Bayesian Learning of LF-MMI Trained Time Delay Neural Networks for Speech Recognition

Shoukang Hu , *Graduate Student Member, IEEE*, Xurong Xie, Shansong Liu , Jianwei Yu , Zi Ye,
Mengzhe Geng, Xunying Liu , *Member, IEEE*, and Helen Meng, *Fellow, IEEE*

Abstract—Discriminative training techniques define state-of-the-art performance for automatic speech recognition systems. However, they are inherently prone to overfitting, leading to poor generalization performance when using limited training data. In order to address this issue, this paper presents a full Bayesian framework to account for model uncertainty in sequence discriminative training of factored TDNN acoustic models. Several Bayesian learning based TDNN variant systems are proposed to model the uncertainty over weight parameters and choices of hidden activation functions, or the hidden layer outputs. Efficient variational inference approaches using as few as one single parameter sample ensure their computational cost in both training and evaluation time comparable to that of the baseline TDNN systems. Statistically significant word error rate (WER) reductions of 0.4%–1.8% absolute (5%-11% relative) were obtained over a state-of-the-art 900 h speed perturbed Switchboard corpus trained baseline LF-MMI factored TDNN system using multiple regularization methods including F-smoothing, L2 norm penalty, natural gradient, model averaging and dropout, in addition to i-Vector plus learning hidden unit contribution (LHUC) based speaker adaptation and RNNLM rescoring. The efficacy of the proposed Bayesian techniques is further demonstrated in a comparison against the state-of-the-art performance obtained on the same task using the most recent hybrid and end-to-end systems reported in the literature. Consistent performance improvements were also obtained on a 450-h HKUST conversational Mandarin telephone speech recognition task. On a third cross domain adaptation task requiring rapidly porting a 1000-h LibriSpeech data trained system to a small DementiaBank elderly speech corpus, the proposed Bayesian TDNN LF-MMI systems outperformed the baseline system using direct weight fine-tuning by up to 2.5% absolute WER reduction.

Index Terms—Bayesian learning, domain adaptation, gaussian process, LF-MMI, variational inference.

Shoukang Hu, Shansong Liu, Jianwei Yu, Zi Ye, Mengzhe Geng, Xunying Liu, and Helen Meng are with the Department of Systems Engineering and Engineering Management, Chinese University of Hong Kong, Hong Kong SAR 999077, China (e-mail: skhu@se.cuhk.edu.hk; ssliu@se.cuhk.edu.hk; jwyu@se.cuhk.edu.hk; zye@se.cuhk.edu.hk; mzgeng@se.cuhk.edu.hk; xyliu@se.cuhk.edu.hk; hmmeng@se.cuhk.edu.hk).

Xurong Xie is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China (e-mail: xr.xie@siat.ac.cn).

I. INTRODUCTION

THERE has been a long history of using discriminative training techniques to improve the performance of automatic speech recognition (ASR) systems. In current neural network based systems, these discriminative training methods define the state-of-the-art performance. From the previous generation of Gaussian mixture model based Hidden Markov Models (HMMs) ASR systems [1]–[6] to the current systems using a hybrid HMM deep neural network (DNN) architecture [7]–[12], performance improvements obtained over the conventional cross-entropy (CE) trained systems have been widely reported. Although in recent years there has been a significant trend of moving from hybrid HMM-DNN system architectures to all neural end-to-end (E2E) modelling paradigm represented by listen, attend and spell (LAS) [13], connectionist temporal classification (CTC) [14], RNN transducers (RNN-T) [15] and neural transformers [16], state-of-the-art hybrid HMM-DNN systems featuring sequence discriminative training techniques, for example, maximum mutual information (MMI) criterion [1], [17], [18] trained factored time delay neural networks (TDNNs) [10], [19]–[21], remain highly competitive against end-to-end approaches to date [17], [22]–[24].

Since discriminative training methods were first introduced to the earlier generation of GMM-HMM based speech recognition systems [1], [25], [26], they have been long known to be prone to overfitting when using limited training data and a sparse representation of the modelling confusion over possible erroneous recognition hypotheses. In the context of deep neural network based ASR systems, this overfitting issue also presents [27], for example, when using smaller sized and shallower lattices to train systems with a very large number of HMM state targets [9]. Such issue is further aggravated by the use of stochastic gradient based optimization techniques that operate sequentially in a batch mode on smaller subsets of data randomly drawn from the complete training data collection.

In order to address the above issue, several categories of techniques have been developed in recent years to improve the generalization performance of discriminative training for DNN based ASR systems. Drawing inspirations from the earlier regularization techniques used in the discriminative training of GMM-HMM systems [28], the first category of methods attempts to alleviate the problem by optimizing the interpolated error cost between a sequence level discriminative training criterion, for example, MMI, and the conventional CE cost,

as in F-smoothing [9]. Motivated by the data intensive nature of deep learning techniques, the second category of techniques reduce the risk of overfitting using data augmentation methods. By expanding the limited training data using, for example, speed perturbation [29], spectral deformation [30], simulation of noisy and reverberated speech [31], the coverage of the augmented training data and the resulting speech recognition systems' generalization performance can be improved. The third category of methods address the overfitting issue by modifying the optimization algorithm. These include the incorporation of an additional L2 norm term into the original discriminative error cost function [8]. Second-order methods represented by Hessian-free optimization [32]–[34] and the use of natural gradient [35]–[37] have been investigated. Model parameter averaging [37] over different batch intervals at the end of a training epoch can also be used, for example, employed in the Kaldi toolkit as a standard regularization technique. Weight noise adds noise directly to the network parameters to improve generalization [38], [39]. Finally, dropout, a simple and effective approach used in deep learning models to avoid overfitting [40] can also be adopted. However, it lacks of a mathematically well defined framework to model the underlying DNN ASR systems' uncertainty arising from the use of limited data, when compared with Bayesian neural networks [41]. Most of the above regularization techniques are currently used in the Kaldi implementation of lattice-free maximum mutual information (LF-MMI) sequence training of factored TDNN systems [10].

This paper presents a mathematically well grounded, full Bayesian framework to account for model uncertainty in sequence discriminative training of factored TDNN acoustic models. In contrast to conventional Bayesian neural networks marginalizing over the CE error cost, an integration of the sequence level MMI criterion is used. Modelling uncertainty is addressed using three full Bayesian approaches. Bayesian TDNN systems [42]–[46] are used to model uncertainty over the weight parameters. Gaussian Process TDNN systems are further introduced to consider both the uncertainty associated with the weight parameters, as well as that over the choice of hidden activation functions. Variational TDNN systems are proposed to consider the uncertainty over the hidden layer outputs. Efficient variational inference approaches developed for all the above Bayesian TDNN systems using a very small number of samples (as low as one) ensure their computational cost in both training and evaluation time comparable to that of the baseline TDNN systems. A theoretical connection is further drawn between full Bayesian inference and dropout by re-formulating the latter as a special case of Bayesian TDNN systems.

Experiments conducted on a state-of-the-art 900-h speed perturbed Switchboard corpus trained baseline LF-MMI factored TDNN system featuring multiple built-in regularization methods including F-smoothing [9], L2 norm [8], natural gradient [35]–[37], model averaging [37] and dropout [40], as well as i-Vector [47], [48] and learning hidden unit contribution (LHUC) [49] speaker adaptation suggests the proposed Bayesian TDNN, Gaussain Process TDNN and variational TDNN systems consistently outperform the baseline systems by a statistically significant margin of 0.4%-1.8% absolute (5%-11% relative)

reduction in word error rate over the NIST Hub5'00, RT02 and RT03 sets. Similar consistent performance improvements were also obtained after the recurrent neural network language model rescoring, as well as on a 450-h (with speed perturbation) HKUST conversational Mandarin telephone speech recognition task. The efficacy of the proposed Bayesian estimation techniques is further demonstrated on a cross domain adaptation task. A 1000-h LibriSpeech corpus trained LF-MMI TDNN system is rapidly domain adapted to a highly challenging elderly speech recognition corpus based on a 10-h Dementia Bank Pitt database. Consistent performance improvements of 1.1% absolute WER reduction over LF-MMI baseline TDNN systems using direct weight fine-tuning were obtained.

The main contributions of this paper are summarized below:

1) This paper presents a first use of a mathematically well grounded, full Bayesian framework to account for model uncertainty in sequence discriminative training of factored TDNN acoustic models. A systematic overview and comparison over different full Bayesian TDNN learning variants is given. In contrast, only limited previous research on Bayesian neural network learning techniques was conducted for language modelling [50]. More recently, a Bayesian learning framework was used to account for model uncertainty in sequence discriminative training of factored TDNN acoustic models in our preliminary research [43], [44]. Stochastic noise injection to model parameters [38] was also exploited to improve the generalization performance of E2E ASR systems [39], [51].

2) Efficient variational inference approaches developed for all the above Bayesian TDNN systems using a very small number of samples (as low as one) ensure their computational cost comparable to that of the baseline systems. The generic nature of the proposed methods also allows them to be extended to other end-to-end approaches to address similar modelling uncertainty issues during system development.

3) Significant performance improvements on multiple data sets were obtained over baseline LF-MMI factored TDNN systems constructed using a large ensemble of built-in regularization methods including F-smoothing, L2 norm penalty, natural gradient, model averaging and dropout.

4) This paper further presents the earliest work on full Bayesian learning driven rapid domain adaptation of LF-MMI TDNN based ASR systems. In contrast to the previous research based on transfer learning [52], the proposed Bayesian domain adaptation technique provides an alternative useful approach to the problem of under-resourced speech recognition system development.

The rest of this paper is organized as follows. Section 2 introduces a full Bayesian learning framework for several neural network model variants that account for the uncertainty over the weight parameters, and the choice of activation functions or the hidden layer outputs. These include Bayesian neural networks (BNNs), Gaussian Process neural networks (GPNNs) and Variational neural networks (VNNs). Time delay neural networks (TDNNs) are presented in Section 3. Section 4 discusses the Bayesian estimation of TDNNs. Section 5 shows the

experiments and results. Finally, the conclusions are drawn in Section 6.

II. BAYESIAN LEANING BASED NEURAL NETWORK

In this section, we introduce several forms of Bayesian learning based neural networks presented in this paper, including Bayesian Neural Networks (BNNs), Gaussian Process Neural Networks (GPNNs) and Variational Neural Networks (VNNs).

A. Bayesian Neural Network

In conventional neural networks using fixed-point parameter estimates, the uncertainty associated with the prediction is hard to quantify. Bayesian neural networks (BNNs) offer a formalism to understand and quantify the uncertainty by using the posterior distribution to model the parameter uncertainty in the predictive distribution [53], [53]–[57]. To make predictions for the observations of test utterance $\mathbf{O}_r^*$, we average over all the parameter values, weighted by their posterior probability.

$$p(\mathbf{H}_r^*|\mathbf{O}_r^*, \mathbf{D}) = \int p(\mathbf{H}_r^*|\mathbf{O}_r^*, \mathbf{w})p(\mathbf{w}|\mathbf{D})d\mathbf{w} \qquad (1)$$

where $\mathbf{H}_r^*$ is the predicted word-sequence for utterance r, $\mathbf{w}$ is the activation parameter, $p(\mathbf{w}\,|\,\mathbf{D})$ denotes the posterior distribution to be learned from training data $\mathbf{D} = \{\mathbf{H}_r, \mathbf{O}_r\}$, $\mathbf{O}_r$ is the sequence of observation for utterance r and $\mathbf{H}_r$ is the reference word transcription for utterance r.

If all subsequent layers $l + 1, \ldots, L$ are removed, the expected hidden node output $h_i^{(l)}$ of the i-th node in the l-th layer is marginalized over different parameter estimates.

$$h_i^{(l)} = \int \phi\left(\mathbf{w}_i^{(l)} \bullet \mathbf{h}^{(l-1)}\right) p(\mathbf{w}_i^{(l)} \,|\, \mathbf{D})d\mathbf{w}_i^{(l)} \qquad (2)$$

where $\mathbf{h}^{(l-1)}$ is the input vector fed into the l-th hidden layer (the output from the previous layer $l-1$), $p(\mathbf{w}_i^{(l)}|\mathbf{D})$ denotes the node dependent activation parameter posterior distribution, $\phi(\cdot)$ is the activation function and $\bullet$ denotes the dot product.

