Consistent DNN Uncertainty Training and Decoding for Robust ASR

Karan Nathwani, Emmanuel Vincent, Irina Illina

CONSISTENT DNN UNCERTAINTY TRAINING AND DECODING FOR ROBUST ASR

Karan Nathwani[1,2,3], *Emmanuel Vincent*[1,2,3], *Irina Illina*[1,2,3]

[1] Inria, Villers-lès-Nancy, F-54600, France
[2] Université de Lorraine, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54506, France
[3] CNRS, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54506, France

ABSTRACT

We consider the problem of robust automatic speech recognition (ASR) in noisy conditions. The performance improvement brought by speech enhancement is often limited by residual distortions of the enhanced features, which can be seen as a form of statistical uncertainty. Uncertainty estimation and propagation methods have recently been proposed to improve the ASR performance with deep neural network (DNN) acoustic models. However, the performance is still limited due to the use of uncertainty only during decoding. In this paper, we propose a consistent approach to account for uncertainty in the enhanced features during both training and decoding. We estimate the variance of the distortions using a DNN uncertainty estimator that operates directly in the feature maximum likelihood linear regression (fMLLR) domain and we then sample the uncertain features using the unscented transform (UT). We report the resulting ASR performance on the CHiME-2 and CHiME-3 datasets for different uncertainty estimation/propagation techniques. The proposed DNN uncertainty training method brings 4% and 8% relative improvement on these two datasets, respectively, compared to a competitive fMLLR-domain DNN acoustic modeling baseline.

***Index Terms*—** DNN, Robust ASR, Unscented transform, Uncertainty training, Uncertainty decoding.

1. INTRODUCTION

Robust automatic speech recognition (ASR) in noisy environments is still a challenging goal. Traditional front-end robust ASR approaches estimate enhanced features from the noisy speech signal, which are then passed to back-end approaches for decoding [1]. In general, the back-end approaches are able to compensate for distortions in the speech features by adapting the model parameters over the duration of one or more utterances. This can be done for instance by training the acoustic model on enhanced training data. However, the ASR performance at a given time still depends on the distortion at that specific time.

In order to address this issue, the idea of uncertainty decoding has emerged. During acoustic model scoring, the uncertainty decoding framework estimates the uncertainty (or variance) of speech distortion in the input features [2–4] in each time frame and modifies the acoustic scores accordingly. The uncertainty can be computed directly in the ASR feature domain [1, 5–10] or propagated from the spectral domain to the feature domain [11–17], under the assumption that it can be represented by Gaussian distribution. For Gaussian mixture model (GMM) based acoustic models, the expectation of the acoustic scores over this distribution can then be computed in closed form by adding the variance of the uncertainty to that of every Gaussian component [2–4]. The computation of this expectation for Deep Neural Network (DNN) acoustic models is less trivial due to the nonlinear activations. It can be approximated by numerical sampling techniques [18–22]. Among them, Monte Carlo (MC) sampling and the unscented transform (UT) have shown good performance on several datasets when applied to logmel features [10, 20–22], but the benefit of DNN uncertainty decoding remains to be proved for more advanced features, such as feature-domain maximum likelihood linear regression (fMLLR). Once computed, the acoustic scores are incorporated in the decoding algorithm.

While the theory of uncertainty decoding assumes that the acoustic model is trained on clean data and the estimated uncertainty matches the true variance of speech distortion, the estimated uncertainty often fails to capture some of the true variance of speech distortion in practice. Indeed, there remains a gap with the performance that could be obtained using the oracle (ground truth) uncertainty [10]. For this reason, applying uncertainty decoding to an acoustic model trained on clean data can underestimate the uncertainty and yield little performance improvement. Most authors reported improved performance by training the acoustic model on enhanced or noisy data instead [10, 20, 21]. This heuristic choice of training data overestimates the uncertainty: the variance of speech distortion is modeled by both the acoustic model and the uncertainty estimator and these two variances add up when performing uncertainty decoding. There is hence a need for an acoustic model training algorithm that accounts for the residual uncertainty in the training data that is not captured by the uncertainty estimator, such that the resulting uncertainty decoding is unbiased.

