Deep Learning Acoustic Model in Microsoft Cortana Voice Assistant

Jinyu Li

AI & Research, Microsoft

Speech Recognition Products

Selected Technologies behind Microsoft Cortana

- Reduce **runtime** cost without accuracy loss
- Adapt to speakers with **low** footprints
- Time-frequency **invariance** modeling
- Enable languages with **limited** training data
- Reduce accuracy **gap** between large and small deep networks
- **New** domain adaptation
- Multi-talker **separation**

Reduce Runtime Cost without Accuracy Loss

[Xue13, Miao16]

Motivation

- The runtime cost of DNN is much larger than that of GMM, which has been fully optimized in product deployment. We need to reduce the runtime cost of DNN in order to ship it.

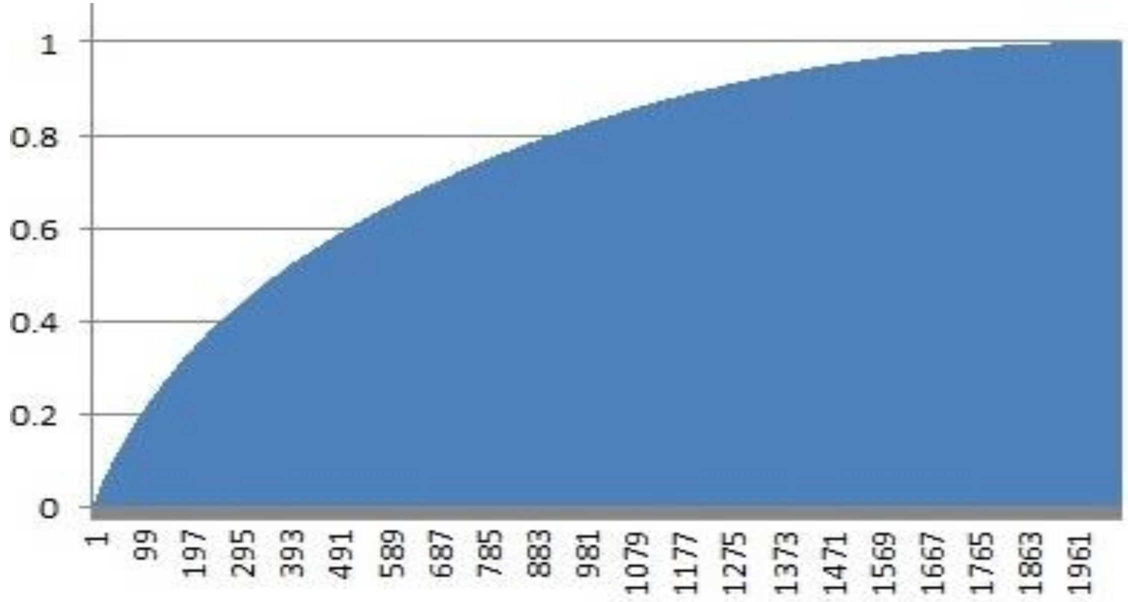
Solution

- The runtime cost of DNN is much larger than that of GMM, which has been fully optimized in product deployment. We need to reduce the runtime cost of DNN in order to ship it.

- We proposed SVD-based model restructuring to compress the DNN models without accuracy loss.

Singular Value Decomposition (SVD)

$$A_{m\times n} = U_{m\times n}\sum\nolimits_{n\times n}V_{n\times n}^{T}=\begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mn} \end{bmatrix}\cdot\begin{bmatrix} \epsilon_{11} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \epsilon_{kk} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \epsilon_{nn} \end{bmatrix}\cdot\begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{bmatrix}$$



SVD Approximation

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mn} \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{11} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \epsilon_{kk} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \epsilon_{nn} \end{bmatrix} \cdot \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{bmatrix}$$

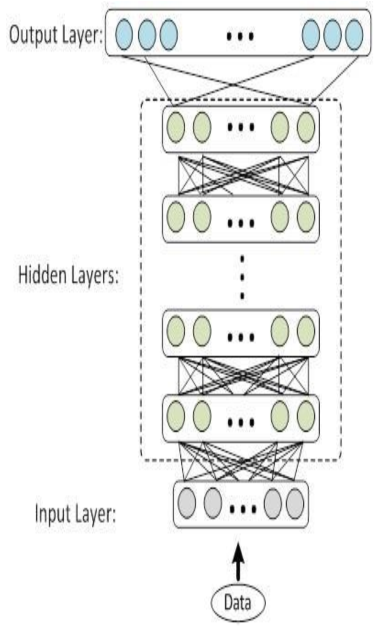
$$\approx \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mn} \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{11} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \epsilon_{kk} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mk} \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \epsilon_{kk} \end{bmatrix} \cdot \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{k1} & \cdots & v_{kn} \end{bmatrix}$$

