BLHUC: BAYESIAN LEARNING OF HIDDEN UNIT CONTRIBUTIONS FOR DEEP NEURAL NETWORK SPEAKER ADAPTATION

Xurong Xie[1,2], Xunying Liu[1,2], Tan Lee[1], Shoukang Hu[1], Lan Wang[2]

[1]Chinese University of Hong Kong
[2]Shenzhen Institutes of Advanced Technology,
Chinese Academy of Sciences

Introduction

- DNN based speaker adaptation
  - Feature based: i-vector, speaker code, LDA
  - Model based: linear transform, CAT (basis interpolation), LHUC

- Learning hidden unit contributions (LHUC) learns
  - Contributions of DNN hidden outputs using speaker-dependent (SD) scaling vectors
  - Deterministic parameters
  - Limited amount of adaptation data leads to over-fitting and poor generalization

Contributions of the work

- Bayesian Learning of hidden unit contributions (BLHUC)
  - Addressing SD parameter uncertainty in standard LHUC
  - Posterior distribution over the LHUC scaling vector is used
  - Variational inference and sampling based approach for estimating posterior parameters

- Two experiment setups to evaluate BLHUC
  - Unsupervised test time speaker adaptation
  - Speaker adaptive training (SAT)

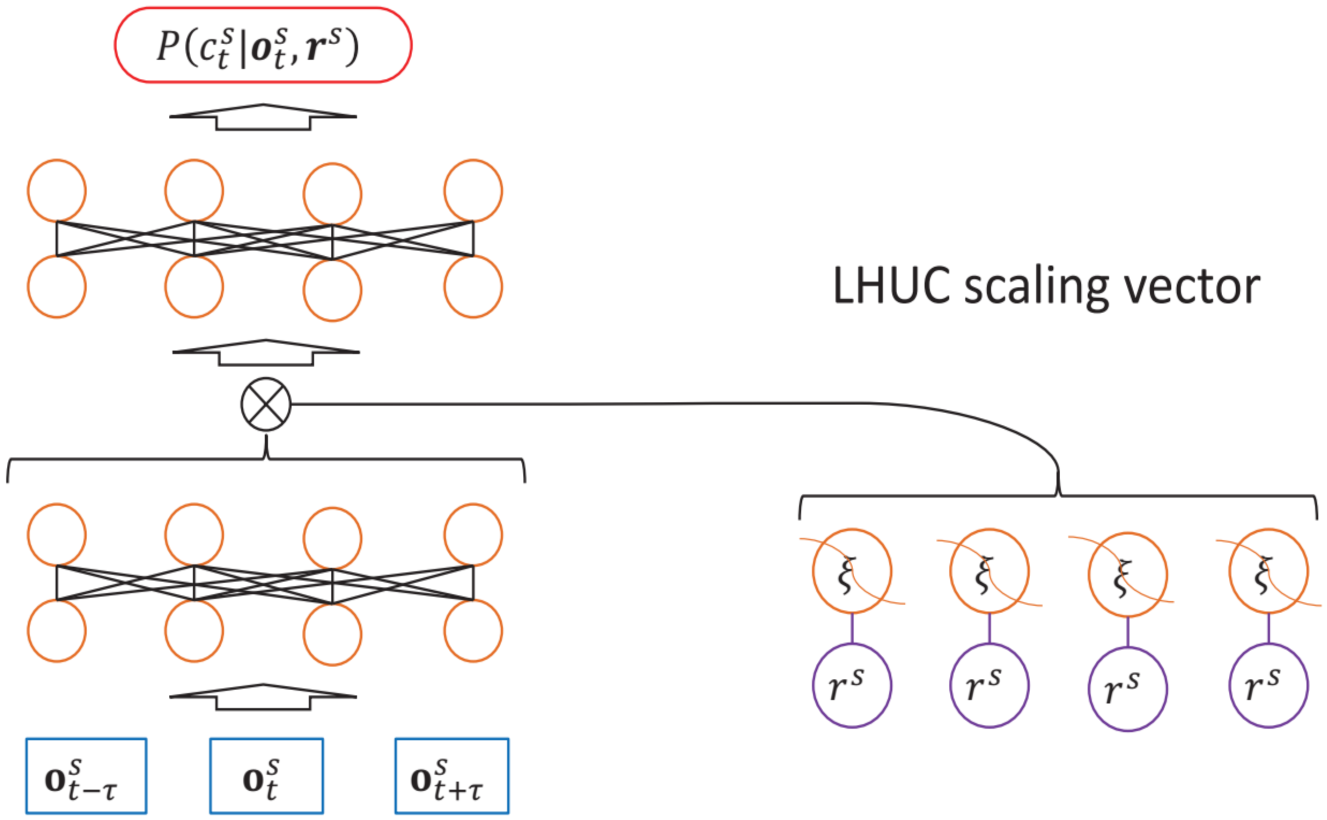
- To the best of our knowledge, this is the first work on using Bayesian learning for DNN speaker adaptation