The authors in [23] proposed such an uncertainty training

algorithm for GMM acoustic models by using the GMM uncertainty decoding criterion for both training and test. They showed improved performance compared to classical training on either clean or enhanced data for a speaker identification task. The use of uncertainty for DNN acoustic model training was later explored in [24], however the authors used inconsistent uncertainty handling schemes, namely MC for training and recognizer output voting error reduction (ROVER) for test. Also, they assumed a heuristic uncertainty distribution based on linear interpolation between the noisy and enhanced feature vectors and they evaluated their approach with logmel features on simulated data only.

In this paper, we propose a principled approach to account for uncertainty in the enhanced features by sampling the input features using UT during both training and test. This can be thought of as a form of uncertainty-motivated training data augmentation. By contrast with [24], we handle uncertainty in a consistent way for training and test. Also, we attempt to estimate the actual uncertainty using an fMLLR-domain deep neural network uncertainty (DNNU) estimator and we evaluate the results on both simulated (CHiME-2) [25] and real (CHiME-3) [26] data. To the best of our knowledge, this is the first time the benefit of DNN uncertainty training and decoding is demonstrated on top of an fMLLR-domain baseline.

The structure of the paper is as follows. Section 2.3 summarizes conventional uncertainty decoding. Section 3 introduces the proposed uncertainty training algorithm. The experimental setup is described in Section 4. Section 5 details the experimental results followed by conclusions in Section 6.

2. BACKGROUND

In the case of noisy uncertain data, rather than assuming clean features $\mathbf{y}$, the posterior distribution $p(\mathbf{y}|\widehat{\mathbf{y}}, \widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}})$ of the clean features given the enhanced features $\widehat{\mathbf{y}}$ can be estimated [10, 20] and exploited for decoding. This distribution is assumed to be Gaussian with mean $\widehat{\mathbf{y}}$ and diagonal covariance (uncertainty) $\widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}}$. The noisy features are denoted by $\mathbf{z}$.

The flow diagram of DNN uncertainty decoding is shown in Fig. 1. The feature-domain uncertainty is first estimated. Then, it is propagated through the DNN acoustic model with parameters θ to compute acoustic scores. Finally, the scores are incorporated in the decoding algorithm to obtain the word sequence.

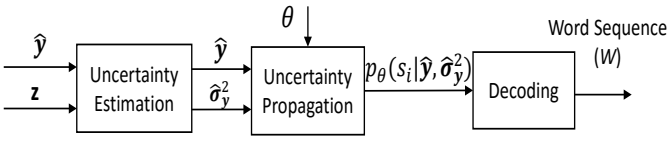

Fig. 1: Flow diagram of DNN uncertainty decoding.

2.1. Uncertainty Estimation

The first step is to estimate the uncertainty $\widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}} = \mathbb{E}(|\mathbf{y} - \widehat{\mathbf{y}}|^2)$. Delcroix [7] proposed a feature-domain estimator that is equal to the (entrywise) squared difference between noisy and enhanced $|\mathbf{z} - \widehat{\mathbf{y}}|^2$, which we refer to as Delcroix's uncertainty (DU) estimator hereafter[1]. DU and other early estimators, e.g., [7, 8, 11, 12, 15] rely on heuristics or approximations which often result in inaccurate estimates [17]. In [10], we tackled this issue by introducing a neural network uncertainty (NNU) estimator trained to predict the oracle uncertainty in the logmel domain given the noisy logmel features $\mathbf{z}$ and the difference between noisy and enhanced features $\mathbf{z} - \widehat{\mathbf{y}}$ as inputs. This estimator was inspired by [17,27], but it has greater learning capacity due to its multilayer architecture and the use of continuous input features.

2.2. DNN Uncertainty Decoding Rule

Given a clean feature vector $\mathbf{y}$, a DNN acoustic model is used to estimate the posterior probability of all hidden Markov model (HMM) states

$$p_\theta(s_i|\mathbf{y}) \tag{1}$$

where s_i denotes the i-th state and θ the set of DNN parameters. In the case of noisy uncertain data, instead of computing the posterior over clean features, the expectation of this quantity over the clean feature distribution must be computed instead [20]:

$$p_\theta(s_i|\widehat{\mathbf{y}}, \widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}}) = \mathbb{E}\left[p_\theta(s_i|\mathbf{y})\middle|\widehat{\mathbf{y}}, \widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}}\right]. \tag{2}$$

Pseudo log-likelihoods are then derived as $\log p_\theta(s_i|\widehat{\mathbf{y}}, \widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}}) - \log p(s_i)$ and used for decoding.