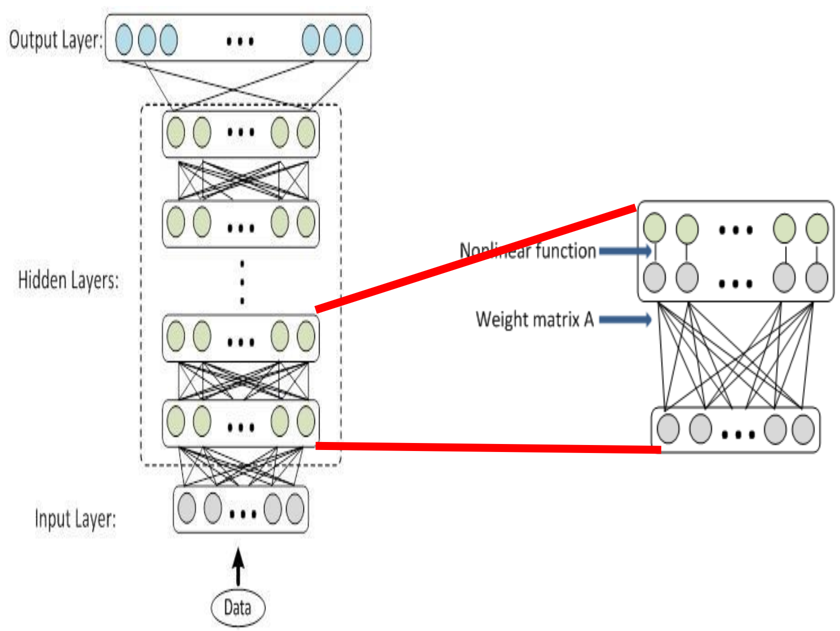
$$= \begin{bmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mk} \end{bmatrix} \cdot \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{k1} & \cdots & w_{kn} \end{bmatrix}$$

- Number of parameters: mn->mk+nk.
- Runtime cost: O(mn) -> O(mk+nk).
- E.g., m=2048, n=2048, k=192.   80% runtime cost reduction without accuracy loss.