B. Gaussian Process Neural Network

Gaussian Processes (GPs) [58] are powerful nonparametric distributions over continuous functions that are used in probabilistic modelling for many machine learning applications including regression and classification tasks and beyond. A function modelled using Gaussian process is represented as

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right) \qquad (3)$$

where $\mathbf{x}, \mathbf{x}'$ are arbitrary inputs, $m(\cdot)$ is the mean function and $k(\cdot, \cdot)$ is the kernel function.

The above formulation is known as the kernel space view of GP models [58]. The associated computational complexity over the kernel covariance function during inference is determined by the size of the training data, and therefore impractical to be directly applied in the large-scale tasks, for example, speech recognition systems that often use tens of millions of frame samples or more in training. An alternative and computationally more tractable form of GP models uses basis function interpolation (see Chapter 2 of [58]), leads to the following weight space

view of GP,

$$f(\mathbf{x}) = \boldsymbol{\lambda}^T \bullet \phi(\mathbf{x}) = \sum_m \lambda^m \phi^m(\mathbf{x}) \qquad (4)$$

where $k(\cdot, \cdot) = \phi(\cdot)^T \phi(\cdot), \boldsymbol{\lambda} \sim \mathcal{N}(\cdot, \cdot)$ represents amplitudes of different basis functions $\phi^m(\mathbf{x})$ in $\phi(\cdot)$.

The connection between neural networks and Gaussian Processes has also been extensively studied. Based on MacKay's work on the Bayesian neural network [54], Neal [59] proved that single-hidden-layer Bayesian neural networks of infinite width are equivalent to Gaussian Processes [58]. Hazan and Jaakkola [60] and later Lee [61] proposed the use of GP kernels to approximate infinitely wide deep neural networks. In deep Gaussian processes (DGPs) [62] models deep belief neural network layers were replaced by Gaussian Processes.

The form of traditional Bayesian neural networks introduced earlier in Sec. II-A only considers the uncertainty associated with weight parameters, but not the network structural configurations. For example, the choice over the hidden activation functions can be learned using a simple output level interpolation of commonly used basis activation functions, i.e., Sigmoid, Tanh, ReLU as the following.

$$h_i^{(l)} = \sum_m \lambda_i^{(l,m)} \phi_m\left(\mathbf{w}_i^{(l,m)} \bullet \mathbf{h}^{(l-1)}\right) \qquad (5)$$

where $\lambda_i^{(l,m)}$ is the m-th basis activation coefficient and ϕ_m is the m-th basis activation function.

Within a more general framework of Gaussian Process neural networks (GPNNs), not only the weight parameters inside the activation functions can be treated as random variables, the additional uncertainty over the basis coefficients can also be considered. The prediction is rewritten as

$$p(\mathbf{H}_r^*|\mathbf{O}_r^*, \mathbf{D}) = \int \int p(\mathbf{H}_r^*|\mathbf{O}_r^*, \mathbf{w}, \boldsymbol{\lambda})p(\mathbf{w}|\mathbf{D})p(\boldsymbol{\lambda}|\mathbf{D})d\mathbf{w}d\boldsymbol{\lambda} \qquad (6)$$

where $p(\boldsymbol{\lambda}\,|\,\mathbf{D})$ and $p(\mathbf{w}\,|\,\mathbf{D})$ denote the basis activation coefficient and parameter posterior distributions respectively. We assume these two variables are independent. The general form of Gaussian Process Neural Network can be further simplified into four special cases in Table I by considering different uncertainty modelling combinations (marginalization over both $\mathbf{w}$ and $\boldsymbol{\lambda}$ or only one of them).

Similarly, the expected hidden node output $h_i^{(l)}$ in GPNN can be modified into the integration of both the weight parameters and basis coefficients in Eqn. (7).

$$h_i^{(l)} = \sum_m \int \int \lambda_i^{(l,m)} \phi_m\left(\mathbf{w}_i^{(l,m)} \bullet \mathbf{h}^{(l-1)}\right)$$
$$p(\mathbf{w}_i^{(l,m)}\,|\,\mathbf{D})p(\lambda_i^{(l,m)}\,|\,\mathbf{D})d\mathbf{w}_i^{(l,m)}d\lambda_i^{(l,m)} \qquad (7)$$

where $\mathbf{h}^{(l-1)}$ is the input vector fed into the l-th hidden layer, $p(\lambda_i^{(l,m)}\,|\,\mathbf{D})$ and $p(\mathbf{w}_i^{(l,m)}\,|\,\mathbf{D})$ denote the basis activation coefficient and parameter posterior distributions.

TABLE I
DESCRIPTION OF BAYESIAN, GAUSSIAN PROCESS AND VARIATIONAL TDNN LAYER IN TERMS OF THEIR RESPECTIVE FORMS OF NUMBER OF SAMPLES, UNCERTAINTY MODELLING, NUMBER OF FREE PARAMETERS AFTER TYING AND THE SPEED RATIO RELATIVE TO THE STANDARD TDNN LAYER IN TRAINING AND EVALUATION, BY ASSUMING THE INPUT VECTOR SIZE AS a, THE NUMBER OF HIDDEN NODES AS b AND THE LATENT VARIABLE $\mathbf{z}$ DIMENSION AS c FOR A SINGLE LAYER. THE BAYESIAN AND GAUSSIAN PROCESS TDNN LAYERS ARE CONSTRUCTED FOLLOWING THE IMPLEMENTATION DETAILS OF SECTION IV-E

Layer	#Sa-mple	Uncertainty			#Param			Speed Ratio	
		λ	w	z	λ	w	z	Train	Test
TDNN	1	✗	✗	✗	0	ab	0	1.0	1.0
B-TDNN	1	✗	✓	✗	0	ab+a	0	1.2	1.0
B-TDNN	2	✗	✓	✗	0	ab+a	0	1.4	1.0
B-TDNN	3	✗	✓	✗	0	ab+a	0	1.6	1.0
GP-TDNN0	1	✗	✗	✗	3b	ab	0	1.1	1.0
GP-TDNN1	1	✓	✗	✗	3b+3	ab	0	1.1	1.0
GP-TDNN2	1	✗	✓	✗	3b	ab+a	0	1.2	1.0
GP-TDNN3	1	✓	✓	✗	3b+3	ab+a	0	1.2	1.0
V-TDNN	1	✗	✗	✓	0	ab+cb	4ac	1.3	1.0

C. Variational Neural Network

In contrast to the BNN and GPNN models presented in Sec. II-A and Sec. II-B, when modelling the uncertainty associated with hidden layer outputs, variational neural networks (VNNs) [63]–[66] can be used. Instead of modelling the uncertainty over the weight parameters inside the activation functions in BNNs or assuming additional uncertainty over the activation basis coefficients in GPNNs, variational neural networks introduce a latent variable $\mathbf{Z}$ to encode the uncertainty associated with the hidden layer outputs, and in turn the final predictive distribution.

$$p(\mathbf{H}_r^*|\mathbf{O}_r^*, \mathbf{D}) = \int p(\mathbf{H}_r^*|\mathbf{O}_r^*, \mathbf{Z}_r^*)p(\mathbf{Z}_r^*|\mathbf{O}_r^*, \mathbf{D})d\mathbf{Z}_r^* \quad (8)$$

where $p(\mathbf{Z}_r^*|\mathbf{O}_r^*, \mathbf{D})$ denotes the latent variable posterior distribution to be learned from the training data $\mathbf{D}$.

Similarly, the expected hidden node output is calculated as in Eqn. (9).

$$\mathbf{h}^{(l)} = \int \phi\left([\mathbf{Z}_r^{*(l-1)}, \mathbf{h}^{(l-1)}]\right)p(\mathbf{Z}_r^{*(l-1)}|\mathbf{O}_r^*, \mathbf{D})d\mathbf{Z}_r^{*(l-1)} \quad (9)$$

where $p(\mathbf{Z}_r^{*(l-1)}|\mathbf{O}_r^*, \mathbf{D})$ denotes the latent variable posterior distribution of layer l, $\phi(\cdot)$ is the activation function. Note that majority of systems only consider one layer to apply the variational distribution.

III. TIME DELAY NEURAL NETWORK

Time delay neural networks (TDNNs) [10], [17], [19]–[21], [67] based hybrid HMM-DNN acoustic models in recent years defined state-of-the-art speech recognition performance over a wide range of tasks, due to their strong power in modelling long range temporal dependencies in speech. In particular, the recently proposed factored TDNN systems [21] featuring lattice-free MMI sequence discriminative training [10] remain highly competitive against all neural end-to-end approaches to date [17], [22]–[24].

TDNNs can be considered as a special form of one-dimensional convolutional neural networks (CNNs) [68] when
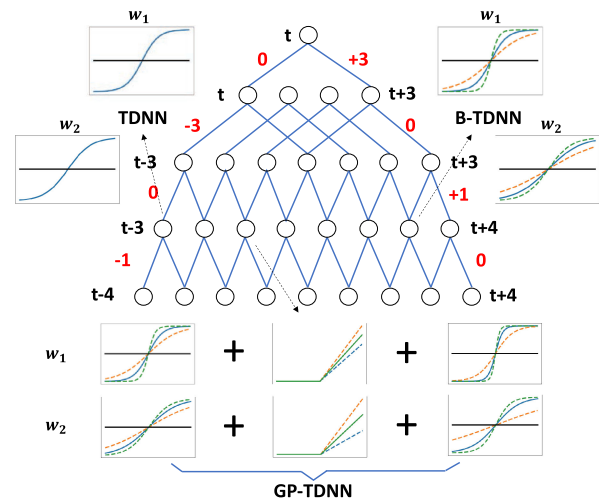

Fig. 1. An example TDNN architecture with the option of using standard TDNN, Bayesian TDNN and Gaussian Process TDNN. In this example, the input dimension of all hidden nodes is assumed to be two. w_1 and w_2 are the corresponding weights for each input dimension of a hidden node. Conventional TDNN systems use fixed value, deterministic estimation of the weight parameters w_1, w_2 (Top left). B-TDNN systems of Sec. IV-A use latent weight posterior distributions to account for model uncertainty. The GP-TDNN systems of Sec. IV-C use latent weight posterior distributions for three basis activations of varying non-linearity to be combined over, thus considering uncertainty over both the weight parameters w_1, w_2 and the choice of hidden activation functions.

parameters are tied across different time steps. An example TDNN model is shown in Fig. 1. The bottom layers of TDNNs are designed to learn a narrower temporal context span, while the higher layers to learn wider, longer range temporal contexts. One important type of hyper-parameters in TDNN models controlling its temporal modelling ability is the left and right splicing context offsets. These alter the temporal context ranges effectively learned in each TDNN hidden layer. The splicing context offsets used in the example of Fig. 1 are {-1,0} {0,1} {-3,0} {0,3} from the bottom to the top layer. In this paper, we adopt a factored form of TDNN model structure [21], which compresses the weight matrix by using semi-orthogonal matrix decomposition.

IV. BAYESIAN ESTIMATION OF TDNN

This section presents the LF-MMI based sequence level discriminative estimation schemes for Bayesian TDNNs (B-TDNNs), Gaussian Process TDNNs (GP-TDNNs) and Variational TDNNs (V-TDNNs). In addition, dropout is re-formulated as a special case of Bayesian TDNN systems, before being further extended into a more generalized form and integrated with the full Baysian TDNN systems.