Learning hidden unit contributions (LHUC)

- Scaling vectors used in element-wise multiplication to modify the DNN hidden node outputs for each speaker

$$\boldsymbol{h}^{l,s} = \xi(\boldsymbol{r}^s) \otimes \psi\left(\boldsymbol{W}^T \boldsymbol{h}^{l-1,s} + \boldsymbol{b}\right)$$

- where $\xi(\boldsymbol{r}^s)$ is the scaling vector parameterized by $\boldsymbol{r}^s$
- $\boldsymbol{r}^s$ encodes speaker information
- $\xi(\cdot) = 2\text{sigmoid}(\cdot)$



Learning hidden unit contributions (LHUC)

- By using LHUC technique, the inference for input feature $\boldsymbol{o}_t^s$ given adaptation data $\boldsymbol{o}^s$ and its alignment c^s is

$$P(c_t^s|\boldsymbol{o}_t^s, \boldsymbol{o}^s, c^s) = \int P(c_t^s|\boldsymbol{o}_t^s, \boldsymbol{r}^s) p(\boldsymbol{r}^s|\boldsymbol{o}^s, c^s) d\boldsymbol{r}^s$$

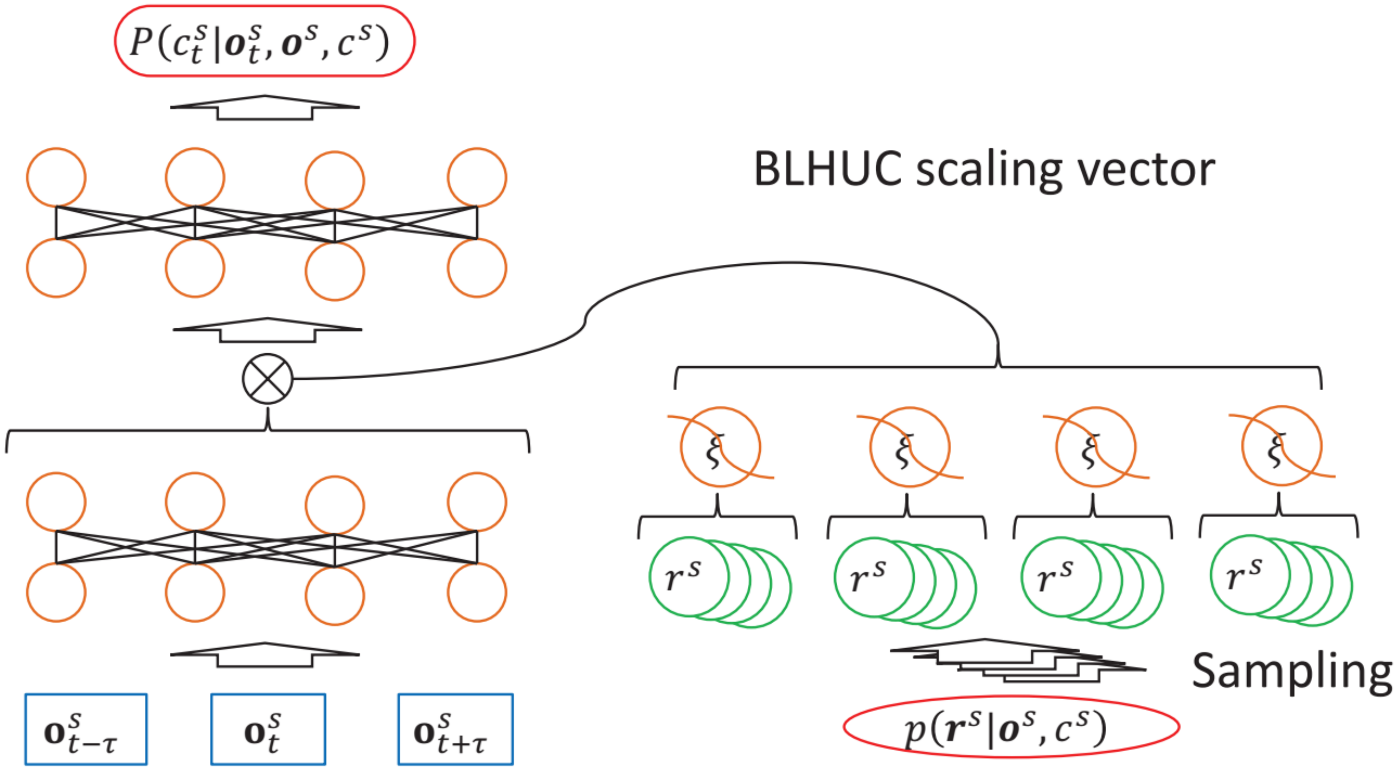
$$\approx P(c_t^s|\boldsymbol{o}_t^s, \hat{\boldsymbol{r}}^s)$$