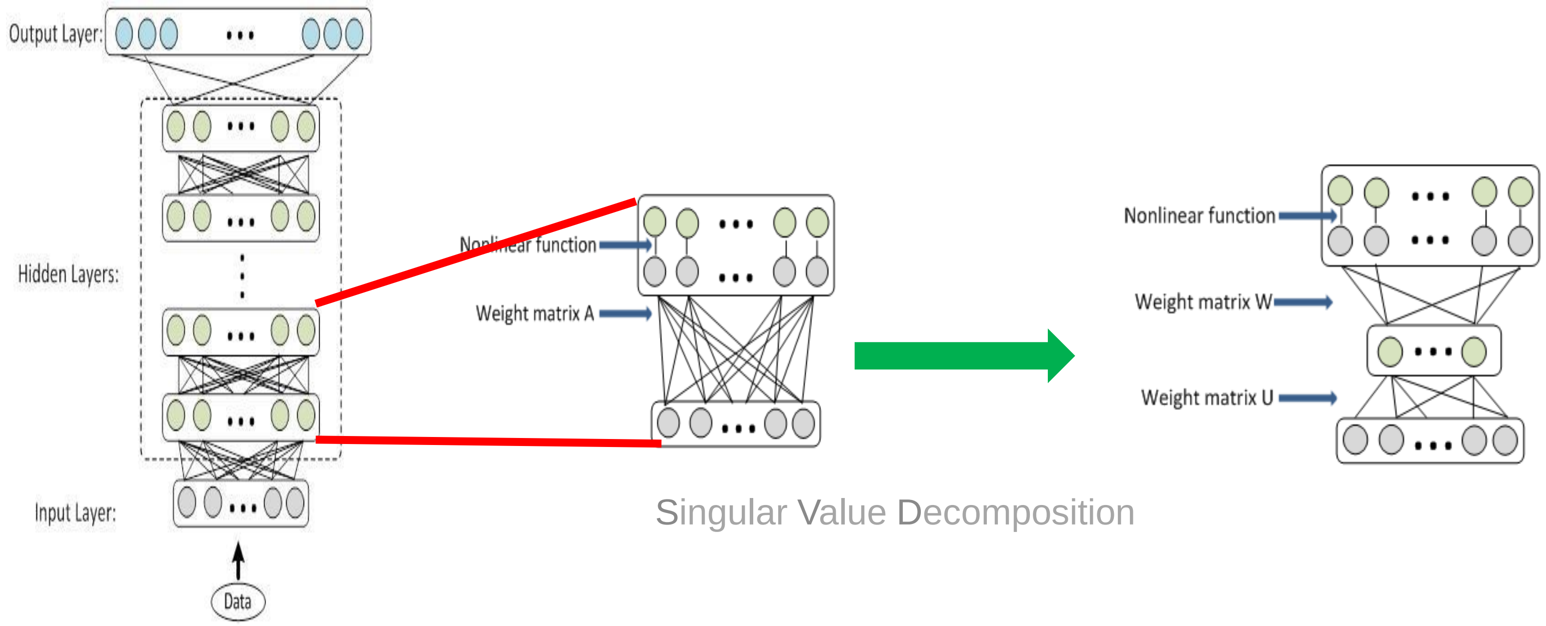
SVD-Based Model Restructuring



SVD-Based Model Restructuring

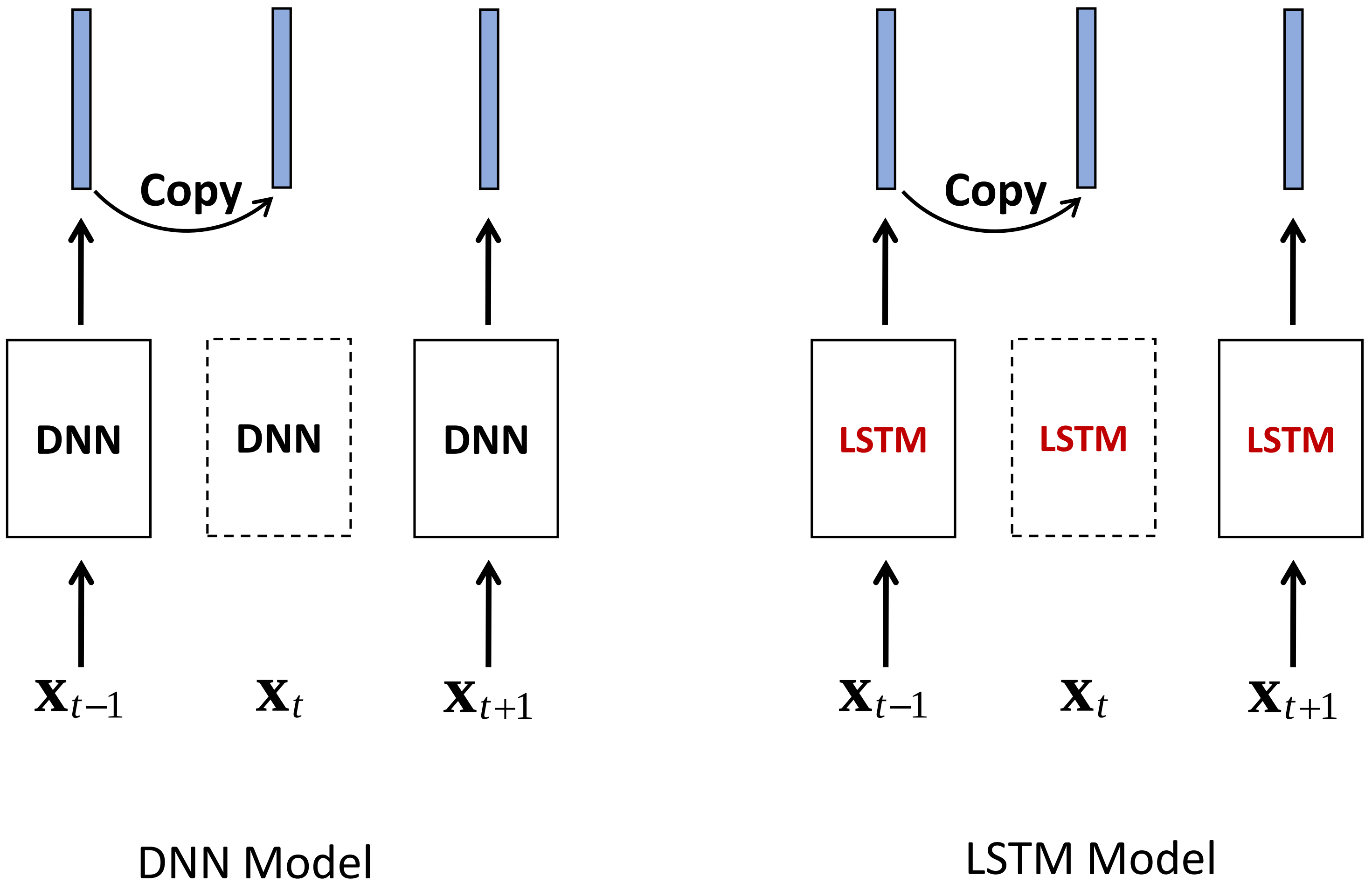


SVD-Based Model Restructuring



Directly training from the low-rank structure without doing SVD costs 4% relative WER increase.

Decoding with Frame Skipping



LSTM Training with Frame Skipping

Split training utterances through frame skipping

- When skipping 1 frame, odd and even frames are picked as separate utterances

