results to $\xi(r) = \max(0, r)$ and $\xi(r) = \exp(r)$ and all were better than re-parametrising with $\xi(r) = 2/(1+\exp(-r))$. This is expected as for full training the last approach constrains the range of back-propagated gradients. From now on, if not stated otherwise, we will use $\xi(r) = \exp(r)$ in the remainder of this paper.

We adapt our all models with the learning rate set to 0.8 (regardless of $\xi(\cdot)$) and the basic training of both the SI and the SAT-LHUC models was performed with the initial learning rate set to 0.08 and was later adjusted according to the newbob learning scheme [70].

B. SAT-LHUC

As described in section IV, SAT-LHUC training aims to regularise the hidden unit feature receptors so that they capture not just the average characteristics of training data, but also specific features of the different distributions the data was drawn from (for example, different training speakers). As a result, the model can be better tailored to unseen speakers by putting more importance to those units that were useful for training speakers with similar characteristics.

Prior to SAT-LHUC training we need to decide on how and which data should be used to estimate speaker-dependent and speaker-independent transforms. In this work we train SAT-LHUC models with frame-level [14], segment-level and speaker-level clusters. For speaker- and segment-level transforms we decide which speakers or segments are going to be treated as SI or SD prior to training. For the frame-level SAT-LHUC approach, the SI/SD decisions are made separately for each data-point during training. In either scenario we ensure that the overall SD/SI ratio determined by γ parameter is satisfied. The WER results for each of these three approaches ($\gamma = 0.5$) are reported in Table III. Speaker-level SAT-LHUC training provides the highest WERs for both SI and SD decodes. Segment-level and frame-level SAT-LHUC training result in similar WERs for SI decodes, with a small advantage (0.1% abs.) for the frame-level approach after adaptation.

TABLE III
WER(%) FOR DIFFERENT SAMPLING STRATEGIES AND SAT-LHUC TRAINING (TED tst2013)

Model	Baseline	WER (%) for sampling strategies		
		Per Speaker	Per Segment	Per Frame
SI	22.1	23.0	22.0	22.0
SD	19.1	18.6	18.1	18.0

Fig. 6 gives more insight on how the ratio of SI and SD data (determined by γ) affects the WER of the first-pass and adapted systems on TED tst2013. The SI/SD split mainly affects the first pass accuracies with a substantial increase in SI WER when less than 30% of the data is used to estimate the SI LHUC transforms. However, once adapted, all variants obtained lower WERs compared to the baseline SI and LHUC adapted model. For instance, when $\gamma = 0.5$ the SAT-LHUC systems operating in SI mode obtained similar accuracies to the baseline SI model (22%WER); however, the adapted SAT-LHUC model gave around 1% absolute (6% relative) decrease in WER compared with the SI baseline test-only adapted LHUC model. The adaptation results for speaker-level SAT-LHUC training were worse by around 0.4% absolute compared to segment- or frame-level SAT-LHUC training. However, the difference, as shown experimentally in [14], is mostly due to poorer quality adaptation targets resulting from the corresponding first pass SAT-LHUC systems rather than the differences in learned representations. Managing a good trade-off between SI and SD ratios for SAT-LHUC is nevertheless an important aspect to take into account, and in our experience using around 50–60% of data for the SI transform is a good task-independent setting. If different models for SI and SD decodes are acceptable, then further small gains in accuracy are observed [14].

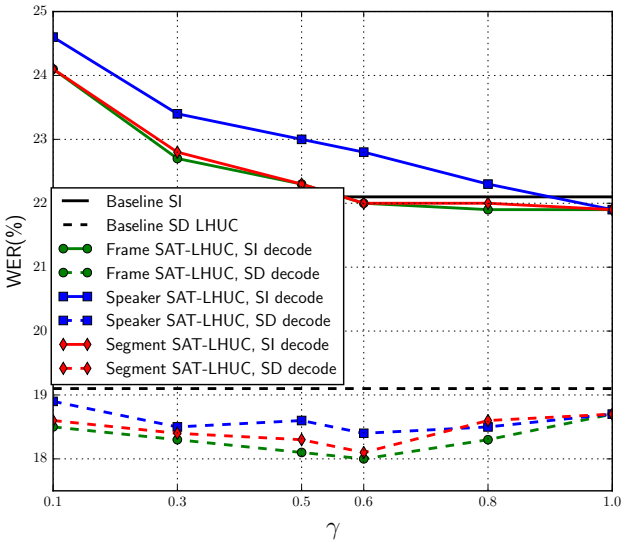
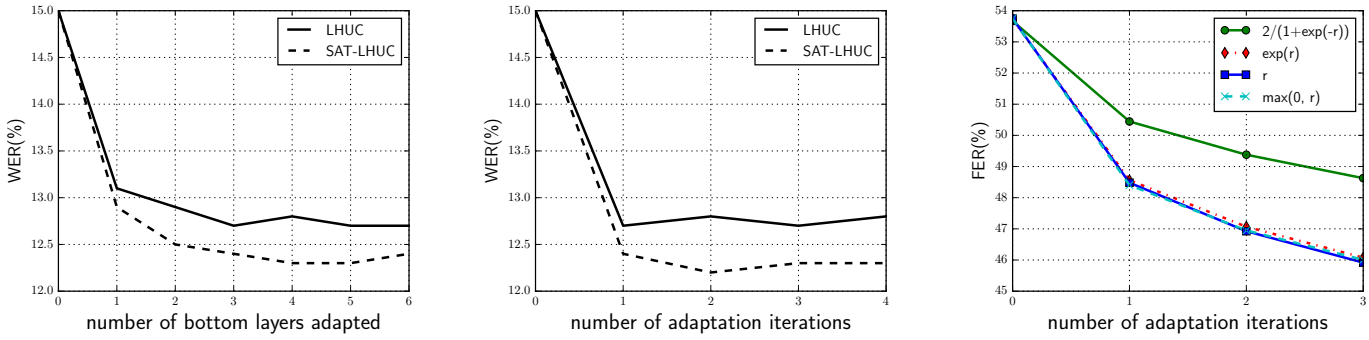


Fig. 6. WER(%) for different sampling strategies {per frame, per segment, per speaker} for SAT-LHUC training and SI and SD decodes on TED tst2013.