2.3. Uncertainty Propagation

In practice, the expectation (2) can be approximated by numerical sampling using either MC or UT [20].

2.3.1. Monte Carlo Sampling

In MC sampling, samples $\widetilde{\mathbf{y}}^n$ are drawn randomly from the feature uncertainty distribution $p(\mathbf{y}|\widehat{\mathbf{y}}, \widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}})$. Each sample is propagated through the entire DNN. Finally, the DNN outputs are averaged across all samples to approximate the posterior expectation.

2.3.2. Unscented Transform

The UT approach is similar to MC sampling. However, the samples are drawn according to a deterministic procedure and each sample is associated with a weight w^n. The samples are passed through the DNN as in MC. The weighted average of the outputs approximates the posterior expectation.

[1] DU is also sometimes called Kolossa's uncertainty (KU) estimator in the particular case when the features are spectral (e.g., logmel) features.

3. DNN UNCERTAINTY TRAINING

The flow diagram for the proposed DNN uncertainty training procedure is shown in Fig. 2. Once the uncertainty over the training data has been estimated, the input features are sampled using the UT sampling approach. These samples and the associated weights are then used in training the DNN. The procedure is detailed below.

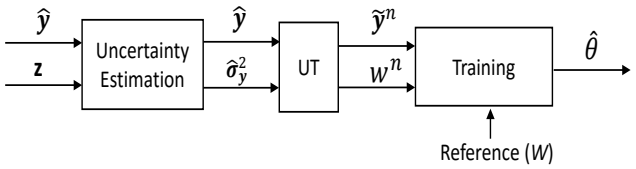

Fig. 2: Flow diagram of DNN uncertainty training.

Given a sequence of clean features $\mathbf{y}_t, t \in \{1, \ldots, T\}$ and the corresponding reference word sequence W, DNN training is classically achieved by solving the following optimization problem:

$$\widehat{\theta} = \arg\min_{\theta} \sum_{t=1}^{T} G_\theta(\mathbf{y}_t, W) \tag{3}$$

where θ denotes the DNN parameters and G_θ a suitable cost function, such as the cross-entropy or any other frame-level discriminative cost. The total number of frames in the training set is denoted by T.

In the case of noisy uncertain data, we follow the same principle as in Section 2.2 and replace the cost function by its expectation over the feature uncertainty distribution:

$$\widehat{\theta} = \arg\min_{\theta} \sum_{t=1}^{T} \mathbb{E}\left[G_\theta(\mathbf{y}_t, W) \middle| \widehat{\mathbf{y}}_t, \widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}_t} \right] \tag{4}$$

with $\widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}_t}$ the uncertainty at time t. Similarly to Section 2.3, we use numerical sampling to compute it. Denoting by $\widetilde{\mathbf{y}}^n_t$, $n \in \{1, \ldots, N\}$, the samples drawn from $p(\mathbf{y}_t|\widehat{\mathbf{y}}_t, \widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}_t})$ and by w^n the associated weights, the problem becomes:

$$\widehat{\theta} \approx \arg\min_{\theta} \sum_{t=1}^{T} \sum_{n=1}^{N} w^n G_\theta(\widetilde{\mathbf{y}}^n_t, W) \tag{5}$$

Here, N is the number of samples drawn for each time frame. The cost function turns out to be similar to the classical cost function in (3) where each clean feature vector has been replaced by N sampled vectors with possibly different weights. The total size of the training set is therefore augmented by a factor of N.

In the following, we use cross-entropy as the objective function. Equation (5) can then be expressed as:

$$\widehat{\theta} \approx \arg\min_{\theta} \sum_{t=1}^{T} \sum_{n=1}^{N} -w^n \log p_\theta(s^*_t|\widetilde{\mathbf{y}}^n_t). \tag{6}$$

The true HMM state s^*_t in the t-th time frame is obtained by Viterbi alignment given W and $p_\theta(s^*_t|\widetilde{\mathbf{y}}^n_t)$ is the DNN output for the n-th sample given the parameters θ.

4. EXPERIMENTAL SETUP

4.1. Datasets

We evaluated the proposed uncertainty training procedure on the CHiME-2 and CHiME-3 datasets.