$$\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_3 \quad \mathbf{x}_4 \quad \mathbf{x}_5 \quad \mathbf{x}_6$$

$$\mathbf{x}_1 \quad \mathbf{x}_3 \quad \mathbf{x}_5 \qquad\qquad \mathbf{x}_2 \quad \mathbf{x}_4 \quad \mathbf{x}_6$$

- Frame labels are selected accordingly

Adapt to Speakers with Low Footprints

[Xue14]

Motivation

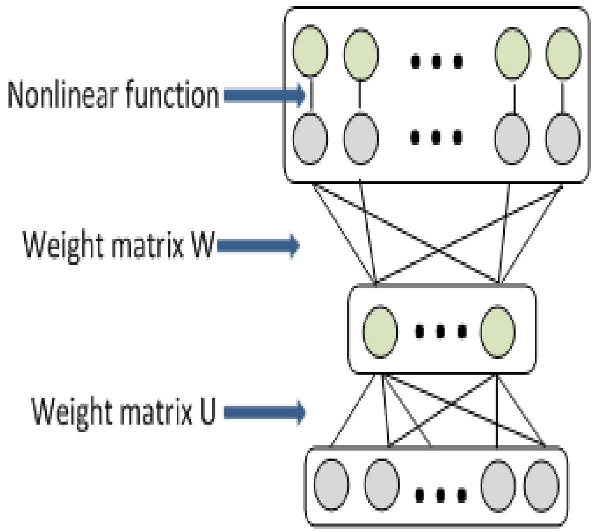
- Speaker personalization with a deep model creates a storage size issue: It is not practical to store an entire deep models for each individual speaker during deployment.