A. Bayesian TDNN

For any cost error function using the cross-entropy or the sequence training criterion, for example, MMI [1], the same back-propagation algorithm in the gradient chain can be applied as in Eqn. (10). The only term needs to be changed is the first part in the chain, which is modified into the specific error cost

function gradient w.r.t the last layer outputs $\frac{\partial F}{\partial h_j^L}$ in different tasks.

$$\nabla_{\theta_i}^l F = \sum_j \frac{\partial F}{\partial h_j^L} \underbrace{\left\{ \sum_k \frac{\partial h_j^L}{\partial h_k^{L-1}} \cdots \left[\sum_i \frac{\partial h_i^l}{\partial \theta_i^l} \right] \right\}}_{\text{gradient chain}} \quad (10)$$

where $\theta_i^{(l)} = w_i^{(l)}$ corresponds to the i-th node dependent parameter in the l-th layer, $h_i^{(l)}$ is the i-th hidden node output in the l-th layer and F is the error cost function, for example, the MMI criterion [1] in Eqn. (11).

$$F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta}) = \sum_r \log \frac{p(\mathbf{O}_r | \mathbf{H}_r)^k P(\mathbf{H}_r)}{\sum_{\mathbf{H}_r'} p(\mathbf{O}_r | \mathbf{H}_r')^k P(\mathbf{H}_r')} \quad (11)$$

where $P(\mathbf{H}_r')$ is the language model probability for the confusable word sequence $\mathbf{H}_r'$, $\mathbf{H}_r$ is the reference word sequences, $\boldsymbol{\Theta}$ contains the hyper-parameters of the latent distributions in Bayesian and Gaussian Process TDNNs as well as fixed parameters inside them if there are any, k is the acoustic scaling factor. The sum of the denominator is taken over all possible word sequences for utterance r.

The following marginalization of the training data MMI loss function in Eqn. (11) is optimized to infer the latent weight parameter distribution in $p(\mathbf{H}_r^* | \mathbf{O}_r^*, \mathbf{D})$.

$$F = \log \int \exp \{ F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta}) \} P_r(\mathbf{w}) d\mathbf{w} \quad (12)$$

where $\mathbf{w} \in \boldsymbol{\Theta}$ and $P_r(\mathbf{w})$ denotes the weight prior distribution.

When the MMI criterion in Eqn. (12) uses no acoustic probability scaling ($k=1$) and a sufficiently large set of confusable word sequence $\mathbf{H}_r'$ for each utterance in the training data, the evidence integral in Eqn. (12) is equivalent to a marginalization of the *conditional maximum likelihood* of the reference word sequences $\mathbf{H}_r$ given $\mathbf{O}_r$.

The commonly used variational inference is used to approximate the integration in Eqn. (12). Instead of explicitly computing the posterior distribution, the evidence lower bound is first derived by Jesen's inequality in Eqn. (13). Then we directly optimize the evidence lower bound to find a variational distribution $q(\mathbf{w})$ to approximate the posterior distribution. The first term in the evidence lower bound of Eqn. (13) can be approximated with Monte Carlo sampling method in Eqn. (15). Further rearranging the second KL term in the lower bound in Eqn. (13) allows the hyper-parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ to be differentiable and updated.

$$F \geq \int q(\mathbf{w}) F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta}) d\mathbf{w} - \text{KL}(q(\mathbf{w}) \| P_r(\mathbf{w}))$$
$$= \mathcal{L}_1^{\text{MMI}} - \mathcal{L}_2^{\text{MMI}} = \mathcal{L}^{\text{MMI}} \quad (13)$$

where $q(\mathbf{w})$ is the variational approximation of the parameter posterior distribution $p(\mathbf{w}|\mathbf{D})$, $\text{KL}(q\|P_r)$ is the Kullback-Leibler (KL) divergence between q and P_r. For simplicity, both q and P_r are assumed to be Gaussian distributions,

$$q(\mathbf{w}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \quad P_r(\mathbf{w}) = \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r^2) \quad (14)$$

The first term $\mathcal{L}_1^{\text{MMI}}$ in Eqn. (13) can be efficiently approximated by Monte Carlo sampling method. The integrand is

re-parameterized so that it does not depend on the $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ directly, but instead on the standard normal distribution $\boldsymbol{\epsilon}$.

$$\mathcal{L}_1^{\text{MMI}} \approx \frac{1}{N} \sum_{k=1}^N F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta}, \mathbf{w}_k) \quad (15)$$
$$\mathbf{w}_k = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_k$$

where $\boldsymbol{\epsilon}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the k-th sample in the total N samples and $\odot$ denotes the Hadamard product.

The KL divergence between q and P_r of the second term $\mathcal{L}_2^{\text{MMI}}$ can be simplified into an analytical form as follows.

$$\mathcal{L}_2^{\text{MMI}} = \sum_j \left\{ \log \frac{\sigma_{r,j}}{\sigma_j} + \frac{\sigma_j^2 + (\mu_j - \mu_{r,j})^2}{2\sigma_{r,j}^2} - \frac{1}{2} \right\} \quad (16)$$

where μ_j and σ_j are the j-th component of variational posterior distribution hyper-parameters $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, $\mu_{r,j}$ and $\sigma_{r,j}$ are the j-th component of prior distribution hyper-parameters $\boldsymbol{\mu}_r$ and $\boldsymbol{\sigma}_r$ respectively.

The gradients w.r.t the hyper-parameters μ_j, σ_j are given as below.

$$\frac{\partial \mathcal{L}^{\text{MMI}}}{\partial \mu_j} = \frac{1}{N} \sum_{k=1}^N \frac{\partial F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta}, \boldsymbol{\epsilon}_k)}{\partial \mu_j} - \frac{\mu_j - \mu_{r,j}}{\sigma_j^2}$$
$$\frac{\partial \mathcal{L}^{\text{MMI}}}{\partial \sigma_j} = \frac{1}{N} \sum_{k=1}^N \frac{\partial F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta}, \boldsymbol{\epsilon}_k)}{\partial \sigma_j} - \frac{\sigma_j^2 - \sigma_{r,j}^2}{\sigma_j \sigma_{r,j}^2} \quad (17)$$

where the gradients required by the right hand side of Eqn. (17) can be directly calculated using the standard back-propagation method.

B. Bayesian Dropout TDNN

Dropout is a standard technique widely used in deep learning to avoid overfitting [40]. It can be viewed as a special form of Bayesian TDNN systems when variational distribution $q(\mathbf{w})$ is written as the following form.

$$q(\mathbf{w}) = a\delta(\mathbf{w}) + (1-a)\mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_1^2) \quad (18)$$

where a is the interpolation weight of two component distributions and $\boldsymbol{\sigma}_1$ is fixed to be a small constant value, for example, $\exp(-3)$. $\delta(\mathbf{w})$ is the delta function taking the value of 1 when using the weight parameter $\mathbf{w}$. Traditionally, a is parametrized as a Bernoulli random variable. In this case, $\mathbf{w}$ either keeps its original value with probability a, or is replaced by a dropout sample drawn from the $\mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_1^2)$ with probability $1 - a$. It is clear that the first variational distribution component in the standard Dropout of Eqn. (18) can not be Bayesian estimated. In order to generalize it and fully integrate it into the Bayesian TDNN system training process, the following Bayesian Dropout [41] in Eqn. (18) can be used.

$$q(\mathbf{w}) = a\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}_0^2) + (1-a)\mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_1^2) \quad (19)$$

where hyper-parameter $\boldsymbol{\sigma}_0$ is learned by variational inference as in B-TDNN while $\boldsymbol{\sigma}_1$ is fixed as a small constant value.

When using Monte Carlo sampling to approximate the first term of the evidence lower bound in Eqn. (13), the corresponding

weight samples for Bayesian Dropout TDNNs are modified as given in Eqn. (20).

$$\mathbf{w}_k = a(\boldsymbol{\mu} + \boldsymbol{\sigma_0} \odot \boldsymbol{\epsilon}_k) + (1-a)\boldsymbol{\sigma_1} \odot \boldsymbol{\epsilon}_k \qquad (20)$$

where a is the interpolation weight fixed to be 0.5 on all our experiments, $\boldsymbol{\epsilon}_k$ is the k-th sample drawn from $\boldsymbol{\epsilon}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ denotes the Hadamard product.

With the variational distribution $q(\mathbf{w})$ defined in Eqn. (19) and the prior distribution $P_r(\mathbf{w}) = \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r^2)$, the second KL divergence of the second term $\mathcal{L}_2^{\text{MMI}}$ in Eqn. (13) can also be approximated as [41]:

$$\mathcal{L}_2^{\text{MMI}} \approx a \sum_j \left\{ \frac{\sigma_{0,j}^2 + (\mu_j - \mu_{r,j})^2}{2\sigma_{r,j}^2} - \log(\sigma_{0,j}) \right\}$$
$$+ (1-a) \sum_j \left\{ \frac{\sigma_{1,j}^2}{2\sigma_{r,j}^2} - \log(\sigma_{1,j}) \right\} - C \qquad (21)$$

where μ_j and $\sigma_{i,j}$ are the j-th components of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}_i$, C is a constant term. Note that the KL divergence term approximated in Eqn. (19) is multiplied by the probability of the weights keeping the values. If the dropout probability is set to be zero ($a = 1$), it leads to the same form of Bayesian TDNNs using the variational distribution of Eqn. (14).

C. Gaussian Process TDNN

Gaussian Process time delay neural networks (GP-TDNN) model can be viewed as a specific type of the more general Gaussian Process neural networks (GPNNs) that are introduced in Sec. II-B. The connection between Gaussian Processes (GP) and GP-TDNNs, lies in the fact that for each hidden node of a TDNN, a weight space view [58] of a Gaussian Processes expressed as an interpolation over Sigmoid, Tanh, ReLU basis activation outputs, as in Eqn. (5) of Sec. II-B, is used to replace the use of a single ReLU activation function of fixed value parameters in a standard TDNN model. This corresponds to the GP-TDNN0 model shown in Table I. Further consideration over the modelling uncertainty associated with either the basis activation coefficients $\boldsymbol{\lambda}$, or the basis activation internal weight parameters $\mathbf{w}$, or both of these, leads to the other GP-TDNN variants (GP-TDNN1, GP-TDNN2 and GP-TDNN3) shown in Table I.

Similar to the variational inference procedure used in Sec. IV-A for Bayesian TDNNs, the evidence lower bound in Eqn. (22) is used to approximate the MMI criterion marginalization F over both $\mathbf{w}$ and $\boldsymbol{\lambda}$.

$$F \geq \iint q(\mathbf{w})q(\boldsymbol{\lambda})F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta})d\mathbf{w}d\boldsymbol{\lambda}$$
$$- \text{KL}(q(\mathbf{w})\|P_r(\mathbf{w})) - \text{KL}(q(\boldsymbol{\lambda})\|P_r(\boldsymbol{\lambda})) \qquad (22)$$
$$= \mathcal{L}_1^{\text{MMI}} - \mathcal{L}_2^{\text{MMI}} - \mathcal{L}_3^{\text{MMI}} = \mathcal{L}^{\text{MMI}}$$

where $\{\boldsymbol{\lambda}, \mathbf{w}\} \in \boldsymbol{\Theta}$ and we assume the statistical independence between $\mathbf{w}$ and $\boldsymbol{\lambda}$ holds. $q(\mathbf{w})$ and $q(\boldsymbol{\lambda})$ are the variational approximations of the parameter posterior distribution $p(\mathbf{w}|\mathbf{D})$ and basis coefficient posterior distribution $p(\boldsymbol{\lambda}|\mathbf{D})$ respectively.
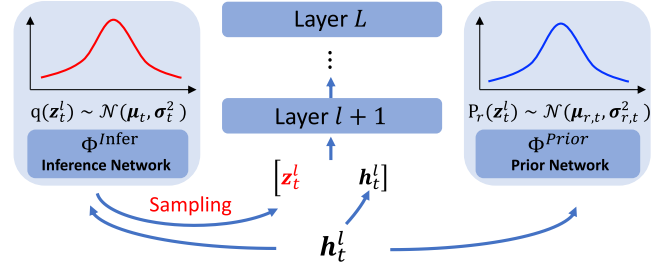

Fig. 2. An example Variational TDNN. Note that the output vector of l-th layer $\mathbf{h}_t^l$ is used as the input of the inference network Φ^{Infer} and prior network Φ^{Prior} to calculate the mean and variance of the latent variable's variational and prior distributions respectively. Then we concatenate the latent variable $\mathbf{z}_t^l$ sampled from the inference network Φ^{Infer} with $\mathbf{h}_t^l$ as the input of the (l+1)-th layer.

$\text{KL}(q\|P_r)$ is the Kullback-Leibler (KL) divergence between q and prior distribution P_r. Following the settings used in Byesian TDNNs of Sec. IV-A, q and P_r are both set to be Gaussian distributions and the first term $\mathcal{L}_1^{\text{MMI}}$ is calculated by Monte Carlo sampling method.