We report the baseline LHUC and SAT-LHUC comparisons on TED and AMI data in Tables IV and V, respectively



(a) (b) (c)

Fig. 5. WER(%) on TED tst2010 as a function of: a) number of adapted layers; and b) number of adaptation iterations; c) FER for re-parameterisation functions (ξ) used in adaptation.

TABLE IV
WER(%) ON TED TALKS (TST2010 AND TST2013).

System		IWSLT Test set	
Training	Decoding	tst2010	tst2013
Baseline speaker-independent systems			
SI	SI	15.0	22.1
SAT-LHUC	SI	15.1	22.0
Adapted systems			
SI	LHUC	12.7	19.1
SAT-LHUC	LHUC	12.4	18.0

TABLE V
WER(%) ON AMI–IHM

Model	Features	dev	eval
DNN	FBANK	26.8	29.1
+LHUC	FBANK	25.6	27.1
+SAT-LHUC	FBANK	24.9	26.1
CNN	FBANK	25.2	27.1
+LHUC	FBANK	24.3	25.3
+SAT-LHUC	FBANK	23.9	24.8

(further results, including a comparison to fMLLR transforms and on Switchboard data are in the next sections). On TED (Table IV), SAT-LHUC models operating in SI mode ($\gamma = 0.6$) have comparable WERs to SI models; however, adaptation resulted in a WER reduction of 0.3–1.1% absolute (2–6% relative) compared to test-only adaptation of the SI models. Similar results were observed on the AMI data (Table V) where for both DNN and CNN models trained on FBANK features LHUC adaptation decreased the WER by 2% absolute (7% relative) and SAT-LHUC training improved this result by 4% relative for DNN models. As expected, the SAT-LHUC gain for CNNs was smaller when compared to DNN models, since the CNN layer can learn different patterns for different speakers which may be selected through the max-pooling operator at run-time.

C. Sequence model adaptation

Model-based adaptation of sequence-trained DNNs (SE-DNN) is more challenging compared to adapting networks trained using cross-entropy: a mismatched adaptation objective (here cross-entropy) can easily erase sequence information from the weight matrices due to the well-known effect of catastrophic forgetting [71] in neural networks. Indeed Huang and Gong [39] report no gain from adapting SE-DNN models with cross-entropy adaptation objective and supervised adaptation targets. In those experiments, all weights in the model were updated and one needs to perform KL divergence regularised adaptation [38] or KL regularised sequence level adaptation to further improve on top of SE-DNN. It remains to be answered if one can get similar improvements using SE-DNN adaptation and first-pass transcripts.

TABLE VI
SUMMARY OF WER RESULTS OF LHUC ADAPTED SEQUENCE MODELS ON TED TST2011 AND TST2013

Model	tst2011	tst2013
DNN-CE	12.1	22.1
DNN-sMBR	10.3	20.2
+LHUC	9.5	18.0
+fMLLR	9.6	18.9
++LHUC	8.9	15.8

In this work we adapt state-level minimum Bayes risk (sMBR) [72], [73] sequence-trained models using LHUC and report results on TED tst2011 and tst2013 in Table VI. We kept all the LHUC adaptation hyper-parameters the same as for CE models and obtained around 2% absolute (11% relative) WER reductions on tst2013 for both SI and fMLLR SAT adapted SE-DNN systems. Interestingly, the obtained adaptation gain was similar to the cross-entropy models and LHUC adaptation did not seem to disrupt the learned model's sequence representation.

We compared our adaptation results to the most accurate system of the IWSLT–2013 TED transcription evaluation, which performed both feature- and model-space speaker adaptation [74]. For model-space adaptation that system used a method which adapts DNNs with a speaker-dependent layer [42]. The results are reported in Table VII where in the first block one can see a standard sequence-trained feature-space adapted system build from TED and 150 hours of out-of-domain data scoring 15.7% WER, similar to the WER of our TED system (15.4%), which also for IWSLT utilised 100 hours of out-of-domain AMI data. The 0.3% difference could

TABLE VII
WERs FOR ADAPTED SEQUENCE-TRAINED MODELS USED IN IWSLT EVALUATION. NOTE, THE RESULTS ARE NOT DIRECTLY COMPARABLE TO THOSE REPORTED ON TED IN TABLE VI DUE DIFFERENT TRAINING DATA AND FEATURE PRE-PROCESSING PIPELINES (SEE REFERENCED PAPERS FOR SYSTEM DETAILS).

Model	tst2011	tst2013
IWSLT2013 winner system (numbers taken from [74])		
DNN (sMBR) + HUB4 + WSJ	-	15.7
+ Six ROVER subsystems	-	14.8
++ Automatic segmentation	-	14.3
+++ LM adapt. + RNN resc.	-	14.1
+++++ SAT on DNN [42]	7.7	13.5
Our system [62]		
DNN (sMBR) + AMI data	9.0	15.4
+LHUC	8.5	13.3

be explained by characteristics of the out-of-domain data used (tst2013 is characterised by a large proportion of non-native speakers which is also typical for AMI data, hence benefits more our baseline systems). When comparing both adaptation approaches operating in an unsupervised manner one can see that LHUC gives much bigger improvements in WER compared to speaker-dependent layer, 2.1% vs. 0.6% absolute (14% vs. 4% relative) on tst2013. This allows our single-model system to match a considerably more sophisticated post-processing pipeline [74], as outlined in Table VII. For less mismatched data (tst2011) adaptation is less important and our system has a WER 0.8% absolute higher compared with the more sophisticated system.

From these experiments we conclude that LHUC is an effective way to adapt sequence models in an unsupervised manner using a cross-entropy objective function, without the risk of removing learned sequence information.