Solution

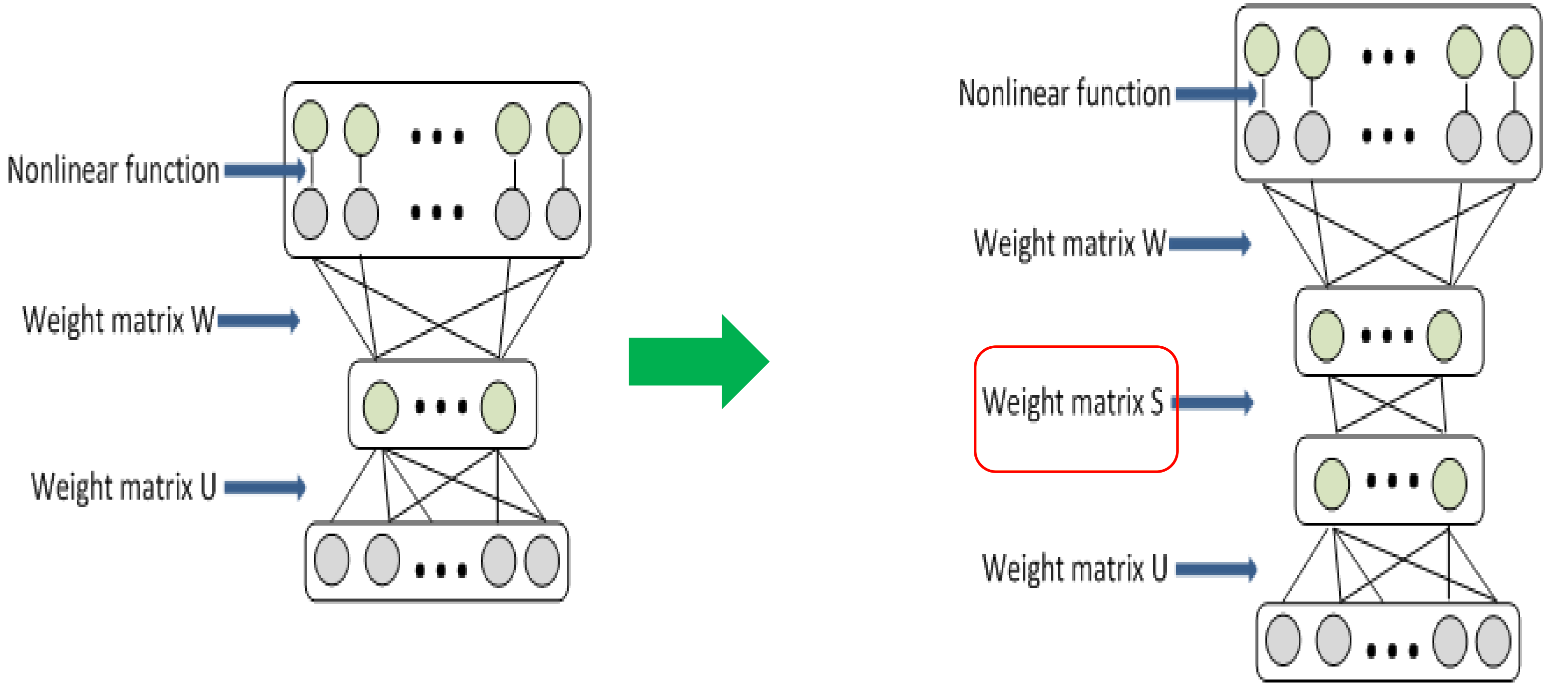
- Speaker personalization with a DNN model creates a storage size issue: It is not practical to store an entire DNN model for each individual speaker during deployment.

- We proposed low-footprint DNN personalization method based on SVD structure.