D. Variational TDNN

In variational TDNNs (V-TDNNs), the MMI cost function is marginalized over a sequence level hidden outputs meta-vector $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T]$, where the time instance level random hidden output vectors $[\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T]$ are assumed to be independent among themselves. This leads to the following marginalized cost function

$$F = \log \int \exp \{F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta}, \mathbf{Z})\} P_r(\mathbf{Z})d\mathbf{Z}$$
$$= \sum_{t=1}^{T} \log \int \exp \{F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta}, \mathbf{z}_t)\} P_r(\mathbf{z}_t)d\mathbf{z}_t \qquad (23)$$

where $\mathbf{z}_t$ is the latent variable at time t and $P_r(\mathbf{z}_t)$ denotes the prior distribution of the latent variable.

Then, the variational lower bound is derived to approximate the marginalization of MMI criterion F in Eqn. (24).

$$F \geq \sum_t \int q(\mathbf{z}_t)F_{\text{MMI}}(\mathbf{D}; \boldsymbol{\Theta})d\mathbf{z}_t - \sum_t \text{KL}(q(\mathbf{z}_t)\|P_r(\mathbf{z}_t))$$
$$= \mathcal{L}_1^{\text{MMI}} - \mathcal{L}_2^{\text{MMI}} = \mathcal{L}^{\text{MMI}} \qquad (24)$$

where $q(\mathbf{z}_t)$ is the variational approximation of the posterior distribution $p(\mathbf{z}_t|\mathbf{D})$ and $P_r(\mathbf{z}_t)$ is the prior distribution. As shown in Fig. 2,

$$q(\mathbf{z}_t) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2), \quad P_r(\mathbf{z}_t) = \mathcal{N}(\boldsymbol{\mu}_{r,t}, \boldsymbol{\sigma}_{r,t}^2) \qquad (25)$$

where hyper-parameters $\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2$ are calculated from an inference network $\Phi^{Infer}(\mathbf{h}_t)$, hyper-parameters $\boldsymbol{\mu}_{r,t}, \boldsymbol{\sigma}_{r,t}^2$ are computed by a prior network $= \Phi^{Prior}(\mathbf{h}_t)$.

In common with the estimation procedures used in B-TDNN, $\mathcal{L}^{\mathrm{MMI}}$ is further approximated by Monte Carlo sampling, i.e.,

$$
\mathcal{L}^{\mathrm{MMI}} = \mathcal{L}_1^{\mathrm{MMI}} - \mathcal{L}_2^{\mathrm{MMI}}
$$

$$
\approx \sum_t \frac{1}{N} \sum_{k=1}^N F_{\mathrm{MMI}}(\mathbf{D}; \boldsymbol{\Theta}, \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_k) \qquad (26)
$$

$$
- \sum_t \mathrm{KL}(q(\mathbf{z}_t) \| P_r(\mathbf{z}_t))
$$

where $\mathbf{z}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_k$, $\boldsymbol{\epsilon}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the k-th sample in the total N samples and $\odot$ denotes the Hadamard product.

E. Implementation Details

The performance and efficiency of the proposed Bayesian, Gaussian Process and Variational TDNN systems in Sec. IV-A to IV-D are affected by the following set of implementation details. Striking a sensible balance between performance and computational cost is crucial for practical implementation of these systems.

1) **Choice of prior distribution:** When training Bayesian learning based models, a suitable choice of parameter prior needs to be set. In our experiments, we set the priors for various Bayesian learned LF-MMI TDNN systems to be based on the comparable converged standard fixed-parameter TDNN systems. In addition, all the other parameters in the Bayesian learning based LF-MMI TDNN models are initialized using the parameters obtained from the comparable half-trained standard TDNN systems. The combination of these two settings in practice was found to yield a good balance between convergence speed and performance.

2) **Modelling uncertainty at different layers** Applying Bayesian estimation of all layers inside TDNN systems is highly expensive in both model training and evaluation. It is well known that deep neural networks including TDNNs are powerful models that are capable of producing denoised and invariant features in their higher layers for accurate classification of the inputs. It is therefore expected that the modelling uncertainty associated with the lower layers of TDNN systems will be much larger than those of the higher layers. This is confirmed in our experiments to be presented in Sec. V-A. In practice, it is found that Bayesian estimation only needs to be applied to the first TDNN hidden layer where the largest modelling uncertainty is expected, while further applying Bayesian estimation of any subsequent higher layers produces no further performance improvement.

3) **Parameter tying for variational distributions** The extensive use of variational distributions during Bayesian inference in Sec. IV-A to IV-D leads to a large number of latent distribution hyper-parameters to be estimated and stored. In order to ensure the number of free parameters in the proposed Bayesian and Gaussian Process TDNN systems to be comparable to that of the standard TDNN systems, the variational distribution variance $\boldsymbol{\sigma}$ is shared among all the hidden nodes of the same layer for Bayesian and Gaussian Process TDNN systems. In addition, we further share the latent distribution over the weight parameters

TABLE II
PERFORMANCE (WER%) COMPARISON BETWEEN TDNN, B-TDNN SYSTEMS CONSTRUCTED USING THE 75-HOUR SWITCHBOARD TRAINING SUBSET BY DRAWING VARYING NUMBER OF SAMPLES (1,2 AND 3). THE WERs WERE EVALUATED ON THE **HUB5' 00**, **RT03S** AND **RT02** TEST SETS

System	#Sam-ple	Hub5' 00		Rt03S		Rt02			Total
		SWB1	CHM	FSH	SWB2	SWB3	SWB4	SWB5	Avg
TDNN	-	12.2	24.2	16.6	26.3	14.5	19.8	27.6	20.7
B-TDNN	1	11.7†	**23.3**†	**15.6**†	**25.0**†	14.3	**19.2**†	25.8†	19.7†
	2	12.0	23.6†	15.7†	25.4†	14.5	19.3†	25.9†	19.9†
	3	**11.6**†	23.7†	15.9†	**25.0**†	14.3	**19.2**†	25.7†	19.8†

†Denotes a Statistically Significant Difference is Obtained Over the TDNN Baseline System (Line 1). (SWB1 and CHM Denote the Switchboard and Callhm Subsets of the Hub5' 00 Test Set; FSH and SWB2 Denote the Fisher and Switchboard Subsets of the RT03S Test Set; SWB3, SWB4 and SWB5 Denote Three Switchboard Subsets in the Rt02 Test Set.)

$\mathbf{w}$ across all basis activation functions of Eqn. (7) in GP-TDNN systems to control the overall system complexity.

4) **Parameter sampling in inference** The inference algorithms of all the Bayesian estimated TDNN systems presented in Sec. IV-A to IV-D require the use of Monte Carlo sampling to approximate the respective parts of their lower bounds computing the MMI criterion expectation in Eqn. (15) and Eqn. (26) given the variational distributions. The resulting inference cost during model training is therefore linearly increased with respect to the number of samples being drawn as in Table I. Experimental results in Table II further show that only a marginal difference in Word Error Rate (WER) was observed by drawing more samples (two and three samples) in the forward pass of Bayesian TDNN systems. In order to maintain the Bayesian learned TDNN systems' overall computational cost during model training comparable to that of the conventional TDNNs as shown in line 2 and 5-9 in Table I, only one sample is drawn in Eqn. (15) and Eqn. (26) for all the Bayesian estimated TDNN systems presented in this paper. The KL term in Eqn. (13), Eqn. (22) and Eqn. (24) is set to be proportional to the batch size. During evaluation, the inference of Bayesian, Gaussian Process and Variational TDNNs in Eqn. (1), Eqn. (6) and Eqn. (8) are efficiently approximated by computing the expectation of the model parameters or the latent variables using the respective posterior distributions. For example, during recognition time, the weight parameters $\mathbf{w}$ in the B-TDNN systems are approximated by the mean of their latent distribution given as follows:

$$
\int p(\mathbf{H}_r^* | \mathbf{O}_r^*, \mathbf{w}) p(\mathbf{w} | \mathbf{D}) d\mathbf{w} \approx p(\mathbf{H}_r^* | \mathbf{O}_r^*, \mathbb{E}[\mathbf{w} | \mathbf{D}]) \qquad (27)
$$

Thus, the speed ratio relative to the standard TDNN system in Table I is approximately 1.0 at the test stage.

5) **System description** Following the above implementation details, the description of a set of Bayesian, Gaussian Process and Variational TDNN systems in terms of their respective forms of uncertainty modelling, number of free parameters after tying and the speed ratio relative to the standard TDNN systems in training and evaluation is presented in Table I. In the table, four variants GP-TDNN systems by considering no uncertainty (GP-TDNN0), or the uncertainty associated with either the activation basis coefficients $\boldsymbol{\lambda}$ (GP-TDNN1), or the

activation internal weight parameters $\mathbf{w}$ alone (GP-TDNN2), or the uncertainty associated with both $\mathbf{w}$ and $\boldsymbol{\lambda}$ (GP-TDNN3), are also shown.

As shown in Table I, by assuming the input vector size as a, the number of hidden nodes as b and the latent variable $\mathbf{z}$ dimension as c for a single layer, each standard TDNN layer has a total of ab parameters in the weight matrix $\mathbf{w}$. With the aforementioned parameter tying, each B-TDNN layer has a total of ab latent distribution parameters for the mean $\boldsymbol{\mu}$, and the number of parameters in the shared variance being a. Compared with a standard TDNN layer, a GP-TDNN0 layer has a total of ab parameters for the weight matrix, plus $3b$ additional parameters for the basis activation coefficients $\boldsymbol{\lambda}$. Compared with a GP-TDNN0 layer, a GP-TDNN1 layer requires 3 more parameters for the shared variance term of the latent distribution over $\boldsymbol{\lambda}$. By sharing the latent distribution over $\mathbf{w}$ across all basis activation functions of Eqn. (7), a GP-TDNN2 layer only needs a total of a more parameters for the shared variance term of the distribution over $\mathbf{w}$. If we model the uncertainty over both the basis activation coefficients $\boldsymbol{\lambda}$ and weight parameters $\mathbf{w}$ inside the activation functions, the GP-TDNN3 layer has a total of $3 + a$ more parameters than a GP-TDNN0 layer. For the Variational TDNN layer modelling the hidden output uncertainty as shown in Fig. 2, it requires $4ac$ parameters for the inference network Φ^{Infer} and prior network Φ^{Prior} to generate the mean and variance of the latent vatiables $\mathbf{z}$, plus additional $ab + cb$ parameters for the weight matrix $\mathbf{w}$.

In addition, when using the above efficient sampling during inference for model training (as low as one sample drawn) and evaluation, the Bayesian, Gaussian Process and Variational TDNN systems only require a moderate increase in system training time of 10%-30% over the standard TDNN baseline systems during training, while their computational complexity is comparable to that of standard TDNN systems during the testing stage.

V. EXPERIMENTS

This section is organized as follows. Firstly, in Section V-A, the performance of various Bayesian, Gaussian Process and Variational LF-MMI TDNN systems constructed using a 75-hour subset of the LDC Switchboard I data are evaluated. This initial set of experiments serve to confirm the implementation details and settings given in Sec. IV-E suitable to use for the subsequent larger experiments in the rest of this paper. Secondly, in Sec. V-B, the main set of experiments are conducted on a full 900-hour speed-perturbed Switchboard corpus to fully evaluate the performance of the Bayesian estimated LF-MMI TDNN systems proposed in Sec. IV. Performance comparison against the baseline LF-MMI TDNN system which used multiple regularization methods (F-smoothing, L2 norm penalty, natural gradient, model averaging and dropout), in addition to i-Vector plus learning hidden unit contribution (LHUC) based speaker adaptation and Kaldi recipe LSTM recurrent neural network language model (RNNLM) rescoring is drawn. Thirdly, in Sec. V-C, a comparable set of experiments are conducted in a 450-hour speed-perturbed HKUST conversational Mandarin telephone

speech recognition task. Finally, the performance of Bayesian estimated LF-MMI TDNN systems are further evaluated on a cross domain adaptation task which requires porting a 1000-h LibriSpeech data trained LF-MMI TDNN system to a small DementiaBank elderly speech corpus. In all our experiments, we follow the Kaldi chain model setup,[1] except that we used 40-dimension filterbank features as the input features instead of the 40-dimension high-resolution Mel-frequency cepstral coefficients (MFCCs). All of our models were trained with one thread on a single NVIDIA Tesla V100 Volta GPU card. For all results presented in this paper, matched pairs sentence-segment word error (MAPSSWE) based statistical significance test was performed at a significance level $\alpha = 0.05$.