D. Other aspects of adaptation

Amount of adaptation data: Fig 7 shows the effect of the amount of adaptation data on WER for LHUC and SAT-LHUC adapted models. As little as 10s of unsupervised adaptation data is already able to substantially decrease WERs (by 0.5–0.8% absolute). The improvement for SAT-LHUC adaptation compared with LHUC is considerably larger – roughly by a factor of two up to 30s adaptation data. As the duration of adaptation data increases the difference gets smaller; however SAT-LHUC results in consistently lower WERs than LHUC in all cases (including full two pass adaptation).

We also investigated supervised (oracle) adaptation by aligning the acoustics with the reference transcriptions (dashed lines). Given supervised adaptation targets, LHUC and SAT-LHUC further substantially decrease WERs, with SAT-LHUC giving a consistent advantage over LHUC.

Quality of adaptation targets: Since our approach relies on a first-pass decoding, we investigated the extent to which LHUC is sensitive to the quality of the adaptation targets. In this experiment we explored the differences resulting from different language models, and assumed that the first pass adaptation data was generated by either an SI or a SAT-LHUC model operating in SI mode. The main results are shown

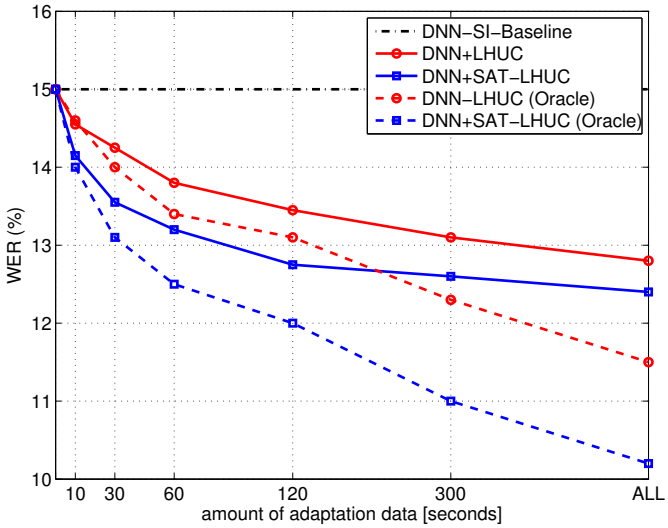


Fig. 7. WER(%) for unsupervised and oracle adaptation data on TED tst2010.

in Fig 8 where the solid lines show WERs obtained with a pruned 3-gram LM and different types of adaptation targets resulting from re-scoring the adaptation data with stronger LMs. One can see there is not much difference unless the adaptation data was re-scored with the largest 4-gram LM. This improvement diminishes in the final adapted system after re-scoring. This suggests that the technique is not very sensitive to the quality of adaptation targets. This trend holds regardless of the amount of data used for adaptation (ranging from 10s to several minutes per speaker). In related work [32] LHUC was employed using alignments obtained from an SI-GMM system with a 8.1% absolute higher WER than the corresponding SI DNN, and substantial gains were obtained over the unadapted SI DNN baseline – although the WER reduction was considerably smaller (1% absolute) compared to adaptation with alignments obtained with the corresponding SI DNN.

Quality of data: We also investigated how the quality of the acoustic data itself affects the adaptation accuracies, keeping the other ASR components fixed. We performed an experiment on the AMI corpus using speech captured by individual headset microphones (IHM) and a single distant tabletop microphone (SDM). In case of IHM we adapt to the headset; in this experiment we assume we have speaker labels for the SDM data[2]. The results are reported in Table VIII: LHUC adaptation improves the accuracy in both experiments, although the gain for the SDM condition is smaller; however, the SDM system is characterised by twice as large WERs. Notice that LHUC has also been successfully applied to channel normalisation between distant and close talking microphones [75].

One-shot adaptation: By one-shot adaptation we mean the scenario in which LHUC transforms were estimated once for a held-out speaker and then used many times in a single pass

[2]In a real scenario for SDM data one would have to perform speaker diarisation in order to obtain speaker labels.

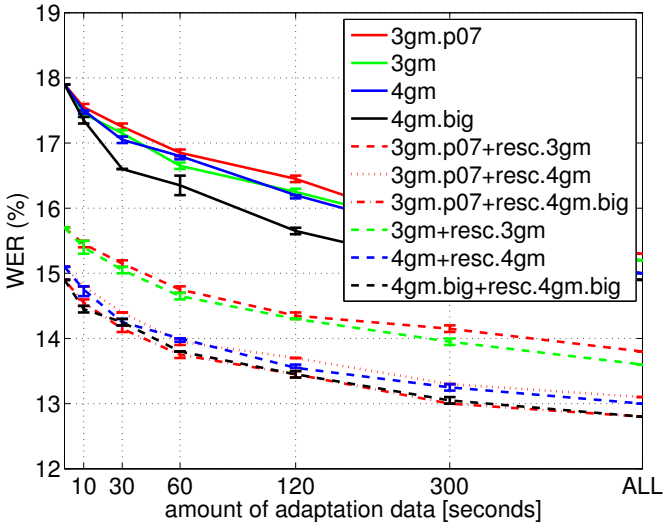


Fig. 8. WER(%) for different qualities of adaptation targets on TED tst2010.

TABLE VIII
WER(%) ON AMI–IHM AND AMI–SDM USING ADAPTED CNNS.

Model	dev	eval
CNN (IHM)	25.2	27.1
+LHUC	24.3	25.3
CNN (SDM)	49.8	54.4
+LHUC	48.8	53.1

system for this speaker. We performed those experiments on AMI IHM data, and report results on dev and eval which contain 21 and 16 unique speakers taking part in 18 and 16 different meetings, respectively. Each speaker participates in multiple meetings: to some degree, adapting to a speaker in one meeting, then applying the adaptation transform to the same speaker in the other meetings simulates a real-life condition where it is possible to assume the speaker identity without having to perform speaker diarisation (e.g. personal devices). The results of this experiment (Table IX) indicate that LHUC retains the accuracies of two-pass systems by providing almost identical results when comparing LHUC estimated in a full two-pass system and when the unsupervised transforms are re-used in the LHUC.one-shot experiment.