- The CHiME-2 dataset was formed by convolving clean Wall Street Journal (WSJ0) utterances with binaural room impulse responses (BRIRs). Real domestic background noise was then added at six different signal-to-noise-ratios (SNRs). The training set consists of 7138 simulated noisy utterances spoken by 83 speakers. The development and test sets contain 2460 and 1980 simulated noisy utterances spoken by 10 and 8 speakers, respectively.

- The CHiME-3 dataset consists of both real and simulated recordings of WSJ0 utterances acquired by a tablet equipped with 6 microphones in four noise environments: bus (BUS), café (CAF), pedestrian area (PED), and street (STR). The training set consists of 1600 real and 7138 simulated utterances pronounced by 87 speakers. The development set contains 1640 real and 1640 simulated utterances from 4 speakers. The test set contains 1320 real and 1320 simulated utterances from 4 speakers. In the following, the results are reported only on the real part of the development and test sets.

It may be noted that for both datasets, different speakers and different noises are used in the training, development and test sets.

4.2. Speech Enhancement and ASR Baseline

We enhanced all noisy datasets (training, development, and test) by multichannel nonnegative matrix factorization [28]. We used the FASST toolbox [29] for this purpose. This choice of enhancement method is motivated by its use in [10, 17, 20, 24]. To facilitate comparison, the same algorithm parameters were employed as in these studies.

We trained separate DNN acoustic models for CHiME-2 and CHiME-3. The DNN architecture consists of a 440-dimensional input layer followed by seven 2048-dimensional hidden layers. At the output layer, there are 2000 and 1978 states for CHiME-2 and CHiME-3 respectively. Restricted Boltzmann machine (RBM) pre-training was used to initialize the DNN parameters. The weights were fine-tuned using stochastic gradient descent (SGD). During decoding, we used the enhanced development and test sets with a trigram language model and 5k vocabulary size. No re-scoring (using,

e.g., neural network language models or sequence-level minimum Bayes risk) was performed.

The input features and the training targets were obtained as follows. We first trained a GMM acoustic model using 40-dimensional fMLLR features on the original clean WSJ0 training data. We used the senone level alignments obtained by this model on this data as training targets for the simulated training data of CHiME-2 and CHiME-3. Regarding the real CHiME-3 training data, the alignments were obtained from a GMM model trained on enhanced data instead. These alignments were then used for DNN training on enhanced training data using 40-dimensional fMLLR features with 11-frame splicing. The development sets were used for early stopping. In the case of CHiME-3, we used the full enhanced training and development sets (real and simulated).

The performance of each ASR system is evaluated in terms of word error rate (WER). For the CHIME-2 development and test sets, the confidence interval is about $\pm 0.4\%$. For CHIME-3, the confidence interval is $\pm 0.3\%$ for the development set and $\pm 0.5\%$ for the test set. In all tables, the best result along each column is shown in bold. For comparison with results obtained using logmel features, see [10].

4.3. Proposed Uncertainty Estimator

We propose a new uncertainty estimator, which operates directly in the fMLLR domain. This DNNU estimator relies on a DNN whose inputs are 80-dimensional feature vectors consisting of the noisy features $\mathbf{z}$ concatenated with the difference between noisy and enhanced features $\mathbf{z} - \widehat{\mathbf{y}}$. The outputs are the 40-dimensional uncertainty vectors $\widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}_t}$. The DNN is trained to predict the oracle uncertainty $|\mathbf{y} - \widehat{\mathbf{y}}|^2$ on the training set.

We use 3 hidden layers, each with 500 sigmoid units. The output layer is also passed through a sigmoid activation function and the training targets are scaled across each dimension by the corresponding maximum value over the training set. The same scale factor is used to restore the original scale during the testing phase. The weights are initialized by RBM pre-training and fine-tuned by SGD using mean square error as the objective function.

This DNNU estimator is similar in essence to the NNU estimator in our previous work [10], except that we use 3 hidden layers instead of 2 and the inputs and outputs are in the fMMLR domain instead of the logmel domain.