- $\hat{\boldsymbol{r}}^s = \arg\max_{\boldsymbol{r}^s} P(\boldsymbol{r}^s|\boldsymbol{o}^s, c^s)$ is the deterministic parameter estimate of $\boldsymbol{r}^s$
- Assuming we are very confident that this deterministic estimate is reliable
- $\boldsymbol{r}^s$ is often of high dimension in practice, and adaptation data is limited
- Parameter uncertainty leads to overfitting and poor generalization

Bayesian learning of hidden unit contributions (BLHUC)

- From deterministic to probabilistic estimate of SD parameter $\boldsymbol{r}^s$
- Parameter posterior handles uncertainty

$$P(c_t^s|\boldsymbol{o}_t^s,\boldsymbol{o}^s,c^s)=\int P(c_t^s|\boldsymbol{o}_t^s,\boldsymbol{r}^s)p(\boldsymbol{r}^s|\boldsymbol{o}^s,c^s)d\boldsymbol{r}^s$$



- Parameter posterior to be learnt
- Integral non-trivial to compute
- Back-propagation algorithm not directly usable

- Two tricks:
  - Variational lower bound
  - Parameter sampling

Variational estimation for BLHUC parameters

- The lower bound of cross entropy loss on adaptation data is

$$\begin{aligned}\text{Loss} &= -\log P(c^s|\boldsymbol{o}^s) \\ &= -\log \int P(c^s|\boldsymbol{o}^s, \boldsymbol{r}^s) p(\boldsymbol{r}^s) d\boldsymbol{r}^s \\ &\leq -\boxed{\int q_s(\boldsymbol{r}^s) \log P(c^s|\boldsymbol{o}^s, \boldsymbol{r}^s) d\boldsymbol{r}^s} + KL(q_s||p)\end{aligned}$$

- where $KL(q_s||p) = \int q_s(\boldsymbol{r}^s) \log \frac{q_s(\boldsymbol{r}^s)}{p(\boldsymbol{r}^s)} d\boldsymbol{r}^s$ is the KL divergence

- Variational distribution $q_s(\boldsymbol{r}^s)$ approximates posterior $p(\boldsymbol{r}^s|\boldsymbol{o}^s, c^s)$
- Assumed to be Gaussian – to be learnt

Variational estimation for BLHUC parameters

- Both $q_s(\boldsymbol{r}^s)$ and prior $p(\boldsymbol{r}^s)$ are assumed to be Gaussian for simplification
  - $q_s(r_d^s) = N\left(r_d^s; \mu_{s,d}, \sigma_{s,d}^2\right)$
  - $p(r_d^s) = N\left(r_d^s; \mu_{0,d}, \sigma_{0,d}^2\right)$

- Then, the KL divergence can be exactly calculated by

$$KL(q_s||p) = \frac{1}{2}\sum_{d=1}^{D}\left\{\frac{(\mu_{s,d}-\mu_{0,d})^2+\sigma_{s,d}^2}{\sigma_{0,d}^2} - \log\frac{\sigma_{s,d}^2}{\sigma_{0,d}^2} - 1\right\}$$