E. Complementarity to feature normalisation

Feature-space adaptation using fMLLR is a very reliable current form of speaker adaptation, so it is of great interest to explore how complementary the proposed approaches are to SAT training with fMLLR transforms.[3]

We compared LHUC and SAT-LHUC to SAT-fMLLR training using TED tst2010 (Fig 9, red curves). We also compared both techniques, including a comparison in terms of the amount of data used to estimate each type of transform.

[3]Due to space constraints we do not make an explicit comparisons to other techniques such as auxiliary i-vector features or speaker-codes; however, the literature suggest that the use of i-vectors give similar [29] results when compared to fMLLR trained models. Related recent studies also show LHUC is at least as good as the standard use of i-vector features [32], [76].

TABLE IX
WER(%) ON AMI–IHM AND ONE-SHOT ADAPTATION

Model	dev	eval
CNN	25.2	27.1
+LHUC	24.3	25.3
+LHUC.one-shot	24.3	25.4

fMLLR transforms estimated on 10s of unsupervised data result in an increase in WER compared with the SI-trained baseline (16.1% vs. 15.0%). When combined with LHUC or SAT-LHUC some of this deterioration was recovered (similar results using LHUC alone were reported in Fig 7). For more adaptation data (30s or more) fMLLR improved the accuracies by around 1–2% absolute and combination with LHUC (or SAT-LHUC) resulted in an additional 1% reduction in WER (see also Table X in the next section for further results).

We also investigated (in a rather unrealistic experiment) how much mismatch in feature space one can normalise in model space with LHUC. To do so, we used a SAT-fMLLR trained model with unadapted PLP features which gave a large increase in WER (26% vs 15%). Then, using unsupervised adaptation targets obtained from the feature-mismatched decoding both LHUC and SAT-LHUC were applied. The results (also presented in Fig. 9) indicate that a very large portion of the WER increase can be effectively compensated in model space – more than 8% absolute. As found before, test-only reparametrisation functions ($\exp(r)$ vs. $2/(1+\exp(-r))$) have negligible impact on the adaptation results, and SAT-LHUC again provides better results.

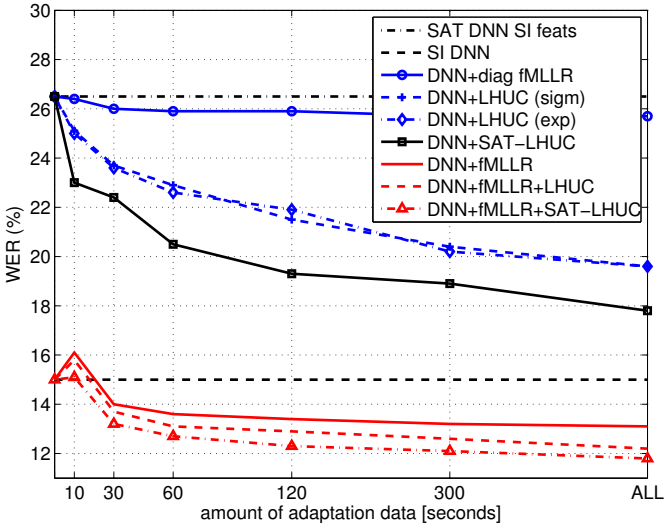


Fig. 9. WER(%) for LHUC, SAT-LHUC, and SAT-fMLLR (and combinations) on TED tst2010.

F. Adaptation Summary

In this section we summarise our results, applying LHUC and SAT-LHUC to TED, AMI, and Switchboard. Table X contains results for four IWSLT test sets (dev2010, tst2010, tst2011, and tst2013): in most scenarios SAT-LHUC

TABLE X
WER (%) ON VARIOUS TED DEVELOPMENT AND TEST SETS FROM IWSLT12 AND IWSLT13 EVALUATIONS.

Model	dev2010	tst2010	tst2011	tst2013
DNN	15.4	15.0	12.1	22.1
+LHUC	14.5	12.8	10.9	19.1
+SAT-LHUC	14.0	12.4	10.9	18.0
+fMLLR	14.5	12.9	10.9	20.8
++LHUC	14.1	11.8	10.3	18.4
++SAT-LHUC	13.7	11.6	9.9	17.6

TABLE XI
WER(%) ON AMI–IHM

Model	Features	dev	eval
DNN	FMLLR	26.2	27.3
+LHUC	FMLLR	25.6	26.2
DNN	FBANK	26.8	29.1
+LHUC	FBANK	25.6	27.1
+SAT-LHUC	FBANK	24.9	26.1

TABLE XII
WER(%) ON SWITCHBOARD EVAL2000.

	eval2000		
Model	SWB	CHE	TOTAL
DNN	15.2	28.2	21.7
+LHUC	14.7	26.6	20.7
++SAT-LHUC	14.6	25.9	20.3
+fMLLR	14.2	26.2	20.2
++LHUC	14.2	25.6	19.9
++SAT-LHUC	14.1	25.6	19.9

TABLE XIII
RESULTS ON AURORA 4. MULTI-CONDITION DNN MODEL.

Model	A	B	C	D	AVG
DNN	5.1	9.3	9.3	20.8	13.9
DNN + $\mathbf{r}_S$	4.3	9.3	6.9	19.3	13.1
DNN + $\mathbf{r}_E$	5.0	9.0	8.5	19.8	13.3
DNN + $\mathbf{r}_{SE\ JOINT}$	4.5	8.6	7.4	18.3	12.4
DNN + $\hat{\mathbf{r}}_{SE}$, $\alpha = 0.5$	4.6	8.9	7.7	19.1	12.9
DNN + $\hat{\mathbf{r}}_{SE}$, $\alpha = 0.7$	4.5	8.8	7.2	18.9	**12.7**

results in a lower WER than LHUC and both techniques are complementary with SAT-fMLLR training.