A. Experiments on 75-Hour Switchboard Task

In this part, an investigation of different full Bayesian TDNN learning variants is conducted on the 75-hour Switchboard task to verify the feasibility of implementation details and settings in Sec. IV-E for further experiments in the rest of this paper. We first investigate the suitable number of layers to apply Bayesian estimation. After determining which layer(s) to incorporate Bayesian modelling, we compare the performance of the Bayesian TDNN system, Bayesian Dropout TDNN system, Gaussian Process TDNN system and Variational TDNN system described in Sec. IV-A to IV-D. Finally, we demonstrate the robustness of different Bayesian learning based TDNN systems by varying the model sizes (by varying hidden layer dimensionality).

Task Description: Our 75-hour Switchboard I data consists of randomly selected 1082 conversational sides out of the 4870 speakers from the 300-hour Switchboard I corpus released by LDC (LDC97S62). On top of the Linear Discriminant Analysis (LDA) transformed Perceptual Linear Prediction (PLP) coefficients up to the second order, our baseline GMM-HMM system with 2904 tied tri-phone states was trained using Maximum Likelihood Linear Transform (MLLT) [69], [70]. The speaker adaptive training (SAT) [71]–[73] approach was also applied to further generate the alignments for neural network training and the numerator lattices for LF-MMI training. For performance evaluation, a four-gram language model (LM) trained on the Switchboard and Fisher transcripts (LDC2004T19, LDC2005T19) was used to evaluate NIST HUB5'00 (LDC2002S09, LDC2002T43), RT03 (LDC2007S10) and RT02 (LDC2004S11) test sets. The performance of the LF-MMI trained standard TDNN system[2] is shown in line 1 of Table III. At this stage, i-Vector [74] and speed perturbation were not incorporated.

Experimental Results and Analysis As shown by most results in Table III, the proposed Bayesian estimated TDNN systems except the Variational TDNN system significantly outperform the TDNN baseline system (line 1 in Table III) across all three test sets. Several trends are listed as follows.

---

[1] All of this is in published Kaldi code at https://github.com/kaldi-asr/kaldi/tree/master/egs/*/*/local/chain/tuning/run_tdnn_7q.sh

[2] All of this is in published Kaldi code at https://github.com/kaldi-asr/kaldi/tree/master/egs/swbd/s5c/local/chain/tuning/run_tdnn_7q.sh

TABLE III
PERFORMANCE (WER%) COMPARISON OF TDNN, B-TDNN, BD-TDNN, GP-TDNN AND V-TDNN SYSTEMS CONSIDERING THE UNCERTAINTY AT DIFFERENT LAYERS CONSTRUCTED USING THE 75-HOUR SWITCHBOARD TRAINING SUBSET. THE WERS WERE EVALUATED ON THE **HUB5' 00**, **RT03S** AND **RT02** TEST SETS

System	Layer	Hub5' 00		Rt03S		Rt02		
		SWB1	CHM	FSH	SWB2	SWB3	SWB4	SWB5
TDNN	1-15	12.2	24.2	16.6	26.3	14.5	19.8	27.6
B-TDNN	1	$11.7^\dagger$	$23.3^\dagger$	$15.6^\dagger$	$25.0^\dagger$	14.3	$19.2^\dagger$	$25.8^\dagger$
	1-2	11.9	$23.5^\dagger$	$16.0^\dagger$	$26.1^\dagger$	14.4	$18.9^\dagger$	$25.9^\dagger$
	1-5	12.1	$23.5^\dagger$	$16.1^\dagger$	$25.3^\dagger$	14.4	19.4	$26.0^\dagger$
	1-8	11.9	$23.6^\dagger$	$15.8^\dagger$	$25.4^\dagger$	14.5	19.3	$26.3^\dagger$
	1-15	12.1	24.3	16.4	26.1	15.2	19.5	27.2
BD-TDNN	1	$11.8^\dagger$	$23.6^\dagger$	$15.6^\dagger$	$25.0^\dagger$	14.4	$19.1^\dagger$	$25.6^\dagger$
GP-TDNN0	1	$11.8^\dagger$	24.2	16.5	$25.8^\dagger$	14.5	19.6	$26.7^\dagger$
GP-TDNN1	1	$11.6^\dagger$	$23.5^\dagger$	$16.2^\dagger$	$25.4^\dagger$	14.4	$18.8^\dagger$	$26.6^\dagger$
GP-TDNN2	1	$11.6^\dagger$	$23.6^\dagger$	$15.9^\dagger$	$25.0^\dagger$	$13.9^\dagger$	$18.8^\dagger$	$26.1^\dagger$
GP-TDNN3	1	$11.7^\dagger$	$23.7^\dagger$	$16.0^\dagger$	$25.1^\dagger$	14.1	$18.4^\dagger$	$25.4^\dagger$
V-TDNN	1	$11.6^\dagger$	24.0	16.3	$25.5^\dagger$	14.2	19.6	27.1

[†]Denotes a Statistically Significant Difference is Obtained Over the TDNN Baseline System (Line 1). (SWB1 and CHM Denote the Switchboard and Callhm Subsets of the Hub5' 00 Test Set; FSH and SWB2 Denote the Fisher and Switchboard Subsets of the RT03S Test Set; SWB3, SWB4 and SWB5 Denote Three Switchboard Subsets in the Rt02 Test Set.)

1) Experiments conducted on the Bayesian TDNN systems (B-TDNN, Sec. IV-A, line 2-6 in Table III) show that Bayesian estimation only needs to be applied at the first layer, as further applying the Bayesian estimation of any subsequent higher layers produces no additional improvement. This confirms the hypothesis previously discussed in Sec. IV-E that the modelling uncertainty associated with the lower layers of TDNN systems will be much larger than those of the higher layers. Based on this set of experiments, the uncertainty is considered at the first layer of all Bayesian learning based TDNN systems in the rest of the paper.

2) Compared with the LF-MMI trained TDNN system (line 1 in Table III), the Bayesian TDNN system (B-TDNN, Sec. IV-A, line 2 in Table III) consistently produces a lower WER across all three test sets. For example, the largest absolute WER reduction (1.8%) is obtained on the SWB5 subset of the Rt02 test set.

3) The Gaussian Process TDNN systems (GP-TDNN, Sec. IV-C, line 8-11 in Table III) outperformed the Bayesian TDNN system (B-TDNN, Sec. IV-A, line 2 in Table III) on the Rt02 test set by 0.2% (SWB3 subset) to 0.8% (SWB4 subset) absolute WER reduction, while the Variational TDNN system (V-TDNN, Sec. IV-D, line 12 in Table III) was outperformed by the Bayesian TDNN system (B-TDNN, Sec. IV-A, line 2 in Table III).

4) The Bayesian Dropout TDNN system (BD-TDNN, Sec. IV-B, line 7 in Table III) achieves similar performance on three test sets as the Bayesian TDNN system (B-TDNN, Sec. IV-A, line 2 in Table III). Based on these results, the Bayesian Dropout TDNN system is not considered in the

following large-scale experiments conducted on the 900-hour speed-perturbed Switchboard corpus in Sec. V-B.

5) Performance comparison of varying model sizes in Fig. 3 suggests that Bayesian and Gaussian Process TDNN systems are more robust against the change of model sizes, in particular over more complex systems containing up to 250 million parameters (14.7 times of the baseline TDNN system size in line 1 of Table III).

B. Experiments on 300-Hour Switchboard Task

To fully evaluate the performance of the Bayesian estimated LF-MMI TDNN systems, experiments were further conducted on the 300-hour (900 h after speed perturbation) Switchboard conversational English telephone speech recognition task.

Task Description: The Switchboard I telephone speech corpus consists of approximately 300 hours audio data released by LDC (LDC97S62). The baseline GMM-HMM system with 6008 tied tri-phone states was trained based on 40-dimensional Mel-frequency cepstral coefficients (MFCCs) to generate alignments for the neural network training. The performance of LF-MMI trained TDNN baseline system incorporated with i-Vector [74] and speed perturbation was shown in line 1 of Table IV. In addition, the effects of LHUC [49] speaker adaptation were investigated. For performance evaluation, Kaldi recipe LSTM recurrent neural network language model (RNNLM) trained on the Switchboard and Fisher transcripts (LDC2004T19, LDC2005T19) was used to rescore the nbest lists produced by the LF-MMI trained systems with a four-gram language model (LM).

Experimental Results and Analysis: Three main trends can be found in the results of Table IV and Table V.

1) The Bayesian TDNN system (B-TDNN, Sec. IV-A, line 9 in Table IV) consistently outperforms the TDNN baseline system (line 8 in Table IV) across all three test sets. For example, 0.9% absolute WER reduction was achieved on the SWB5 subset of Rt02 test set. When compared with the Bayesian TDNN system (B-TDNN, Sec. IV-A, line 9 in Table IV), the Gaussian Process TDNN system (GP-TDNN2, Sec. IV-C, line 12 in Table IV) produced by up to 0.5% absolute WER reduction on the SWB3 subset of Rt02 test set. On this task, the varational TDNN (V-TDNN, Sec. IV-D, line 14 in Table IV) made no significant improvements over the TDNN baseline system (line 8 in Table IV). The hidden output distribution in the variational neural network depends on the input data on a frame-by-frame time varying basis. This may introduce undesired artefacts in the resulting hidden layer outputs that are expected to be more invariant to variability in data. This may in part explain the performance difference between V-TDNN and B-TDNN/GP-TDNN systems consistently found in the experiments of this paper.

2) By further incorporating LHUC or Kaldi recipe LSTM recurrent neural network language model (RNNLM) or both of them, similar performance improvements can still be maintained. Statistically significant WER reductions of

TABLE IV
PERFORMANCE (WER%) COMPARISON OF TDNN, B-TDNN AND GP-TDNN SYSTEMS ON THE **HUB5' 00**, **RT03S** AND **RT02** TEST SETS BEFORE AND AFTER APPLYING LHUC AND RNNLM RESCORING