- Hyper parameters of both p and q_s are updatable

- But non-trivial to compute $\int q_s(\boldsymbol{r}^s)\log P(c^s|\boldsymbol{o}^s,\boldsymbol{r}^s)d\boldsymbol{r}^s$ – parameter sampling

Variational estimation for BLHUC parameters

- The BLHUC scaling vector posterior can be parameterized by

$$\theta_s^{\mathrm{B}} = \{\boldsymbol{\mu}_s, \boldsymbol{\gamma}_s\}$$

- where $\boldsymbol{\sigma}_s = \exp \boldsymbol{\gamma}_s$

- θ_s^{B} in the integral term of CE is not directly differentiable and updatable
- Re-parameterization used in sampling over θ_s^{B}

$$\boxed{\int q_s(\boldsymbol{r}^s) \log P(c^s|\boldsymbol{o}^s, \boldsymbol{r}^s) d\boldsymbol{r}^s}$$

$$= \int \mathcal{N}(\boldsymbol{\epsilon}; 0, I) \log P(c^s|\boldsymbol{o}^s, \boldsymbol{\mu}_s + \exp(\boldsymbol{\gamma}_s) \otimes \boldsymbol{\epsilon}) d\boldsymbol{\epsilon}$$

$$\approx \frac{1}{J} \sum_{j=1}^{J} \log P(c^s|\boldsymbol{o}^s, \theta_s^{\mathrm{B}}, \boldsymbol{\epsilon}_j)$$

- where $\boldsymbol{\epsilon}_j$ is the jth Monte Carlo sample drawn from $N(0,1)$

Variational estimation for BLHUC parameters

- Then, the gradient of θ_s^{B} in one data batch can be computed by

$$\frac{\partial \mathrm{Loss}_m}{\partial \theta_s^{\mathrm{B}}} \approx \alpha \left\{ -\frac{1}{J} \sum_{j=1}^{J} \frac{\partial \log P\left(c_m^s | \boldsymbol{o}_m^s, \theta_s^{\mathrm{B}}, \boldsymbol{\epsilon}_j\right)}{\partial \theta_s^{\mathrm{B}}} + \frac{N_{m,s}}{N_s} \frac{\partial KL(q_s || p)}{\partial \theta_s^{\mathrm{B}}} \right\}$$

- To be used in back-propagation for estimation of θ_s^{B}
- $\alpha = \frac{N_s}{N_{m,s}}$ can be absorbed by the learning rate
- The coefficient $\frac{N_{m,s}}{N_s}$ adjusts the weight of KL regularization term

Variational estimation for BLHUC parameters

- We set the sampling number by $J = 1$ during adaptation for efficiency
- Then, the resulting gradient is closely related to DNN adaptation using KL-divergence regularization (Yu, Yao, Su, Li & Seide 2013, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition")
- But with additional parameter uncertainty modeled in first term of variational lower bound

$$\frac{\partial \text{Loss}_m}{\partial \theta_s^{\text{B}}} \approx \alpha \left\{ -\frac{1}{J} \sum_{j=1}^{J} \frac{\partial \log P\left(c_m^s | \boldsymbol{o}_m^s, \theta_s^{\text{B}}, \boldsymbol{\epsilon}_j\right)}{\partial \theta_s^{\text{B}}} + \frac{N_{m,s}}{N_s} \frac{\partial KL(q_s || p)}{\partial \theta_s^{\text{B}}} \right\}$$

* The gradient form of standard LHUC using $\boldsymbol{\epsilon}_j$

Inference for BLHUC in decoding

- Inference can be directly approximated by Monte Carlo sampling in the test stage

$$p(c_t^s|\boldsymbol{o}_t^s, \boldsymbol{o}^s, c^s) = \int P(c_t^s|\boldsymbol{o}_t^s, \boldsymbol{r}^s) p(\boldsymbol{r}^s|\boldsymbol{o}^s, c^s) d\boldsymbol{r}^s \approx \frac{1}{J}\sum_{j=1}^{J} P\left(c_t^s|\boldsymbol{o}_t^s, \boldsymbol{r}_j^s\right)$$

- where $\boldsymbol{r}_j^s \sim p(\boldsymbol{r}^s|\boldsymbol{o}^s, c^s) \approx q_s(\boldsymbol{r}^s)$