Similar conclusions can be drawn from experiments on AMI (Table XI) where LHUC and SAT-LHUC were found to effectively adapt DNN and CNN models trained on FBANK features. SAT-LHUC trained DNN models gave the same final results as the more complicated SAT-fMLLR+LHUC system.

On Switchboard, in contrast to other corpora, we observed that test-only LHUC does not match the WERs obtained from SAT-fMLLR models (Table XII). The SI system has a WER of 21.7% compared with 20.7% for the test-only LHUC and 20.2% for the SAT-fMLLR system. The improvement obtained using test-only LHUC is comparable to that obtained with other test-only adaptation techniques, e.g. feature-space discriminative linear regression (fDLR) [4], but neither of these matches SAT trained feature transform models. This could be due to the fact Switchboard data is narrow-band and as such contains less information for discrimination between speakers [77], especially when estimating relevant statistics from small amounts of unsupervised adaptation data. Another potential reason could be related to the fact that the Switchboard part of eval2000 is characterised by a large overlap between training and test speakers – 36 out of 40 test speakers are observed in training [78], which limits the need for adaptation, but also enables models to learn much more accurate speaker-characteristics during supervised speaker adaptive training.

Adaptation using SAT-LHUC (20.3% WER) almost matches SAT-fMLLR (20.2%). We also observe that LHUC performs relatively better under more mismatched conditions (the Callhome (CHE) subset of eval2000), similar to what we observed on TED.

Finally, in Fig 10 we show the WERs obtained for 200 speakers across the TED, AMI, and SWBD test sets. We observe that for 89% of speakers LHUC and SAT-LHUC adaptation reduced the WER, and that SAT-LHUC gives a consistent reduction over LHUC.

VII. LHUC FOR FACTORISATION

We applied LHUC to adapt to both the speaker and the acoustic environment. If multi-condition data is available for a speaker, then it is possible to define a set of joint speaker-environment LHUC transforms. Alternatively, we can estimate two set of transforms – for speaker $\mathbf{r}_S$ and for environment $\mathbf{r}_E$ – and then linearly interpolate them, using hyper-parameter α, to derive a combined transform $\hat{\mathbf{r}}_{SE}$ as follows:

$$\xi\left(\hat{\mathbf{r}}^l_{SE}\right) = \alpha\xi\left(\mathbf{r}^l_S\right) + (1-\alpha)\xi\left(\mathbf{r}^l_E\right) \tag{10}$$

Notice, that although both types of transforms are estimated in an unsupervised manner we assume that the test environment is known, allowing the correct environmental transform to be selected. This adaptation to the test environment is similar to that of Li et al [79].

We adapted baseline multi-condition trained DNN models [80] to the speaker ($\mathbf{r}_S$) and the environment ($\mathbf{r}_E$). The $\mathbf{r}_S$ transforms were estimated only on *clean* speech; similarly the environment transforms were estimated for each scenario (one set of $\mathbf{r}_E$ per scenario) using multiple speakers (hence, we have 7 different environmental transforms). To avoid learning joint speaker-environment transforms the target speaker's data was removed from environment adaptation material (e.g. when estimating transforms for the restaurant environment, we use all restaurant data except the one for the target speaker).

The results (Table XIII) show that both standalone speaker or environment adaptation LHUC adaptation improve over an unadapted system (13.1%(S) and 13.3%(E) vs. 13.9%) but, as expected, a single transform estimated jointly on the target speaker and environment has a lower WER (12.4%). However, when interpolated with $\alpha = 0.7$ the result of the factorised model improves to 12.7% WER, although still higher that joint estimation. However, adaptation data for joint speaker-environment adaptation is not available in many scenarios, and the factorised adaptation based on interpolation is more flexible.

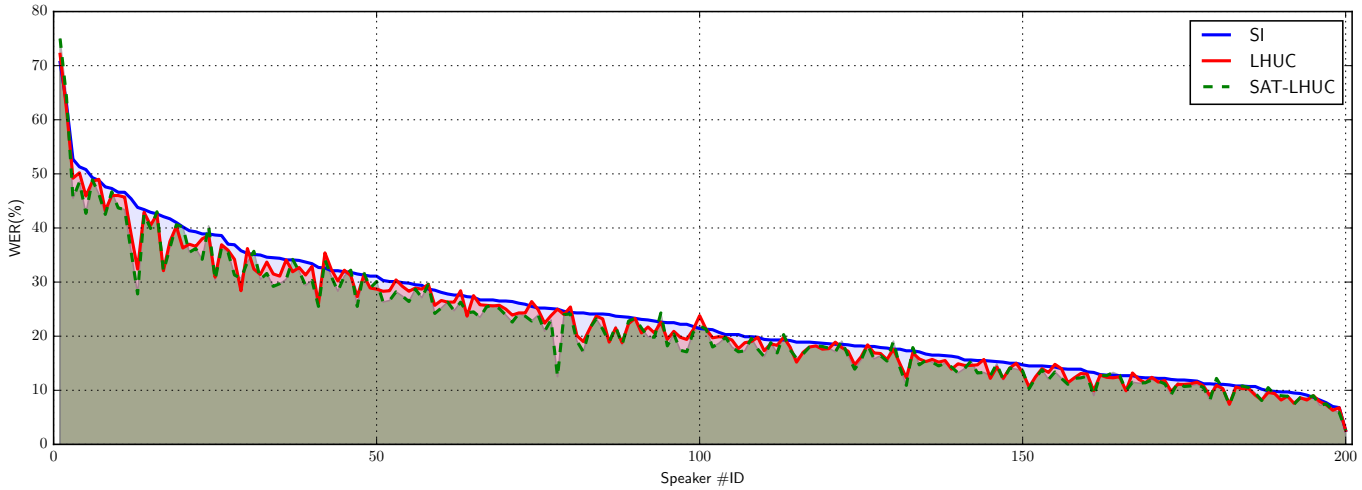


Fig. 10. Summary of WERs(%) obtained with LHUC and SAT−LHUC adaptation techniques on test speakers of TED, SWBD and AMI corpora (results are sorted in descending WER order for the SI system). For LHUC the average observed improvement per speaker was at 1.6% absolute (7.0% relative). The same statistic for SAT−LHUC was at 2.3% absolute (9.7% relative). The maximum observed WER decrease per speaker was 11.4% absolute (32.7% relative) and 16.0% absolute (50% relative) for LHUC and SAT−LHUC, respectively. WERs decreased for 89% of speakers using LHUC adaptation.