	System	I-Vector	Speed perturb	LHUC	LM	Hub5' 00		Rt03S		Rt02		
						SWB1	CHM	FSH	SWB2	SWB3	SWB4	SWB5
1	TDNN	✓	✗	✓	+RNNLM	7.8	15.2	9.9	16.2	9.1	12.1	17.0
2	B-TDNN					7.6	15.0	9.8	15.8†	9.0	12.1	**15.6**†
3	GP-TDNN0					7.5†	14.9†	**9.5**†	15.7†	9.0	12.0	15.9†
4	GP-TDNN1	✓	✗	✓	+RNNLM	**7.3**†	15.0	9.6†	**15.3**†	**8.7**†	12.2	**15.6**†
5	GP-TDNN2					**7.3**†	**14.5**†	9.8	15.5†	9.0	12.0	**15.6**†
6	GP-TDNN3					7.5†	14.8†	9.8	15.6†	9.2	12.1	16.2†
7	V-TDNN					7.5†	14.9†	9.8	15.8†	**8.7**†	12.3	16.5†
8	TDNN	✓	✓	✗	4-gram	9.7	18.0	12.6	19.5	11.5	15.3	20.0
9	B-TDNN					9.4	17.3†	12.1†	19.2	11.4	14.7†	19.1†
10	GP-TDNN0					**9.2**†	17.6†	12.0†	19.1†	**10.8**†	14.6†	19.3†
11	GP-TDNN1	✓	✓	✗	4-gram	**9.2**†	17.2†	11.9†	**19.0**†	11.0†	14.6†	19.5
12	GP-TDNN2					9.3	17.2†	**11.8**†	19.1†	11.0†	**14.4**†	**19.1**†
13	GP-TDNN3					9.5	17.3†	12.0†	19.1†	11.1†	14.8†	19.2†
14	V-TDNN					9.5	17.9	12.5	19.6	11.5	15.2	19.6
15	TDNN	✓	✓	✓	4-gram	9.5	17.6	12.1	19.0	11.0	14.8	19.1
16	B-TDNN					**9.2**†	17.1†	11.6†	**18.1**†	10.8	14.0†	17.9†
17	GP-TDNN0					9.0†	17.1†	11.6†	**18.1**†	**10.4**†	14.2†	17.9†
18	GP-TDNN1	✓	✓	✓	4-gram	9.1†	17.0†	11.9	18.7	10.7	14.4	17.9†
19	GP-TDNN2					9.3	**16.8**†	11.5†	**18.1**†	10.8	**13.9**†	18.0†
20	GP-TDNN3					**9.2**†	16.9†	**11.3**†	**18.1**†	10.6†	14.1†	**17.7**†
21	V-TDNN					9.3	17.6	12.0	19.1	11.1	14.6	18.8
22	TDNN	✓	✓	✗	+RNNLM	8.1	15.6	10.4	17.2	9.9	13.0	17.3
23	B-TDNN					7.7†	**14.7**†	10.2	16.6†	9.5	12.6†	16.7†
24	GP-TDNN0					7.8†	15.3†	10.1†	16.6†	**9.0**†	12.4†	16.8†
25	GP-TDNN1	✓	✓	✗	+RNNLM	7.7†	15.1†	**10.0**†	16.6†	9.3†	12.5†	16.7†
26	GP-TDNN2					7.9	**14.7**†	10.1†	**16.3**†	9.2†	**12.3**†	**16.2**†
27	GP-TDNN3					7.8†	15.1†	10.1†	16.4†	9.4†	**12.3**†	16.4†
28	V-TDNN					8.1	15.6	10.5	17.1	9.6	13.0	16.9
29	TDNN	✓	✓	✓	+RNNLM	7.9	15.2	10.1	16.3	9.5	12.4	16.1
30	B-TDNN					**7.4**†	14.6†	9.6†	15.4†	8.8†	11.9†	15.4†
31	GP-TDNN0					7.5†	14.8†	9.9†	15.6†	8.9†	12.1†	15.1†
32	GP-TDNN1	✓	✓	✓	+RNNLM	7.5†	14.9†	9.7†	15.5†	8.9†	11.6†	14.6†
33	GP-TDNN2					7.6†	**14.2**†	9.4†	15.1†	**8.7**†	11.7†	**14.3**†
34	GP-TDNN3					7.5†	14.4†	**9.3**†	15.1†	8.9†	**11.5**†	14.9†
35	V-TDNN					8.0	15.3	10.1	16.3	9.3	12.6	16.0

†Denotes a Statistically Significant Difference is Obtained Over the TDNN Baseline System (Line 1, 8, 14, 21). (SWB1 and CHM Denote the Switchboard and Callhm Subsets of the Hub5' 00 Test Set; FSH and SWB2 Denote the Fisher and Switchboard Subsets of the RT03S Test Set; SWB3, SWB4 and SWB5 Denote Three Switchboard Subsets in the Rt02 Test Set.)

TABLE V
PERFORMANCE CONTRASTS OF LHUC ADAPTED TDNN, B-TDNN, GP-TDNN SYSTEMS RESCORED BY LARGE RNNLMS AGAINST OTHER STATE-OF-THE-ART SYSTEMS CONDUCTED ON THE 300-HOUR SWITCHBOARD TASK. THE OVERALL WERS IN "()" ARE NOT REPORTED BY THE ORIGINAL PAPERS AND ARE RECALCULATED USING THE SUBSET WERS

|	System	#Param	Hub5' 00			Rt03S		
			SWB1	CHM	Avg.	FSH	SWB2	Avg.
1	RWTH SMBR BLSTM [75]	-	6.7	14.7	10.7	-	-	-
2	+ Affine transform based environment adaptation	-	6.7	13.5	10.2	-	-	-
3	Google Listen, Attend and Spell network + SpecAugment [30]	-	6.8	14.1	(10.5)	-	-	-
4	IBM LSTM based Attention encoder-decoder	29M	7.4	14.6	(11.0)	-	-	-
5	+ SpecAugment + weight noise [51]	75M	6.8	13.4	(10.1)	-	-	-
6		280M	6.4	12.5	(9.5)	8.4	14.8	(11.7)
7	LF-MMI TDNN + LHUC + Large RNNLM	19M	7.5	14.5	11.0	9.5	15.5	12.6
8	LF-MMI B-TDNN + LHUC + Large RNNLM		7.0	13.9	10.5	9.0	14.4	11.8
9	LF-MMI GP-TDNN0 + LHUC + Large RNNLM		7.2	14.1	10.7	9.4	14.9	12.2
10	LF-MMI GP-TDNN1 + LHUC + Large RNNLM	19M	7.3	14.1	10.7	9.1	15.0	12.2
11	LF-MMI GP-TDNN2 + LHUC + Large RNNLM		7.2	13.8	10.6	9.0	14.4	11.8
12	LF-MMI GP-TDNN3 + LHUC + Large RNNLM		7.2	13.6	10.4	8.9	14.4	11.8
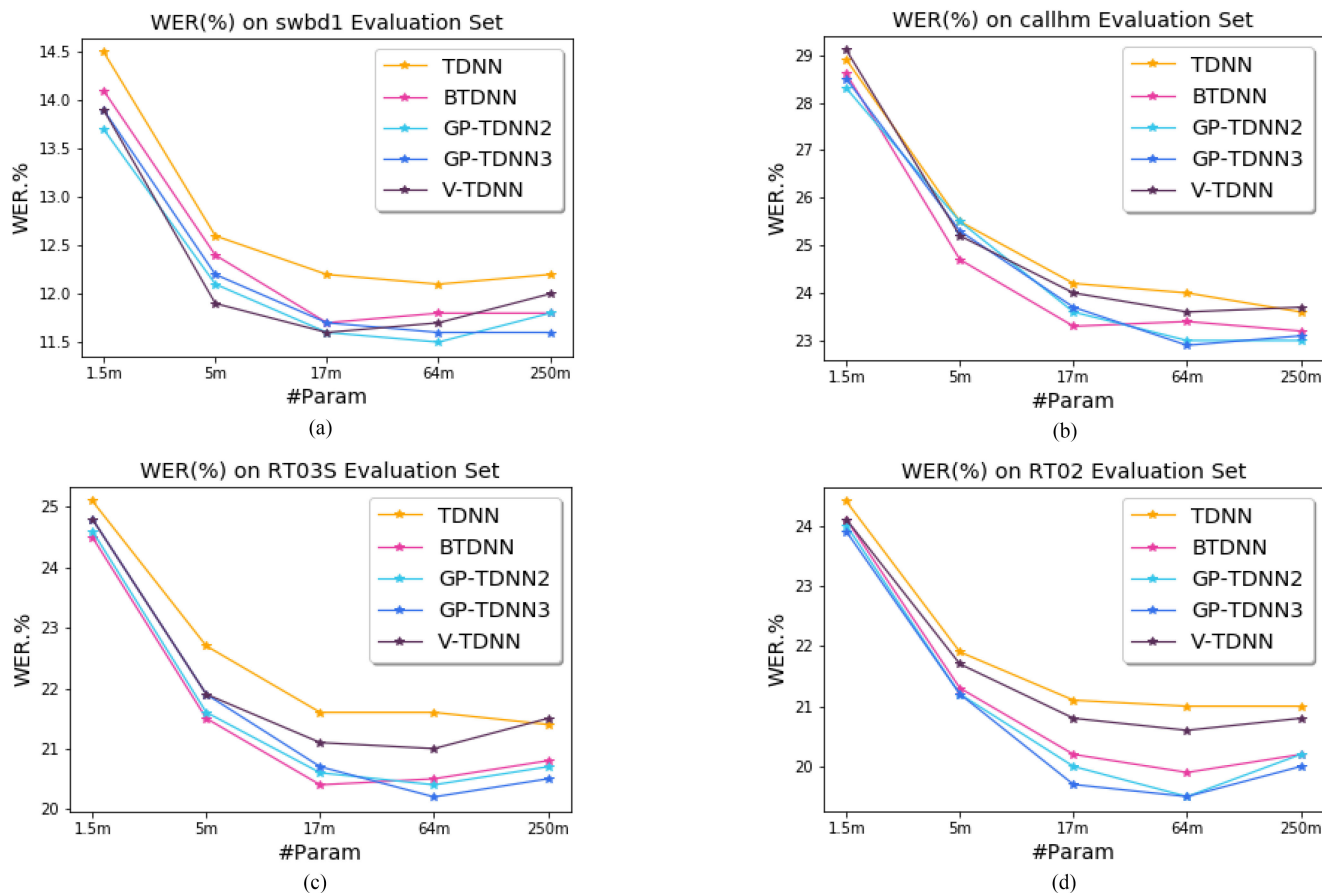
Fig. 3. Performance comparison of different TDNN, B-TDNN, GP-TDNN2, GP-TDNN3 and V-TDNN systems with equal model complexity on the **HUB5' 00** (a,b), **Rt03S** (c) and **Rt02** (d) test sets. The model size (measured in the number of parameters) is increased from 1.5 m to 5 m, 17 m, 64 m and up to 250 m by varying the linear, projection and affine layers dimensionality. The standard TDNN system (line 1 in Table III) containing 17 m parameters is shown in the middle of the four figures.

0.3% (SWB1 subset of Hub5'00 test set) to 1.8% (SWB5 subset of Rt02 test set) absolute (4% to 11% relative) were obtained by the Gaussian Process TDNN system (GP-TDNN2, Sec. IV-C, line 33 in Table IV) over the TDNN baseline system (line 29 in Table IV).

3) The experimental results show that the proposed Bayesian TDNN, Gaussian Process TDNN systems (line 2-6, line 30-34 in Table IV) significantly outperform the TDNN baseline systems (line 1, line 29 in Table IV) with and without speed perturbation. This suggests that the proposed method and data augmentation are mostly complementary and their improvements largely additive.

4) In order to further evaluate the best performing Bayesian trained systems in Table IV, the LHUC adapted baseline TDNN (line 29 in Table IV), Bayesian TDNN (B-TDNN, line 30 in Table IV) and Gaussian Process TDNNs (GP-TDNN, line 31-34 in Table IV) were evaluated using a larger RNNLM. The performance of these systems are shown in Table V. These are then compared with the state-of-the-art performance obtained on the Switchboard task using the most recent hybrid and end-to-end systems reported in the literature (line 1-6 in Table V). Two larger LSTM recurrent neural network language models

(RNNLMs)[3] performing forward and backward contexts based word prediction respectively with twice the number of LSTM cells (2048) and projection dimensionality (1024) compared with the smaller LSTM RNNLM used in Table III were trained. System (1) and System (2) in Table V were RWTH BLSTM hybrid systems without and with affine transformation for environment adaptation [75]. System (3) was the Google Listen, Attend and Spell end-to-end system built with SpecAugment [30]. System (4)-(6) were the IBM LSTM based attention encoder-decoder end-to-end systems built with SpecAugment and weight noise [51]. Competitive performance is achieved by the Bayesian estimated TDNN systems (line 8-12 in Table V) on the CHM subset of Hub5'00 test set and Rt03S test set when compared with the state-of-the-art systems (line 1-6 in Table V). A general trend can be observed in Table V such that our B-TDNN and GP-TDNN systems can produce WERs similar to state-of-the-art end-to-end systems with much fewer parameters. For example, by achieving the similar 13.6% WER on the CHM subset

[3]Dropout operation with 85% retention was applied to the output nodes of each layer.

TABLE VI
PERFORMANCE (CER%) COMPARISON OF TDNN, B-TDNN AND GP-TDNN SYSTEMS ON THE HKUST MANDARIN CONVERSATIONAL TELEPHONE SPEECH **DEV**, NIST MANDARIN **RT03S** AND **EVAL97** EVALUATION SETS

	System	I-Vector	Speed perturb	LHUC	Dev	Rt03S	Eval97
1	TDNN	✓	✓	✗	24.6	37.2	37.1
2	B-TDNN				**23.6**	**36.2**†	**36.1**†
3	GP-TDNN0				24.5	37.0†	36.7†
4	GP-TDNN1	✓	✓	✗	24.0	36.4†	36.6†
5	GP-TDNN2				23.9	**36.2**†	36.5†
6	GP-TDNN3				24.0	36.6†	36.4†
7	TDNN	✓	✓	✓	24.1	36.4	36.5
8	B-TDNN				**23.3**	35.4†	**35.3**†
9	GP-TDNN0				23.5	35.9†	35.6†
10	GP-TDNN1	✓	✓	✓	**23.3**	**35.1**†	35.5†
11	GP-TDNN2				23.4	35.3†	35.5†
12	GP-TDNN3				23.6	35.6†	35.9

†Denotes a Statistically Significant Difference is Obtained Over the TDNN Baseline System (Line 1, 7). Note That Significant Different Test Was Not Performed on the **Dev** Test Set

TABLE VII
PERFORMANCE (WER%) OF THE LF-MMI TRAINED TDNN SYSTEM ON THE LIBRISPEECH **DEV** AND **TEST** TEST SETS

System	I-Vector	Speed perturb	Dev		Test	
			clean	other	clean	other
TDNN	✓	✓	3.56	10.0	4.0	10.3

Sec. IV-C) over the Bayesian TDNN system (B-TDNN, Sec. IV-A, line 2 in Table IV).