- A more efficient approximation (used in the paper) is using the mean of the posterior (Normal distribution)

$$\int P(c_t^s|\boldsymbol{o}_t^s, \boldsymbol{r}^s) p(\boldsymbol{r}^s|\boldsymbol{o}^s, c^s) d\boldsymbol{r}^s \approx P(c_t^s|\boldsymbol{o}_t^s, \mathbb{E}[\boldsymbol{r}^s|\boldsymbol{o}^s, c^s]) = P(c_t^s|\boldsymbol{o}_t^s, \boldsymbol{\mu}_s)$$

Different adaptation setups

- Test time adaptation only
  - Standard LHUC estimation
    - Deterministic estimation on adaptation data (Swietojanski & Renals 2016 “Learning hidden unit contributions for unsupervised acoustic model adaptation”)
  - BLHUC estimation
    - SI prior can be separately estimated by training data
    - SI prior can also be zero mean and unit variance for convenience (used in the paper)

- Speaker adaptive training (SAT)
  - Standard LHUC training + standard LHUC test time adaptation
  - Standard LHUC training + BLHUC test time adaptation
    - SI prior mean and variance are computed over training speakers’ LHUC vectors
  - BLHUC training + BLHUC test time adaptation
    - SI prior is updated during training

Experiment setup

- 300 hrs SWBD setup
- Hub5’ 00 for test (SWBD test set + CallHome test set)

- HMMs: 8929 states
- DNN setting
  - Input: 9 successive frames
  - Hidden layer: 2000 nodes, 6 layers, sigmoid
  - Output: 8929 nodes, softmax

- LM: 4-gram, 30,000 words, Fisher + SWBD training

- Features: 80 dimensional f-bank + delta

- Implemented on modified version of Kaldi toolkit and HTK

Result of test time adaptation

- Using all test data as adaptation data
- The BLHUC adapted systems significantly outperformed both the SI baseline system and standard LHUC adapted CE and MPE systems

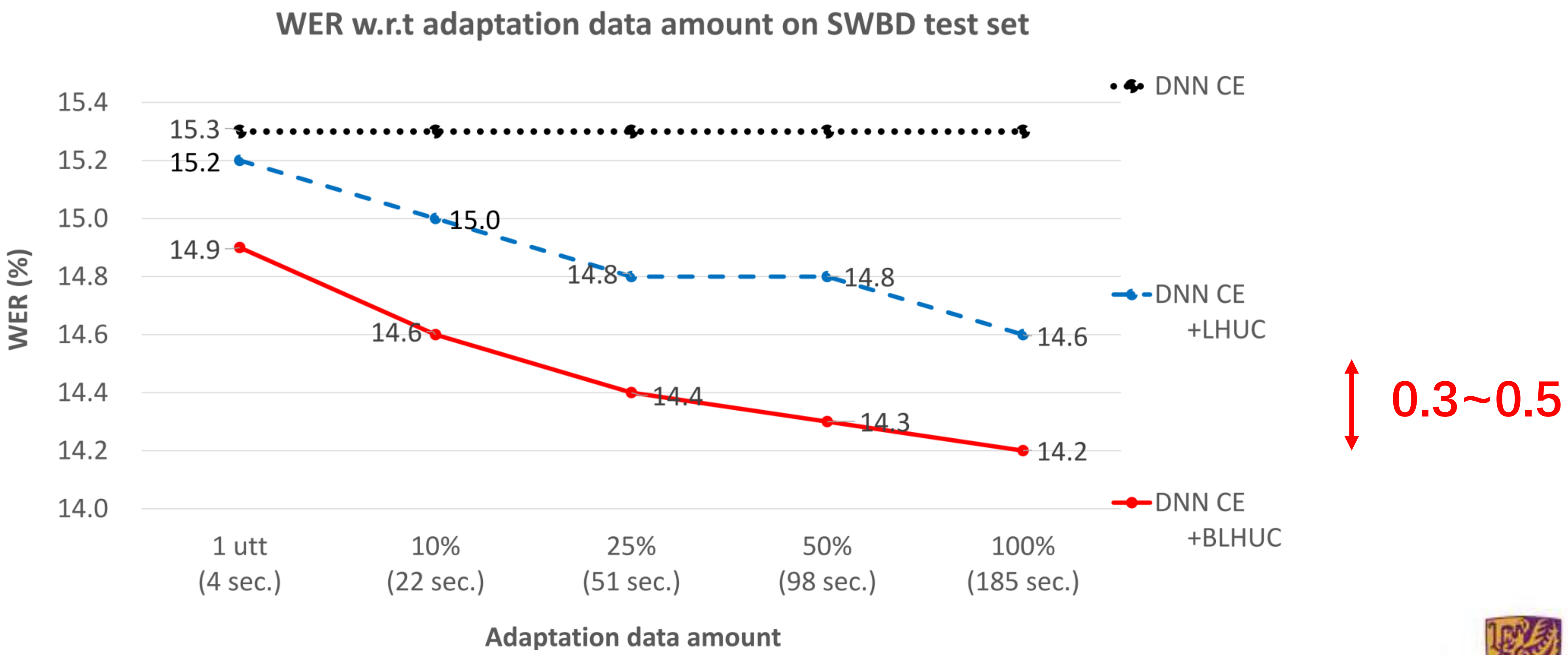
DNN criterion	**Test adapt**	**WER (%)**	
		SWBD	CallHome
CE	-	15.3	27.6
	LHUC	14.6	25.8
	BLHUC	**14.2**	**25.3**
MPE	-	13.4	26.8
	LHUC	12.8	24.0
	BLHUC	**12.4**	**23.1**