TABLE XIV
RESULTS ON AURORA 4. MULTI-CONDITION MAXOUT-CNN MODEL, WITH AND WITHOUT ANNEALED DROPOUT (AD).

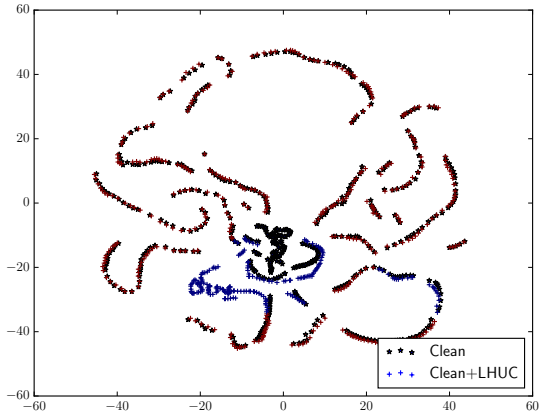
Model	A	B	C	D	AVG
MaxCNN	4.2	7.7	7.9	17.4	11.6
MaxCNN + $\mathbf{r}_{SE\ JOINT}$	3.7	6.3	5.5	14.3	9.5
AD MaxCNN	4.3	7.7	7.2	15.6	10.9
AD MaxCNN + $\mathbf{r}_{SE\ JOINT}$	3.4	5.7	6.1	13.4	8.7

We also trained more competitive models following Rennie et al [81]: Maxout [82] CNN models were trained using annealed dropout [83]. In this work we used alignments obtained by aligning a corresponding multi-condition model as ground-truth labels, rather than replicating clean alignments to multi-condition data, in contrast to [81]: this is likely to explain differences in the reported baselines (10.9% compared with 10.5% in [81]). The results for the joint optimisation are reported in Table XIV where one can notice large improvements with unsupervised LHUC adaptation.

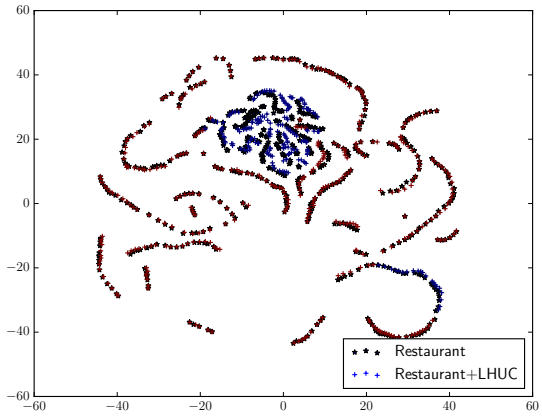
Finally, we visualise the top hidden layer activations of the annealed dropout Maxout CNN using stochastic neighbourhood embedding (tSNE) [84] for one utterance recorded under clean and noisy (restaurant) conditions (Fig. 11).

VIII. CONCLUSIONS

We have presented the LHUC approach to unsupervised adaptation of neural network acoustic models in both test-only (LHUC) and SAT (SAT−LHUC) frameworks, evaluating them using four standard speech recognition corpora: TED talks as used in the IWSLT evaluations, AMI, Switchboard, and Aurora4. Our experimental results indicate that both LHUC and SAT−LHUC can provide significant improvements in word error rates (5–23% relative depending on test set and task). LHUC adaptation works well unsupervised and with small amounts of data (as little as 10s), is complementary to



(a)



(b)

Fig. 11. tSNE plots (best viewed in color) of the top hidden layer before and after adaptation for an utterance recorded in (a) clean and (b) noisy (restaurant) environment, using the annealed dropout maxout CNN. The model can normalise the phonetic space between conditions (brown color), keeping two different spaces for non-speech frames (blue color) under clean and noisy conditions. The effect of LHUC is mostly visible for non-speech frames.

feature space normalisation transforms such as SAT-fMLLR, and can be used for unsupervised adaptation of sequence-trained DNN acoustic models using a cross-entropy adaptation objective function. Furthermore we have demonstrated that it can be applied in a factorised way, estimating and interpolating separate transforms for adaptation to the acoustic environment and speaker.

REFERENCES

[1] G Hinton, L Deng, D Yu, GE Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, TN Sainath, and B Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[2] H Bourlard and N Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[3] S Renals, N Morgan, H Bourlard, M Cohen, and H Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Trans Speech and Audio Processing*, vol. 2, pp. 161–174, 1994.

[4] F Seide, X Chen, and D Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE ASRU*, 2011.

[5] GE Dahl, D Yu, L Deng, and A Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[6] H Hermansky, DPW Ellis, and S Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. IEEE ICASSP*, 2000, pp. 1635–1638.

[7] F Grezl, M Karafiat, S Kontar, and J Cernocky, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proc. IEEE ICASSP*, 2007, pp. IV–757–IV–760.

[8] TN Sainath, A Mohamed, B Kingsbury, and B Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE ICASSP*, 2013.

[9] P Swietojanski, A Ghoshal, and S Renals, "Revisiting hybrid and GMM-HMM system combination techniques," in *Proc. IEEE ICASSP*, 2013.