4.4. Uncertainty Training and Decoding Parameters

In our experiments, we used the proposed DNNU estimator together with UT based sampling for both training and decoding. Following [20], we drew $N = 3$ deterministic samples for each time frame which are given by

$$\widetilde{\mathbf{y}}_t^n = \widehat{\mathbf{y}}_t + \alpha^n \mathbf{u}_t \tag{7}$$

with $\mathbf{u}_t$ the (entrywise) square root of the estimated uncertainty $\widehat{\boldsymbol{\sigma}}^2_{\mathbf{y}_t}$. The values of α^n were chosen as $\alpha^1 = 0$, $\alpha^2 = -\sqrt{3}$, $\alpha^3 = \sqrt{3}$, and the associated weights as $w^1 = 2/3$ and $w^2 = w^3 = 1/6$ [20].

The authors in [24] also drew $N = 3$ samples per time frame according to (7). However, they defined $\mathbf{u}_t$ as the difference between noisy and enhanced features $\mathbf{z} - \widehat{\mathbf{y}}$. This quantity is equal up to the sign to the square root of the DU estimator, hence we call it DU in the following. Also, they chose the coefficients as $\alpha^1 = 0$, $\alpha^2 = 0.1$, $\alpha^3 = 0.2$ with equal weights $w^1 = w^2 = w^3 = 1/3$. We refer to this variant of UT as UT$^+$. This sampling process boils down to linear interpolation between the noisy and enhanced feature vectors. Yet, the fact that the coefficients are non-negative results in a bias compared to symmetric coefficients in conventional UT.

5. RESULTS AND DISCUSSION

We compare the ASR performance obtained by training and testing DNN acoustic models on enhanced data without uncertainty (denoted as "None" in the tables below) with that achievable by DNN uncertainty training and/or decoding. For the latter, we compare DNNU combined with UT on the one hand and DU combined with UT$^+$ on the other hand.

5.1. Overall Results

Table 1 presents the average results obtained on the CHiME-2 (simulated) and CHiME-3 (real) test and development sets. The following observations can be made:

1. DNNU-UT uncertainty training and decoding consistently improves the ASR performance compared to the baseline (no uncertainty training and decoding), except for the CHIME-3 development set where the improvement is not significant. The obtained improvement is equal to 5% relative for the CHIME-2 test set and 8% relative for the CHIME-3 test set, which is statistically significant according to a paired difference test.

2. This improvement can be attributed both to uncertainty training and uncertainty decoding. Uncertainty training

Uncertainty		**CHiME-2**		**CHiME-3**	
Training	**Decoding**	**Test**	**Dev.**	**Test**	**Dev.**
None	None	17.62	24.19	19.72	9.03
	DNNU-UT	17.46	23.15	19.29	8.71
DNNU -UT	None	17.13	23.34	18.75	8.71
	DNNU-UT	**16.77**	**22.50**	**18.13**	**8.63**
DU -UT$^+$	None	17.17	23.44	19.21	9.03
	DU-UT$^+$	17.13	23.24	19.02	9.02

Table 1: WER (%) on the CHiME-2 (simulated) and CHiME-3 (real) test and development sets.

Uncertainty		Test Set						Development Set					
Training	**Decoding**	**-6dB**	**-3dB**	**0dB**	**3dB**	**6dB**	**9dB**	**-6dB**	**-3dB**	**0dB**	**3dB**	**6dB**	**9dB**
None	None	29.28	21.69	17.78	14.06	12.20	10.73	36.43	29.57	25.03	21.10	17.45	15.58
	DNNU-UT	28.03	21.74	17.53	13.93	11.79	10.76	36.22	28.37	24.79	19.45	15.49	14.60
DNNU -UT	None	28.61	21.01	17.51	13.77	11.79	10.1	35.89	29.10	24.13	19.91	16.75	14.30
	DNNU-UT	**28.29**	20.78	**17.05**	**13.33**	11.19	**10.02**	35.39	**27.88**	24.11	**19.15**	**14.81**	**13.67**
DU -UT$^+$	None	29.12	20.85	17.67	14.14	**11.12**	10.14	35.76	28.93	24.16	20.19	16.92	14.72
	DU-UT$^+$	29.17	**20.62**	17.81	14.05	**11.11**	10.02	**35.22**	28.67	**24.09**	20.16	16.79	14.53

Table 2: WER (%) per SNR condition on the CHiME-2 (simulated) test and development sets.