↓ 0.5

↓ 0.9

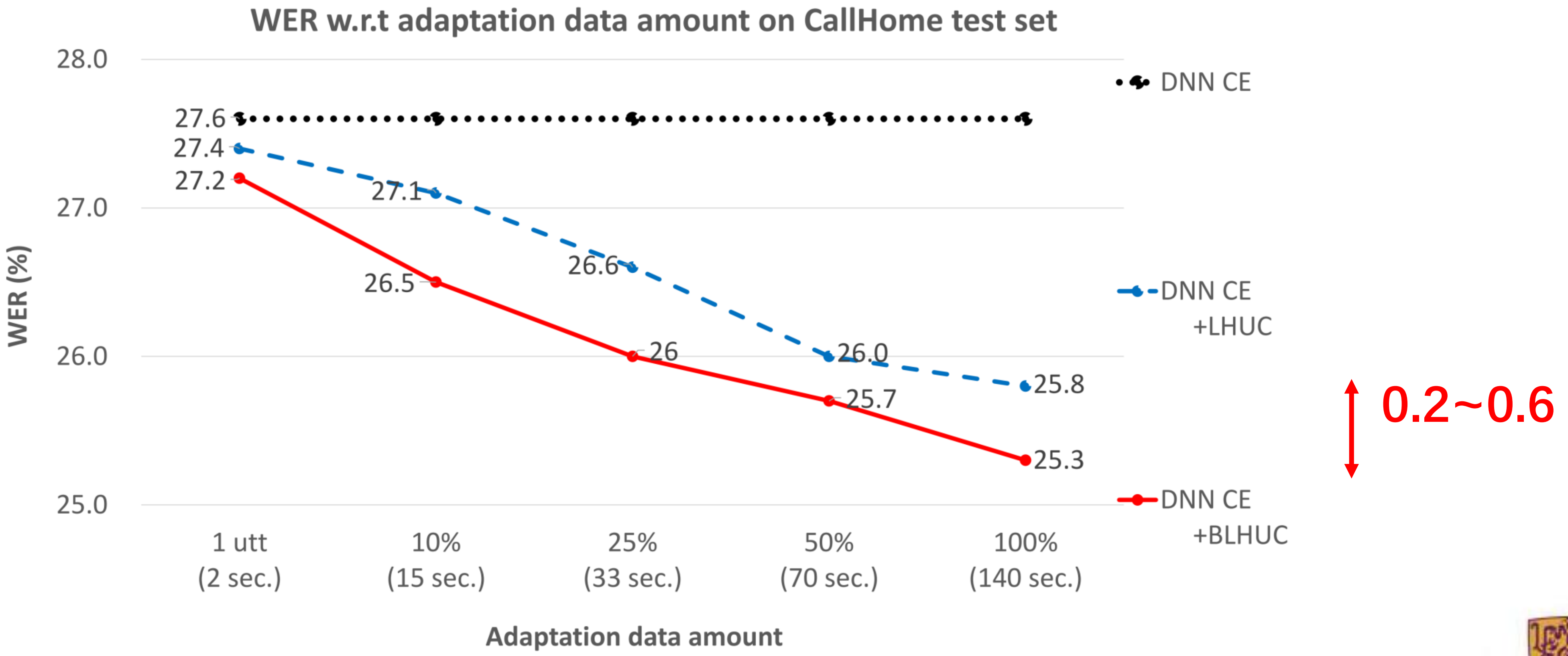
Using different amount of adaptation data for CE systems