[10] PC Woodland, X Liu, Y Qian, C Zhang, MJF Gales, P Karanasou, P Lanchantin, and L Wang, "Cambridge University transcription systems for the Multi-Genre Broadcast Challenge," in *Proc. IEEE ASRU*, 2015.

[11] PC Woodland, "Speaker adaptation for continuous density HMMs: A review," in *Proceedings of the ISCA workshop on adaptation methods for speech recognition*, 2001, pp. 11–19.

[12] O Abdel-Hamid and H Jiang, "Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition.," in *Proc. Interspeech*, pp. 1248–1252.

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

[14] P Swietojanski and S Renals, "SAT-LHUC: Speaker adaptive training for learning hidden unit contributions," in *Proc. IEEE ICASSP*, 2016.

[15] M Cettolo, C Girardi, and M Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proc. EAMT*, 2012, pp. 261–268.

[16] 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.

[17] JJ Godfrey, EC Holliman, and J McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. IEEE ICASSP*. IEEE, 1992, pp. 517–520.

[18] N Parihar, J Picone, D Pearce, and HG Hirsch, "Performance analysis of the Aurora large vocabulary baseline system," in *Proc. EUSIPCO*, 2004.

[19] MJF Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, April 1998.

[20] A Mohamed, TN Sainath, G Dahl, B Ramabhadran, GE Hinton, and MA Picheny, "Deep belief networks using discriminative features for phone recognition," in *Proc. IEEE ICASSP*, 2011, pp. 5060–5063.

[21] T Hain, L Burget, J Dines, PN Garner, F Grézl, A El Hannani, M Karafíat, M Lincoln, and V Wan, "Transcribing meetings with the AMIDA systems," *IEEE Trans Audio, Speech and Language Processing*, vol. 20, pp. 486–498, 2012.

[22] TN Sainath, B Kingsbury, and B Ramabhadran, "Auto-encoder bottleneck features using deep belief networks.," in *Proc. IEEE ICASSP*, 2012, pp. 4153–4156.

[23] P Bell, P Swietojanski, and S Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *Proc. IEEE ICASSP*, 2013.

[24] 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 *Proc. Eurospeech*, 1995, pp. 2171–2174.

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

[26] B Li, , and KC Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems," in *Proc. Interspeech*, 2010.

[27] N Dehak, PJ Kenny, R Dehak, P Dumouchel, and P Ouellet, "Front end factor analysis for speaker verification," *IEEE Trans Audio, Speech and Language Processing*, vol. 19, pp. 788–798, 2010.

[28] M Karafiat, L Burget, P Matejka, O Glembek, and J Cernozky, "iVector-based discriminative adaptation for automatic speech recognition," in *Proc. IEEE ASRU*, 2011.

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

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

[31] P Karanasou, Y Wang, MJF Gales, and PC Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *Proc. Interspeech*, 2014.

[32] Y Miao, H Zhang, and F Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 1938–1949, Nov 2015.

[33] Y Liu, P Zhang, and T Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *Proc. IEEE ICASSP*, 2014.

[34] JS Bridle and S Cox, "Recnorm: Simultaneous normalisation and classification applied to speech recognition," in *Advances in Neural Information Processing Systems 3*, 1990, pp. 234–240.

[35] O Abdel-Hamid and H Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proc. IEEE ICASSP*, 2013, pp. 4277–4280.

[36] S Xue, O Abdel-Hamid, J Hui, L Dai, and Q Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1713–1725, Dec 2014.

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

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

[39] Y Huang and Y Gong, "Regularized sequence-level deep neural network model adaptation," in *Proc. Interspeech*, 2015.

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

[41] L Samarakoon and K C Sim, "Learning factorized feature transforms for speaker normalization," in *Proc. IEEE ASRU*. IEEE, 2015, pp. 145–152.

[42] T Ochiai, S Matsuda, X Lu, C Hori, and S Katagiri, "Speaker adaptive training using deep neural networks," in *Proc. IEEE ICASSP*, 2014.

[43] K Yao, D Yu, F Seide, H Su, L Deng, and Y Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition.," in *Proc. IEEE SLT*, 2012.

[44] Y Zhao, J Li, J Xue, and Y Gong, "Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data," in *Proc. IEEE ICASSP*, 2015.

[45] P Swietojanski and S Renals, "Differentiable pooling for unsupervised speaker adaptation," in *Proc. IEEE ICASSP*, 2015.

[46] SM Siniscalchi, J Li, and CH Lee, "Hermitian polynomial for speaker adaptation of connectionist speech recognition systems," *IEEE Trans Audio, Speech, and Language Processing*, vol. 21, pp. 2152–2161, 2013.

[47] Z Huang, S M Siniscalchi, I-F Chen, J Wu, and C-H Lee, "Maximum a posteriori adaptation of network parameters in deep models," *arXiv preprint arXiv:1503.02108*, 2015.

[48] Z Huang, J Li, S M Siniscalchi, I-F Chen, J Wu, and C-H Lee, "Rapid adaptation for deep neural networks through multi-task learning," in *Proc. Interspeech*, 2015.

[49] P Swietojanski, P Bell, and S Renals, "Structured output layer with auxiliary targets for context-dependent acoustic modelling," in *Proc. Interspeech*, 2015.

[50] R Price, K Iso, and K Shinoda, "Speaker adaptation of deep neural networks using a hierarchy of output layers," in *Proc. IEEE SLT*, 2014.

[51] K Hornik, M Stinchcombe, and H White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[52] K Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251 – 257, 1991.

[53] AR Barron, "Universal approximation bounds for superpositions of a sigmoidal function.," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 930–945, 1993.

[54] C Zhang and PC Woodland, "Parameterised Sigmoid and ReLU Hidden Activation Functions for DNN Acoustic Modelling," in *Proc. Interspeech*, 2015.

[55] E Trentin, "Networks with trainable amplitude of activation functions," *Neural Networks*, vol. 14, pp. 471–493, 2001.

[56] C Wu and M Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *Proc. IEEE ICASSP*, 2015.