(2) By further incorporating LHUC based speaker adaptation, similar performance improvements can still be maintained. The best performance was achieved by the B-TDNN system (Sec. IV-A, line 8 in Table VI) and GP-TDNN1 (Sec. IV-C, line 10 in Table VI). For example, up to 1.3% absolute CER reduction was achieved on the Rt03S set by GP-TDNN1 system (Sec. IV-C, line 10 in Table VI) when compared with TDNN baseline system (line 7 in Table VI).

D. Experiments on DementiaBank Pitt Elderly Speech

We further evaluate the performance of Bayesian estimated LF-MMI TDNN systems on a cross domain adaptation task which requires porting a 1000-h LibriSpeech corpus trained LF-MMI TDNN system to an elderly speech recognition task based on the DementiaBank Pitt database.

Task Description: The DementiaBank Pitt corpus[5] [79] contains 33-hour audio data, which was split into 27.16-hour training data and 5.81-hour test data. The training data segmentation refinement was first performed by removing excessive silence at the start and end of each utterance in the DementiaBank Pitt corpus. After silence stripping, the DementiaBank Pitt corpus contains 15.75-hour training data (9.72-hour elderly participant data + 6.03-hour investigator data) and 3.14-hour test data (1.93-hour elderly participant data + 1.21-hour investigator data). A GMM-HMM baseline was trained with 39-dimensional PLP features following the same procedures described in V-A. A 4-gram language model based on the DementiaBank Pitt transcripts, Switchboard and Fisher transcripts and additional text data of 392.4 millions words from the Gigaword collection released by LDC (LDC2011T07) was used in decoding. More details can be found in [80]. The TDNN baseline system was trained with the DementiaBank Pitt data only and its performance was shown in line 7 of Table V-D. Speed perturbation was also applied to expand the training data to 59 hours in total. The performance of the TDNN system trained on the augmented 59-hour data was shown in line 10 of Table V-D. The Librispeech corpus [81] contains 1000 hours of English read speech. Table VII shows the performance of the Librispeech based LF-MMI TDNN system.[6] During domain adaptation, the fine-tuning adapted baseline TDNN systems (line 2,5,8,11 in Table VIII) reinitialized the input and the output layer of the LibriSpeech corpus trained LF-MMI TDNN model, while the Bayesian adapted B-TDNN

of the Hub5'00 test set, our GP-TDNN3 system only needs 25% number of parameters of the IBM system (5) producing a comparable WER 13.4%. Furthermore, our GP-TDNN3 system achieves a state-of-the-art WER of 11.8% on the Rt03S test set with 93.2% parameter size reduction when compared with the IBM system (6).

C. Experiments on 150-Hour HKUST Task

The performance of Bayesian estimated TDNN systems are further evaluated on a 150-hour (450 h after speed perturbation) HKUST conversational Mandarin telephone speech recognition task.

Task Description: The HKUST Mandarin Telephone Speech contains 150 hours training data released by LDC (LDC2005S15, LDC2005T32). The development set released in 2014 [76] was used as the validation set. The NIST Rt03S (LDC2007S10) and 1997 NIST Hub5 Mandarin evaluation set [77] were used to form a 2.7-hour test set. Based on the speaker adaptive training (SAT) [71]–[73], a GMM-HMM baseline system with 4000 tied triphone states was trained[4] with 40-dimensional MFCCs and 3-dimensional pitch features [78] to generate the alignment for the neural network training. The tri-gram language model trained with the training data transcript (LDC2005T32) was used in decoding. All our recognition results were evaluated based on Character Error Rate (CER).

Experimental Results and Analysis Two similar trends observed in Table IV can also be found in Table VI.

(1) Compared with the TDNN baseline system (line 1 in Table VI), the Bayesian TDNN system (B-TDNN, Sec. IV-A, line 2 in Table IV) produced 1% absolute CER reduction across all three test sets. No additional CER improvement was further obtained on the Gaussian Process TDNN systems (GP-TDNN, line3-6 in Table VI,

---

[4]Following the published Kaldi code at github.com/kaldi-asr/kaldi/egs/hkust/s5/run.sh

[5]https://dementia.talkbank.org/access/English/Pitt.html
[6]Following the setup in github.com/kaldi-asr/kaldi/egs/librispeech/s5/run.sh and github.com/kaldi-asr/kaldi/egs/librispeech/s5/local/chain/run _ tdnn.sh

TABLE VIII
PERFORMANCE (WER%) COMPARISON OVER THE DEMENTIABANK PITT CORPUS TRAINED TDNN SYSTEMS, FINE-TUNING ADAPTED BASELINE TDNN SYSTEMS AND BAYESIAN ADAPTED B-TDNN SYSTEMS ON THE **PARTICIPANT** AND **INVESTIGATOR** TEST SETS BY USING 4-HOUR, 8-HOUR 16-HOUR SUBSET OF THE PITT DATA OR AN AUGMENTED 55-HOUR PITT DATA SET

Row	System	I-Vector	Speed perturb	Domain Adaptation	Training Data	Participant	Investigator	All
1	TDNN			✗		62.67	28.11	47.57
2	TDNN	✗	✗	✓	4h	58.52	26.45	44.51†
3	B-TDNN			✓		**55.95**	**24.98**	**42.02**†
4	TDNN			✗		52.79	23.99	40.21
5	TDNN	✗	✗	✓	8h	51.08	22.69	38.68†
6	B-TDNN			✓		**50.32**	**22.14**	**38.00**†
7	TDNN			✗		47.18	21.24	35.84
8	TDNN	✗	✗	✓	16h	45.71	20.15	34.54†
9	B-TDNN			✓		**44.80**	**19.82**	**33.88**†
10	TDNN			✗		43.88	19.84	33.37
11	TDNN	✓	✓	✓	59h	44.08	19.73	33.44
12	B-TDNN			✓		**42.53**	**19.20**	**32.33**†

†Denotes a Statistically Significant Difference is Obtained Over the TDNN Baseline System (Line 1, 4, 7, 10).

systems (B-TDNN, Sec. IV-A, line 3,6,9,12 in Table VIII) replaced the first layer of the LibriSpeech corpus trained TDNN model with the Bayesian layer and reinitialized the output layer. The baseline TDNN system was cross domain adapted to the Pitt data using parameter fine-tuning, while the B-TDNN system was Bayesian adapted to the same data. The fine-tuning adapted baseline TDNN system serves as the prior of the Bayesian adapted B-TDNN system.

Experimental Results and Analysis Performance comparison between the fine-tuning adapted baseline TDNN and Bayesian adapted B-TDNN systems was shown in Table VIII. Two main trends can be concluded.

1) By using different amounts of training data, the Bayesian adapted B-TDNN systems (B-TDNN, Sec. IV-A, line 3, 6, 9, 12 in Table VIII) consistently outperform the fine-tuning adapted baseline TDNN systems (line 2, 5, 8, 11 in Table VIII). The best performance was obtained on the Bayesian adapted B-TDNN system (line 12 in Table VIII) using the augmented 59-hour DementiaBank Pitt data. This corresponds to a total 1.1% absolute WER reduction over the fine-tuning adapted baseline TDNN system (line 11 in Table VIII).

2) When using the 4-hour subset of the Pitt data for cross adaptation, the largest WER absolute reduction up to 2.5% was obtained by the Bayesian adapted B-TDNN system (B-TDNN, Sec. IV-A, line 3 in Table VIII) over the fine-tuning adapted baseline TDNN system (line 2 in Table VIII).

VI. CONCLUSION

This paper presents a full Bayesian framework to account for model uncertainty in sequence discriminative training of factored TDNN acoustic models. Several Bayesian learning based TDNN variant systems are proposed to model the uncertainty over weight parameters and choices of hidden activation functions, or the hidden layer outputs. Efficient variational inference approaches using as few as one single parameter sample ensure their computational cost in both training and evaluation time

comparable to that of the baseline TDNN systems. The dropout technique is reformulated as a special case of Bayesian TDNN systems.

Experiments conducted on a state-of-the-art 900-h speed perturbed Switchboard corpus suggests the proposed Bayesian TDNN, Gaussain Process TDNN and variational TDNN systems consistently outperform the LF-MMI trained TDNN baseline systems by a statistically significant margin of 0.4%-1.8% absolute (5%–11% relative) reduction in word error rate over the NIST Hub5'00, RT02 and RT03 test sets. Similar consistent performance improvements were also obtained on a 450-h (with speed perturbation) HKUST conversational Mandarin telephone speech recognition task. On a third cross domain adaptation task requiring rapidly porting a 1000-h LibriSpeech data trained system to a 10-h Dementia Bank elderly speech corpus, the proposed Bayesian TDNN LF-MMI systems outperformed the baseline TDNN system domain adapted using direct weight fine-tuning by 1.1% absolute WER reduction.

The proposed Bayesian learning methods applied to TDNNs benefit from a distinct advantage of the underlying latent variable distributions estimation being fully integrated with the overall system training consistently using the same sequence level MMI error cost function. This is in contrast to many existing regularization techniques employed in state-of-the-art speech recognition systems including, but not limited to, Gaussian-based weight noise [38], [39], L2 norm [8] and model averaging [37]. More specifically, the Gaussian-based weight noise is normally kept fixed and not learnable. L2 norm regularization may be considered as a special form of maximum a posteriori (MAP) estimation [82], which restricts the estimation of weight parameters using a fixed Gaussian prior distribution of zero mean and unit variance. Model averaging that is currently used as a standard regularization method in the Kaldi toolkit [10] can be viewed as averaging over the weight parameters drawn from an unknown distribution that model the parameter estimates obtained at different training epochs or intervals.

Experimental results obtained across three task domains suggest among all the techniques presented in this paper, the proposed Bayesian TDNNs and Gaussian Process TDNNs

(GP-TDNN2 and GP-TDNN3 variants in particular) consistently outperform the baseline TDNN systems featuring state-of-the-art configurations including multiple regularization methods, data augmentation, speaker adaptation and RNNLM rescoring, and therefore are worth further studying on end-to-end speech recognition systems.

We would also like to note that Gaussian Process TDNNs have the additional ability of modelling neural architecture uncertainty in terms of the suitable activation functions to be used in TDNNs. This unique advantage of GP-TDNNs is as expected and also precisely one of the strengths traditionally associated with Gaussian Processes that is well known for providing powerful non-parametric modelling and black box optimization, for example, in the context of auto-configured Bayesian neural architecture search [83].

REFERENCES

[1] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 86, 1986, pp. 49–52.

[2] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2008, 2008, pp. 4057–4060.

[3] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2005.

[4] J. Kaiser, B. Horvat, and Z. Kacic, "A novel loss function for the overall risk criterion based discriminative training of HMM models," in *Proc. 6th Int. Conf. Spoken Lang. Process.*, 2000, pp. 887–890.

[5] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proc. 9th Int. Conf. Spoken Lang. Process.*, 2006.

[6] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2007, pp. IV-321–IV-324.

[7] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2009, pp. 3761–3764.

[8] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH*, vol. 2013, 2013, pp. 2345–2349.

[9] H. Su, G. Li, D. Yu, and F. Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 6664–6668.

[10] D. Povey et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *INTERSPEECH*, 2016, pp. 2751–2755.

[11] H. Sak et al., "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in *Proc. INTERSPEECH*, 2014, pp. 1209–1213.

[12] W. Michel, R. Schlüter, and H. Ney, "Frame-level MMI as a sequence discriminative training criterion for LVCSR," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6904–6908.