- On SWBD test set
- BLHUC adapted systems consistently achieved the best performance using different adaptation data



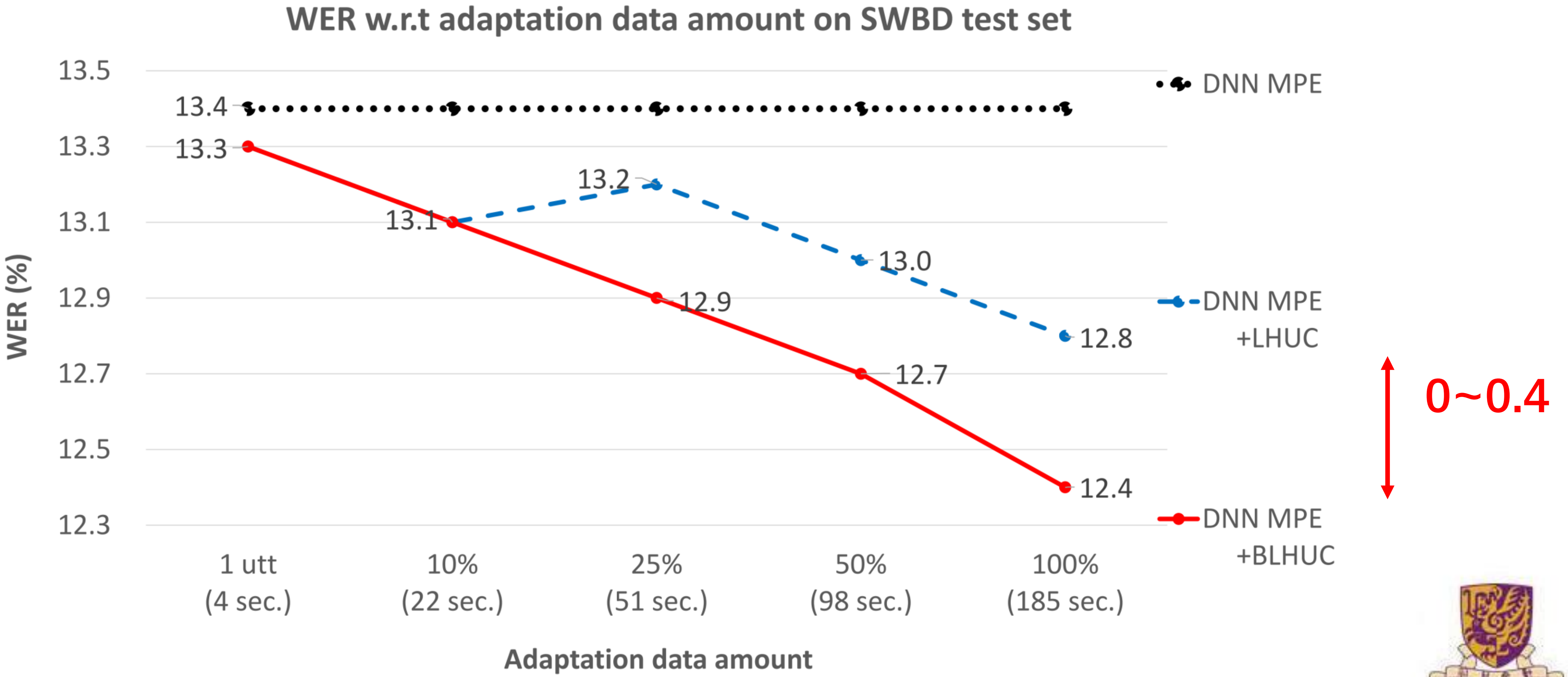
Using different amount of adaptation data for CE systems

- on CallHome test set
- BLHUC adapted systems consistently achieved the best performance using different adaptation data