Uncertainty		Test Set				Development Set			
Training	**Decoding**	**BUS**	**CAF**	**PED**	**STR**	**BUS**	**CAF**	**PED**	**STR**
None	None	24.38	17.23	26.93	10.34	11.02	9.44	7.17	8.50
	DNNU-UT	22.60	17.22	27.04	10.30	10.38	9.38	7.03	8.07
DNNU -UT	None	21.39	17.22	25.94	10.45	10.52	9.31	7.03	**7.99**
	DNNU-UT	**19.74**	17.19	**25.36**	**10.23**	**10.25**	**9.27**	**7.01**	**7.99**
DU -UT$^+$	None	22.32	17.25	26.58	10.72	10.84	9.66	7.58	8.07
	DU-UT$^+$	22.21	**17.18**	26.27	10.42	10.75	9.67	7.22	8.44

Table 3: Detailed WER (%) per noise environment on the CHiME-3 (real) test and development sets.

or uncertainty decoding alone also improve the WER, albeit to a lesser extent.

3. DNNU-UT outperforms DU-UT$^+$.

4. To the best of our knowledge, this is the first time the benefit of DNN uncertainty training and decoding is demonstrated on top of an fMLLR-domain baseline.

5.2. Impact of SNR

To analyze the impact of SNR for each configuration, Table 2 presents the WER results with respect to the SNR for the CHiME-2 development and test sets. Our main previous findings still hold:

- For almost all considered SNRs, DNNU-UT uncertainty training improves the WER compared to no uncertainty training, and DNNU-UT uncertainty decoding improves the WER compared to no uncertainty decoding. For some SNRs, these improvements are statistically significant.

- DNNU-UT outperforms DU-UT$^+$ for a majority of SNRs.

Also, the proposed DNNU uncertainty training procedure shows a stable relative WER improvement across SNRs over no uncertainty training. For example, for the test set the improvement varies between 3% at -6 dB and 7% at 9 dB. The improvement is slightly larger for high SNRs.

5.3. Impact of Noise Environment

Table 3 reflects the impact of the noise environment on the WER for the CHiME-3 development and test data. It can be noted that these CHiME-3 results are more difficult to analyze. This is because different noise environments do not only have different noise properties, but also different SNRs. However, some of our previous findings still hold:

- For many noise environments, DNNU-UT uncertainty training and decoding both improve the WER compared to conventional uncertainty training or decoding. The total improvement compared to the baseline can be as large as 19% relative on the BUS test set.

- DNNU-UT outperforms DU-UT$^+$ for most noise environments.

5.4. Comparison with Conventional Data Augmentation

Motivated by the interpretation of the proposed uncertainty training procedure as a form of data augmentation (see Section 2.2), we also compare with a conventional data augmentation (DA) technique, which consists of mixing the speech and noise signals at different SNRs from the original SNR [30]. In our experiment, we used only the CHIME-3 dataset and we increased the simulated training data 3 times by reducing or increasing the SNR by 5 dB compared to the original SNR. The generated data were enhanced using the same enhancement algorithm and settings as the original data. We then used these augmented simulated training data together with the original real training data to train a DNN acoustic

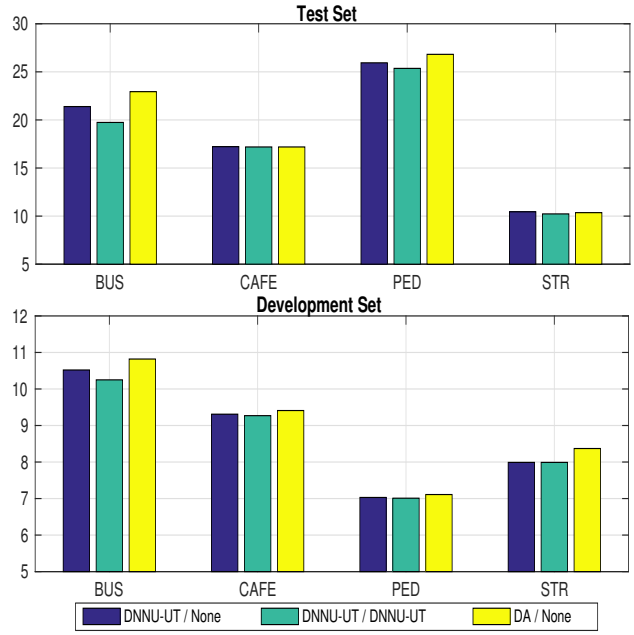

Fig. 3: WER (%) obtained by DNNU-UT training and conventional decoding (blue), DNNU-UT training and decoding (gray), and DA based training and conventional decoding (yellow) for each noise environment in the CHiME-3 (real) test and development sets.

model and we used it for conventional decoding (without uncertainty).