[13] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend, and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2016, pp. 4960–4964.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[15] A. Graves, "Sequence transduction with recurrent neural networks," 2012, *arXiv:1211.3711*.

[16] Y. Wang et al., "Transformer-Based acoustic modeling for hybrid speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6874–6878.

[17] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free MMI," in *INTERSPEECH*, 2018, pp. 12–16.

[18] W. Michel, R. Schlüter, and H. Ney, "Comparison of lattice-free and lattice-based sequence discriminative training criteria for LVCSR," in *Proc. INTERSPEECH*, 2019, pp. 1601–1605.

[19] A. Waibel, "Consonant recognition by modular construction of large phonemic time-delay neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 1989, pp. 215–223.

[20] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. INTERSPEECH*, 2015, pp. 1–5.

[21] D. Povey et al., "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *INTERSPEECH*, 2018, pp. 3743–3747.

[22] C. Lüscher et al., "Rwth asr systems for librispeech: Hybrid vs attention-w/o data augmentation," 2019, *arXiv:1905.03072*.

[23] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schlüter, and H. Ney, "The RWTH ASR system for Ted-Lium release 2: Improving hybrid HMM with specaugment," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 7839–7843.

[24] J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, and S. Liu, "On the comparison of popular end-to-end models for large scale speech recognition," in *INTERSPEECH*, pp. 1–5, 2020.

[25] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young, "Mmie training of large vocabulary recognition systems," *Speech Commun.*, vol. 22, no. 4, pp. 303–314, 1997.

[26] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 25–47, 2002.

[27] J. Benesty, J. Chen, and Y. Huang, "Automatic speech recognition: A deep learning approach," 2008.

[28] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2002, pp. I-105–I-108.

[29] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio Augmentation for Speech Recognition," in *INTERSPEECH*, pp. 3586–3589, 2015.

[30] D. S. Park et al., "Specaugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, pp. 2613–2617, 2019.

[31] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 5220–5224.

[32] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012.

[33] T. N. Sainath, L. Horesh, B. Kingsbury, A. Y. Aravkin, and B. Ramabhadran, "Accelerating hessian-free optimization for deep neural networks by implicit preconditioning and sampling," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 303–308.

[34] I.-H. Chung et al., "Parallel deep neural network training for big data on blue Gene/Q," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 6, pp. 1703–1714, Jun. 2017.

[35] H. H. Yang and S.-i. Amari, "Complexity issues in natural gradient descent method for training multilayer perceptrons," *Neural Comput.*, vol. 10, no. 8, pp. 2137–2157, 1998.

[36] N. L. Roux, P.-A. Manzagol, and Y. Bengio, "Topmoumoute online natural gradient algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 849–856.

[37] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," in *3rd Int. Conf. Learn. Representations, Workshop Track Proc.*, 2015.

[38] A. F. Murray and P. J. Edwards, "Enhanced mlp performance and fault tolerance resulting from synaptic weight noise during training," *IEEE Trans. Neural Netw.*, vol. 5, no. 5, pp. 792–802, Sep. 1994.

[39] S. Braun and S.-C. Liu, "Parameter uncertainty for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 5636–5640.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," in *Proc. J. Mach. Learn. Res.*, 2014, 1929–1958.

[41] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33nd Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[42] M. W. Lam *et al.*, "Gaussian Process Neural Networks for Speech Recognition," in *INTERSPEECH*, 2018, pp. 1778–1782.

[43] S. Hu *et al.*, "Bayesian and gaussian process neural networks for large vocabulary continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 6555–6559.

[44] S. Hu *et al.*, "LF-MMI training of Bayesian and Gaussian process time delay neural networks for speech recognition," in *INTERSPEECH*, 2019, pp. 2793–2797.

[45] X. Xie, X. Liu, T. Lee, S. Hu, and L. Wang, "BLHUC: Bayesian learning of hidden unit contributions for deep neural network speaker adaptation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 5711–5715.

[46] X. Li *et al.*, "Bayesian x-vector: Bayesian neural network based x-vector system for speaker verification," *ISCA SPLC-ODYSSEY*, 2020.

[47] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[48] S. Madikeri, S. Dey, P. Motlicek, and M. Ferras, "Implementation of the standard i-vector system for the kaldi speech recognition toolkit," *Idiap, Tech. Rep.*, 2016.

[49] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. IEEE Spoken Lang. Technol. Workshop.* 2014, pp. 171–176.

[50] J.-T. Chien and Y.-C. Ku, "Bayesian recurrent neural network for language modeling," *IEEE Trans, Neural, Netw. Learn. Syst.*, vol. 27, no. 2, pp. 361–374, Feb. 2016.

[51] Z. Tüske, G. Saon, K. Audhkhasi, and B. Kingsbury, "Single headed attention based sequence-to-sequence model for state-of-the-art results on switchboard-300," *arXiv:2001.07263.*

[52] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for ASR using LF-MMI trained neural networks," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2017, pp. 279–286.

[53] R. M. Neal, *Bayesian Learning for Neural Networks*. Springer, vol. 118, 2012.

[54] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, 1992.

[55] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[56] A. Graves, "Practical variational inference for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2348–2356.

[57] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," *Ser. Proceedings Machine Learning Research*, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07-09 Jul 2015, pp. 1613–1622.

[58] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press Cambridge, vol. 1, 2006.

[59] R. M. Neal, "Priors for infinite networks," in *Proc. Bayesian Learn. Neural Netw.*, 1996, pp. 29–53.

[60] T. Hazan and T. Jaakkola, "Steps toward deep kernel methods from infinite neural networks," 2015, *arXiv:1508.05133.*

[61] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as Gaussian processes," *arXiv:1711.00165.*

[62] A. Damianou and N. Lawrence, "Deep Gaussian processes," in *Proc. Artif. Intell. Statist.*, 2013, pp. 207–215.

[63] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.

[64] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2980–2988.

[65] J.-T. Chien *et al.*, "Variational recurrent neural networks for speech separation," in *Proc. 18th Annu. Conf. International Speech Commun. Assoc.*, 2017, pp. 1193–1197.

[66] J. Yu *et al.* "Comparative study of parametric and representation uncertainty modeling for recurrent neural network language models," in *INTERSPEECH*, 2019, pp. 3510–3514.

[67] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust. Speech, Signal, Process.*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[68] Y. LeCun *et al.* "Convolutional networks for images, speech, and time series," *Handbook Brain Theory Neural Netw.*, vol. 3361, no. 10, 1995, Art. no. 1995.

[69] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 9, no. 2, pp. 171–185, 1995.

[70] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech, Lang.*, vol. 12, no. 2, pp. 75–98, Apr. 1998.

[71] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. 4th IEEE Int. Conf. Spoken Lang. Process.*, vol. 2, 1996, pp. 1137–1140.

[72] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: A maximum likelihood approach to speaker normalization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2, 1997, pp. 1043–1046.

[73] D. Povey, H.-K. J. Kuo, and H. Soltau, "Fast speaker adaptive training for speech recognition," in *Proc. 9th Annu. Conf. Int. Speech Commun. Assoc.*, 2008, pp. 1245–1248.

[74] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[75] M. Kitza, P. Golik, R. Schlüter, and H. Ney, "Cumulative adaptation for BLSTM acoustic models," in *Proc. INTERSPEECH*, 2019, pp. 754–758.

[76] X. Liu, F. Flego, L. Wang, C. Zhang, M. J. Gales, and P. C. Woodland, "The cambridge university 2014 bolt conversational telephone mandarin chinese LVCSR system for speech translation," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3145–3149.

[77] C. Zhang and P. C. Woodland, "Parameterised sigmoid and relu hidden activation functions for DNN acoustic modelling," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015.

[78] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 2494–2498.

[79] J. T. Becker, F. Boiler, O. L. Lopez, J. Saxton, and K. L. McGonigle, "The natural history of alzheimer's disease: Description of study cohort and accuracy of diagnosis," *Arch. Neurol.*, 1994.

[80] Z. Ye, S. Hu, J. Li, "Development of the CUHK elderly speech recognition system for neurocognitive disorder detection using the dementiabank corpus," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021.

[81] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 5206–5210.

[82] Z. Huang, S. M. Siniscalchi, I. Chen, J. Li, J. Wu, and C. Lee, "Maximum a posteriori adaptation of network parameters in deep models," in *Proc. INTERSPEECH*, 2015, pp. 1076–1080.

[83] K. Kandasamy, W. Neiswanger, J. Schneider, B. Póczos, and E. P. Xing, "Neural architecture search with Bayesian optimisation and optimal transport," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2020–2029.
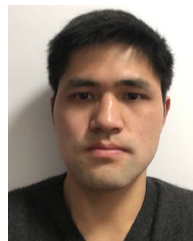
Shoukang Hu (Graduate Student Member, IEEE) received the B.E. degree in mechanical and electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2017. He is currently working toward the Ph.D. degree with The Chinese University of Hong Kong, Hong Kong. His current research interests include speech recognition, bayesian modeling, and neural architecture search.

Xurong Xie received the B.S. degree in mathematics and applied mathematics from Sun Yat-sen University, Guangzhou, China, in 2011, the M.S. degree in computational statistics and machine learning from the University College London, London, U.K., in 2012, and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 2020. He is currently a Research Assistant Professor with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Beijing, China. His research interests include adaptation techniques for acoustic modeling, speech processing for disorder speech, and statistic methods applied to speech technologies such as Bayesian learning.

Shansong Liu received the B.E. degree in automation from Sichuan University, Chengdu, China, in 2014 and the M.S. degree in control science and engineering from Tsinghua University, Beijing, China, in 2017. He is currently working toward the Ph.D. degree with The Chinese University of Hong Kong, Hong Kong. His current research interests include disordered speech recognition and multimodal speech recognition.

Jianwei Yu received the B.E. degree from Nanjing University, Nanjing, China, in 2017. He is currently working toward the Ph.D. degree with The Chinese University of Hong Kong, Hong Kong. His current research interests include language modeling, multi-model speech enhancement, and recognition.

Zi Ye received the B.S. and M.S. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2019. She is currently working toward the Ph.D. degree with The Chinese University of Hong Kong, Hong Kong. Her current research mainly focuses on automatic speech recognition for elderly speech and spoken language based early diagnosis for neurocognitive disorders.

Mengzhe Geng received the B.Sc. degree in 2019 in mathematics and information engineering from The Chinese University of Hong Kong, Hong Kong, where he is currently working toward the Ph.D. degree. His current research interests include data augmentation and speaker adaption.

Xunying Liu (Member, IEEE) received the undergraduate degree from Shanghai Jiao Tong University, Shanghai, China, and the M.Phil. degree in computer speech and language processing and the Ph.D. degree in speech recognition from the University of Cambridge, Cambridge, U.K. He was a Senior Research Associate with the Machine Intelligence Laboratory, Cambridge University Engineering Department, and since 2016, has been an Associate Professor with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong. His current research interests include machine learning, large vocabulary continuous speech recognition, statistical language modeling, noise robust speech recognition, audio-visual speech recognition, computer aided language learning, speech synthesis, and assistive technology. He is a Member of the ISCA. He and his students were the recipient of the number of best paper awards and nominations, including the Best Paper Award at ISCA Interspeech 2010 for the paper titled Language Model Cross Adaptation for LVCSR System Combination, and the Best Paper Award at IEEE ICASSP2019 for their paper titled BLHUC: Bayesian Learning of Hidden Unit Contributions for Deep Neural Network Speaker Adaptation.

Helen Meng (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA. In 1998, she joined The Chinese University of Hong Kong, Hong Kong, where she is currently the Chair Professor with the Department of Systems Engineering and Engineering Management. She was the former Department Chairman and the Associate Dean of research with the Faculty of Engineering. Her research interests include human–computer interaction via multimodal and multilingual spoken language systems, spoken dialog systems, computer-aided pronunciation training, speech processing in assistive technologies, health-related applications, and big data decision analytics. Between 2009 and 2011, she was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. She was also on the Elected Board Member of the International Speech Communication Association and an International Advisory Board Member. She is a ISCA, HKCS, and HKIE. She was the recipient of the IEEE Signal Processing Society Leo L. Beranek Meritorious Service Award in 2019.