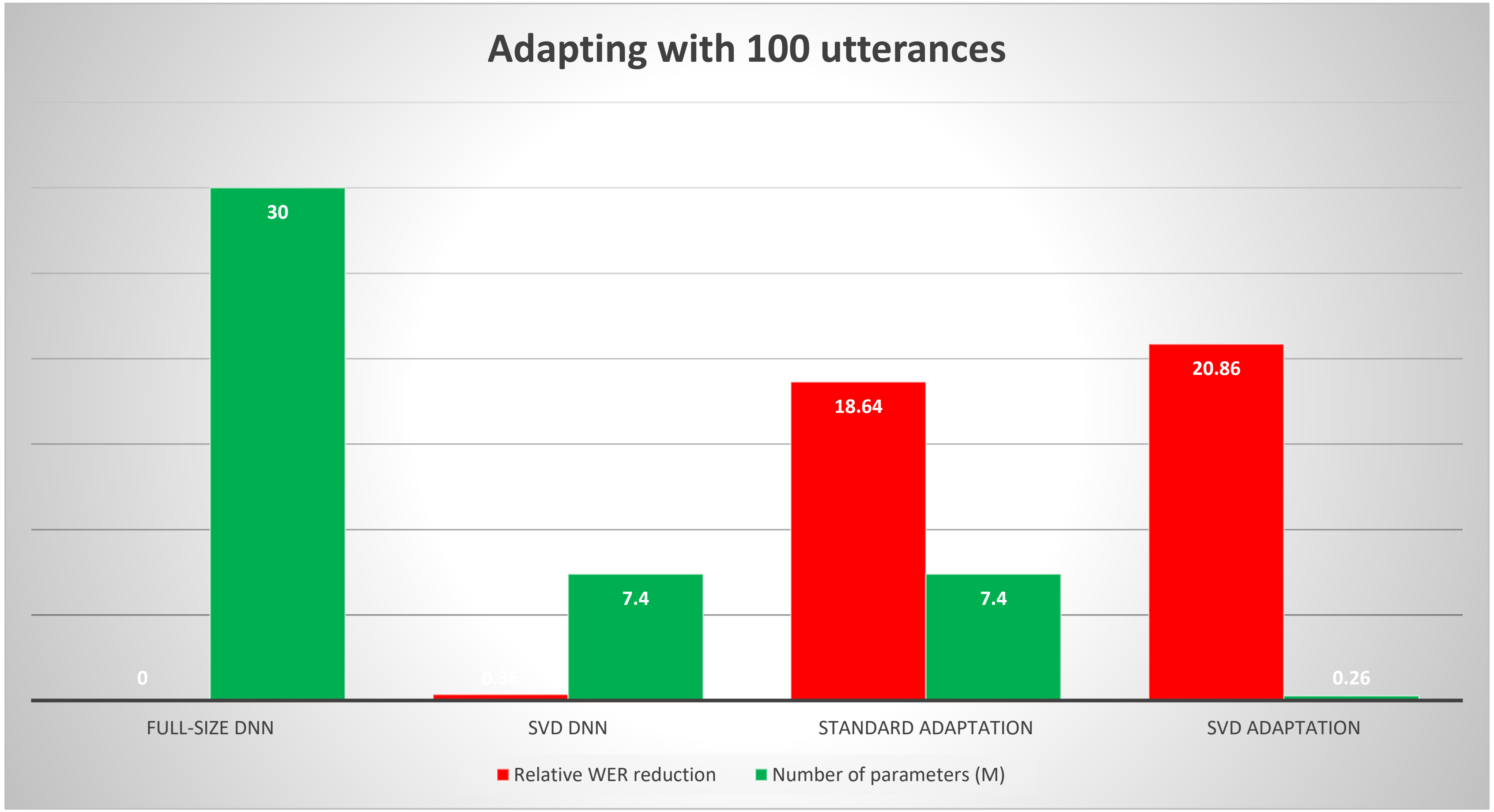
SVD Personalization



SVD Personalization



Adaptation with 100 Utterances

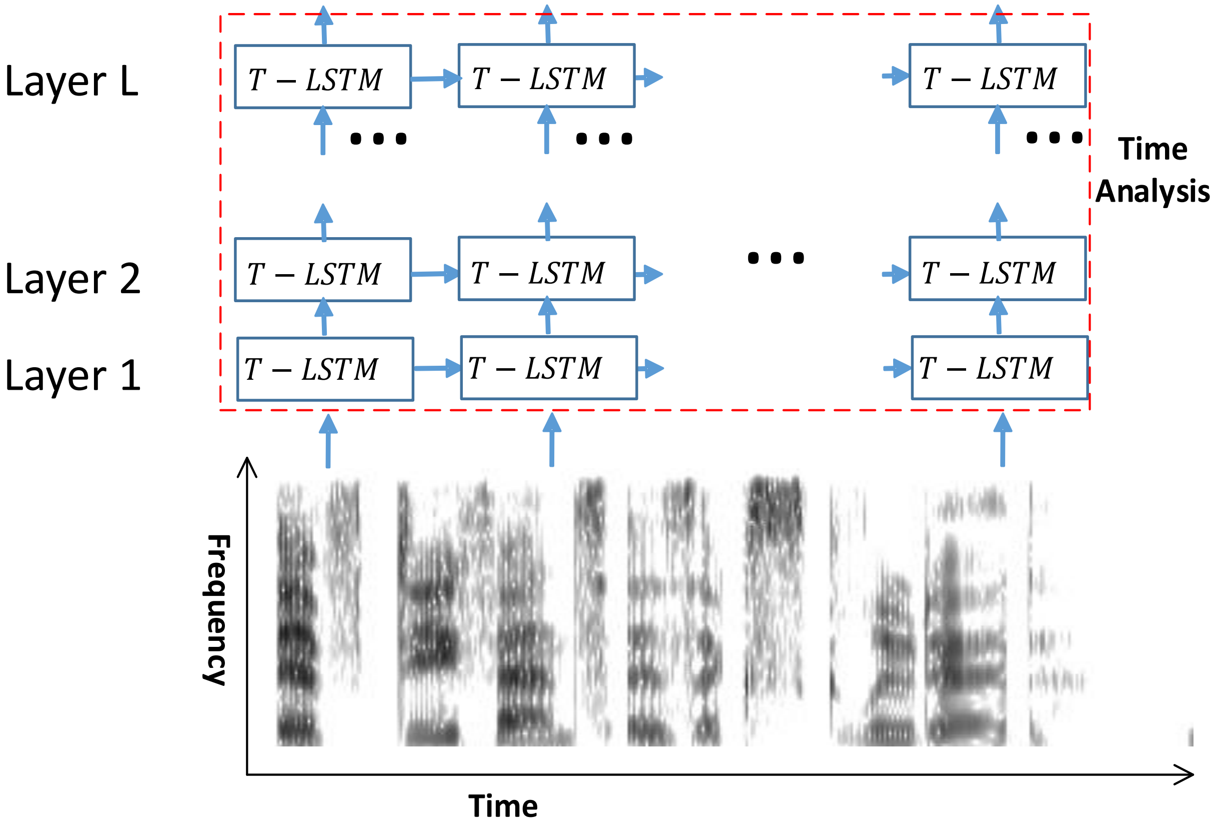