Figure 3 compares the resulting WER performance. It can be seen that DNNU-UT uncertainty training and decoding bring a large improvement compared to DA based training for the BUS and PED environments in the test set and for the BUS and STR environments in the development set, and that they perform comparably in other environments. The total WER improvement is equal to 6% and 3% relative on the test and development sets, respectively. This suggests that DNNU-UT provides an approach for augmenting the training data that consistently improves the WER compared to a conventional, heuristic data augmentation approach.

5.5. Impact of Different α Values on WER

In a final experiment, we compare the performance of the DNNU and DU estimators when combined with different values of α^n and w^n. To do so, we swap the above choices and multiply or divide by 5 to account for the different scale of the two estimators. Specifically, we associate $\alpha^1 = 0$, $\alpha^2 = 0.5$, $\alpha^3 = 1$, and $w^1 = w^2 = w^3 = 1/3$ with DNNU and $\alpha^1 = 0$, $\alpha^2 = -\sqrt{3}/5$, $\alpha^3 = \sqrt{3}/5$, $w^1 = 2/3$, and $w^2 = w^3 = 1/6$ with DU, respectively.

The results obtained with these new weights (not shown here due to space reasons) show that DNNU-UT uncertainty training and decoding still yields 8% relative WER improvement over the baseline on the CHiME-3 real test set, while the improvement brought by DU-UT$^+$ decreases from 5% to 3% relative. This indicates that the choice of α is not critical when the DNNU estimator is used. In other words, the choice of the estimator plays a more important role than the propagation technique. Similar observations are obtained for CHiME-2 dataset.

6. CONCLUSION

In this work, we proposed a new consistent DNN uncertainty training and decoding approach for noisy robust speech recognition. We estimate the variance of the distortions using a DNN uncertainty estimator that operates directly in the fMLLR domain and we then sample the uncertain features using UT during both training and test. We reported experimental results on the CHiME-2 and CHiME-3 datasets using a competitive fMLLR-domain baseline. The proposed method shows better ASR performance than the baseline, the approach in [24], and a conventional data augmentation technique. Also, it appears not to be too sensitive to the choice of the sampling coefficients α^n and the weights w^n.

In future work, we would like to investigate the performance of DNNU uncertainty training on top of conventional data augmentation. Additionally, we would like to compare the performance of oracle uncertainty (OU) estimator with DNNU estimator when used in uncertainty training and decoding.

7. ACKNOWLEDGMENTS

We acknowledge the support of Bpifrance (FUI voiceHome). Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

8. REFERENCES

[1] L. Deng, "Front-end, back-end, and hybrid techniques for noise-robust speech recognition," in *Robust Speech Recognition of Uncertain or Missing Data*, 2011, pp. 67–99.

[2] J. A. Arrowood and M. A. Clements, "Using observation uncertainty in HMM decoding," in *Proc. Interspeech*, 2002, pp. 1561–1564.

[3] N. Becerra Yoma and M. Villar, "Speaker verification in noise using a stochastic version of the weighted Viterbi algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 158–166, 2002.

[4] L. Deng, J. Droppo, and A. Acero, "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 412–421, 2005.

[5] H. Liao and M. Gales, "Joint uncertainty decoding for noise robust speech recognition," in *Proc. Interspeech*, 2005, pp. 3129–3132.

[6] V. Stouten and P. Wambacq, "Model-based feature enhancement with uncertainty decoding for noise robust ASR," *Speech Communication*, vol. 48, no. 11, pp. 1502–1514, 2006.

[7] M. Delcroix, T. Nakatani, and S. Watanabe, "Static and dynamic variance compensation for recognition of reverberant speech with dereverberation preprocessing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 324–334, 2009.

[8] M. Delcroix, S. Watanabe, T. Nakatani, and A. Nakamura, "Cluster-based dynamic variance adaptation for interconnecting speech enhancement pre-processor and speech recognizer," *Computer Speech and Language*, vol. 27, no. 1, pp. 350–368, 2013.