[57] T Tan, Y Qian, M Yin, Y Zhuang, and K Yu, "Cluster adaptive training for deep neural network," in *Proc. IEEE ICASSP*, 2015.

[58] M Delcroix, K Kinoshita, T Hori, and T Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation," in *Proc. IEEE ICASSP*, 2015.

[59] T Anastasakos, J McDonough, R Schwartz, and J Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, 1996, pp. 1137–1140.

[60] S Renals, T Hain, and H Bourlard, "Recognition and understanding of meetings: The AMI and AMIDA projects," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU'07*, Kyoto, 12 2007, IDIAP-RR 07-46.

[61] M Federico, M Cettolo, L Bentivogli, M Paul, and S Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. IWSLT*, 2012.

[62] P Bell, P Swietojanski, J Driesen, M Sinclair, F McInnes, and S Renals, "The UEDIN ASR Systems for the IWSLT 2014 Evaluation," in *Proc. IWSLT*, 2014.

[63] P Swietojanski, A Ghoshal, and S Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Proc. IEEE ASRU*, 2013.

[64] TN Sainath, B Kingsbury, A Mohamed, GE Dahl, G Saon, H Soltau, T Beran, AY Aravkin, and B Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *Proc. IEEE ASRU*, 2013, pp. 315–320.

[65] Y. LeCun, L. Bottou, G. Orr, and K. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, chapter 2. Springer, 1998.

[66] O Abdel-Hamid, A-R Mohamed, J Hui, and G Penn, "Applying convolutional neural networks concepts to hybrid NN–HMM model for speech recognition.," in *Proc. IEEE ICASSP*, 2012, pp. 4277–4280.

[67] P Swietojanski, A Ghoshal, and S Renals, "Convolutional neural networks for distant speech recognition," *Signal Processing Letters, IEEE*, vol. 21, no. 9, pp. 1120–1124, Sept. 2014.

[68] K Vesely, A Ghoshal, L Burget, and D Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, Lyon, France, August 2013.

[69] D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlíček, Y Qian, P Schwarz, J Silovský, G Stemmer, and K Veselý, "The Kaldi speech recognition toolkit," in *Proc. IEEE ASRU*, December 2011.

[70] S Renals, N Morgan, M Cohen, and H Franco, "Connectionist probability estimation in the DECIPHER speech recognition system," in *Proc. IEEE ICASSP*, 1992.

[71] RM French, "Catastrophic forgetting in connectionist networks: Causes, consequences and solutions," *Trends in Cognitive Sciences*, vol. 3, pp. 128–135, 1999.

[72] J Kaiser, B Horvat, and Z Kacic, "A novel loss function for the overall risk criterion based discriminative training of HMM models," in *Proc. ICSLP*, 2000, pp. 887–890.

[73] B Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. IEEE ICASSP*, 2009, pp. 3761–3764.

[74] C-L Huang, PR Dixon, S Matsuda, Y Wu, X Lu, M Saiko, and C Hori, "The NICT ASR system for IWSLT 2013," in *Proc. IWSLT*, 2013.

[75] I Himawan, P Motlicek, M Ferras, and S Madikeri, "Towards utterance-based neural network adaptation in acoustic modeling," in *Proc. IEEE ASRU*, 2015.

[76] L Samarakoon and K C Sim, "On combining i-vectors and discriminative adaptation methods for unsupervised speaker normalization in dnn acoustic models," in *Proc. IEEE ICASSP*, 2016.

[77] M Wester, Z Wu, and J Yamagishi, "Human vs machine spoofing detection on wideband and narrowband data," in *Proc. Interspeech*, 2015.

[78] J Fiscus, WM Fisher, AF Martin, MA Przybocki, and DS Pallett, "2000 NIST evaluation of conversational speech recognition over the telephone: English and Mandarin performance results," in *Proc. Speech Transcription Workshop*. Citeseer, 2000.

[79] J Li, J-T Huang, and Y Gong, "Factorized adaptation for deep neural network," in *Proc. IEEE ICASSP*, 2014.

[80] M Seltzer, D Yu, and Y Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE ICASSP*, 2013.

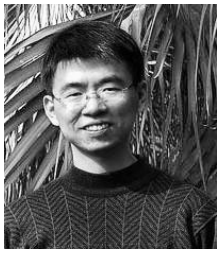
[81] SJ Rennie, V Goel, and S Thomas, "Annealed dropout training of deep networks," in *Proc. IEEE SLT*, 2014.

[82] IJ Goodfellow, D Warde-Farley, M Mirza, A Courville, and Y Bengio, "Maxout networks," *arXiv:1302.4389*, 2013.

[83] N Srivastava, G Hinton, A Krizhevsky, I Sutskever, and R Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[84] LJP van der Maaten and GE Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9: 25792605, Nov 2008.

Pawel Swietojanski received his M.Sc. degree from AGH University of Science and Technology in Cracow, Poland and is now a Ph.D. candidate in Informatics at the Centre for Speech Technology Research, School of Informatics, University of Edinburgh, UK. His main research interests are in machine learning and its applications to speech processing, with a particular focus on learning representations for acoustic modelling in speech recognition.



Jinyu Li (M'08) received the Ph.D. degree from Georgia Institute of Technology, U.S. From 2000 to 2003, he was a Researcher at Intel China Research Center and a Research Manager at iFlytek, China. Currently, he is a Principal Applied Scientist at Microsoft, working as a technical lead to design and improve speech modeling algorithms and technologies that ensure industry state-of-the-art speech recognition accuracy for Microsoft products. His major research interests cover several topics in speech recognition and machine learning, including noise robustness, deep learning, discriminative training, and feature extraction. He has authored one book and over 60 papers, and awarded over 10 patents.

Steve Renals (M'91 — SM'11 – F'14) is professor of speech technology at the University of Edinburgh. He received a BSc from the University of Sheffield and an MSc and PhD from Edinburgh. He has previously had positions at ICSI Berkeley, the University of Cambridge, and the University of Sheffield. His research interests are in speech and language processing.