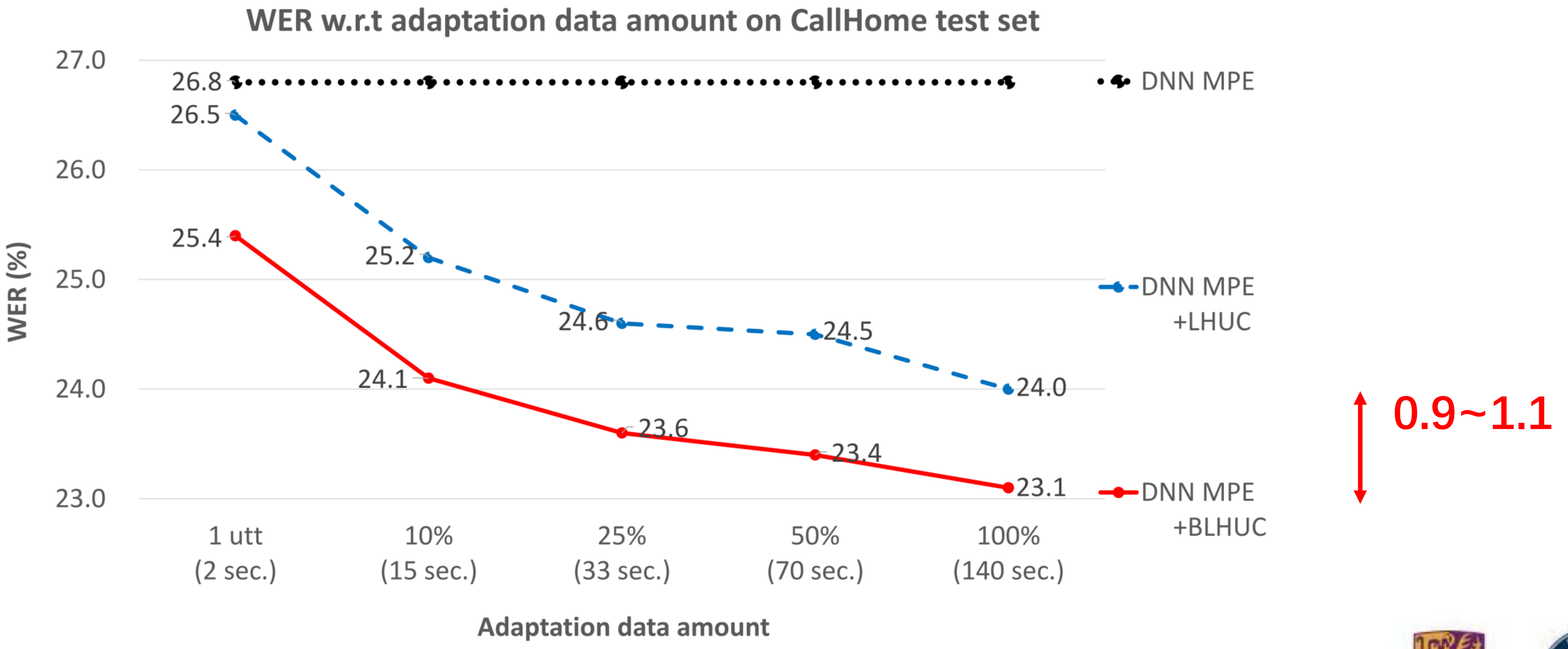
Using different amount of adaptation data for MPE systems

- On SWBD test set
- BLHUC adapted systems consistently achieved the best performance using different adaptation data



Using different amount of adaptation data for MPE systems

- On the harder CallHome test set, BLHUC adaptation obtained significantly improvement by even using only one utterance (2 seconds on average)



Result of SAT

- Using all test data as adaptation data for CE systems
- Using BLHUC for both training and testing achieved the best performance

DNN criterion	SAT	Test adapt	WER (%)	
			SWBD	CallHome
CE	-	-	15.3	27.6
	LHUC	LHUC	13.2	23.5
	LHUC	BLHUC	13.0	23.4
	BLHUC	BLHUC	**12.8**	**22.9**

0.6

Conclusion

- Bayesian learning of the hidden unit contribution for DNN based speaker adaptation is proposed in the work

- An efficient variational approximation for learning LHUC parameter posterior

- BLHUC adaptation consistently outperformed the standard LHUC adaptation, especially on the harder CallHome data set and using limited amount of adaptation data (as minimum as 2 sec of speech)

- To the best of our knowledge, this is the first work on using Bayesian learning for DNN speaker adaptation

- Future work: Bayesian learning of other adaptation techniques