[9] L. Lu, K. Chin, A. Ghoshal, and S. Renals, "Joint uncertainty decoding for noise robust subspace Gaussian mixture models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1791–1804, 2013.

[10] K. Nathwani, J. Morales-Cordovilla, S. Sivasankaran, I. Illina, and E. Vincent, "An extended experimental investigation of DNN uncertainty propagation for noise robust ASR," in *Proc. HSCMA*, 2017, pp. 26–30.

[11] D. Kolossa, R. F. Astudillo, E. Hoffmann, and R. Orglmeister, "Independent component analysis and time-frequency masking for speech recognition in multitalker conditions," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, no. 1, pp. 1–13, 2010.

[12] R. F. Astudillo, "Integration of short-time Fourier domain speech enhancement and observation uncertainty techniques for robust automatic speech recognition," Ph.D. dissertation, TU Berlin, 2010.

[13] R. F. Astudillo and R. Orglmeister, "Computing MMSE estimates and residual uncertainty directly in the feature domain of ASR using STFT domain speech distortion models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1023–1034, 2013.

[14] A. H. Abdelaziz, S. Zeiler, D. Kolossa, V. Leutnant, and R. Haeb-Umbach, "GMM-based significance decoding," in *Proc. ICASSP*, 2013, pp. 6827–6831.

[15] F. Nesta, M. Matassoni, and R. Astudillo, "A flexible spatial blind source extraction framework for robust speech recognition in noisy environments," in *Proc. CHiME*, 2013, pp. 33–40.

[16] D. T. Tran, E. Vincent, and D. Jouvet, "Fusion of multiple uncertainty estimators and propagators for noise robust ASR," in *Proc. ICASSP*, 2014, pp. 5512–5516.

[17] ——, "Nonparametric uncertainty estimation and propagation for noise robust ASR," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1835–1846, 2015.

[18] R. F. Astudillo and J. P. da Silva Neto, "Propagation of uncertainty through multilayer perceptrons for robust automatic speech recognition," in *Proc. Interspeech*, 2011, pp. 461–464.

[19] R. F. Astudillo, A. Abad, and I. Trancoso, "Accounting for the residual uncertainty of multi-layer perceptron based features," in *Proc. ICASSP*, 2014, pp. 6859–6863.

[20] A. H. Abdelaziz, S. Watanabe, J. R. Hershey, E. Vincent, and D. Kolossa, "Uncertainty propagation through deep neural networks," in *Proc. Interspeech*, 2015, pp. 3561–3565.

[21] C. Huemmer, R. Maas, A. Schwarz, R. F. Astudillo, and W. Kellermann, "Uncertainty decoding for DNN-HMM hybrid systems based on numerical sampling," in *Proc. Interspeech*, 2015, pp. 3556–3560.

[22] C. Huemmer, A. Schwarz, R. Maas, H. Barfuss, R. F. Astudillo, and W. Kellermann, "A new uncertainty decoding scheme for DNN-HMM hybrid systems with multichannel speech enhancement," in *Proc. ICASSP*, 2016, pp. 5760–5764.

[23] A. Ozerov, M. Lagrange, and E. Vincent, "Uncertainty-based learning of acoustic models from noisy data," *Computer Speech and Language*, vol. 27, no. 3, pp. 874–894, 2013.

[24] Y. Tachioka and S. Watanabe, "Uncertainty training and decoding methods of deep neural networks based on stochastic representation of enhanced features," in *Proc. Interspeech*, 2015, pp. 3541–3545.

[25] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: An overview of challenge systems and outcomes," in *Proc. ASRU*, 2013, pp. 162–167.

[26] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Analysis and outcomes," *Computer Speech and Language*, to appear.

[27] S. Srinivasan and D. Wang, "Transforming binary uncertainties for robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2130–2140, 2007.

[28] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1118–1133, 2012.

[29] Y. Salaün, E. Vincent, N. Bertin, N. Souviraà-Labastie, X. Jaureguiberry *et al.*, "The flexible audio source separation toolbox version 2.0," in *ICASSP Show & Tell*, 2014.

[30] S. Sivasankaran, E. Vincent, and I. Illina, "Discriminative importance weighting of augmented training data for acoustic model training," in *Proc. ICASSP*, 2017, pp. 4885–4889.