Time-Frequency Invariance Modeling

[Li15, Li16]

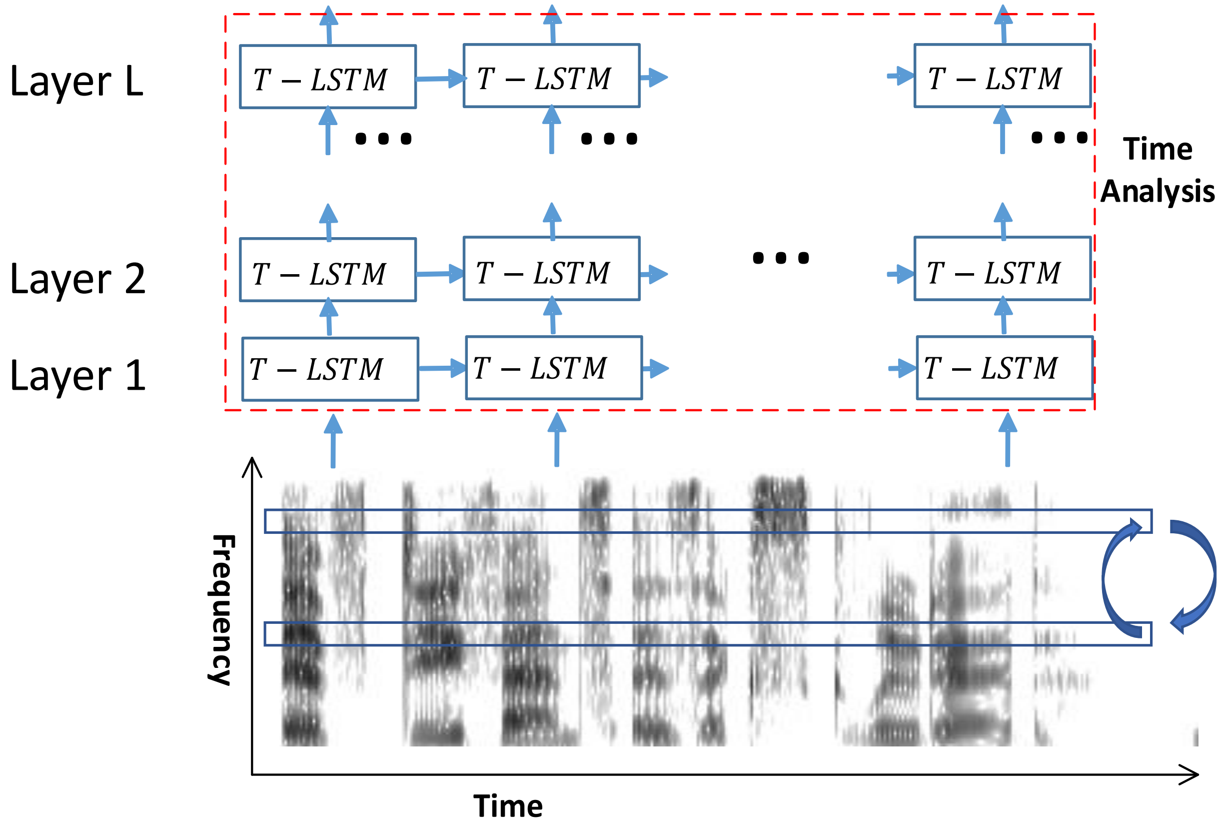
How DNN and (LSTM-)RNN Process an Utterance

- Independence between LFBs



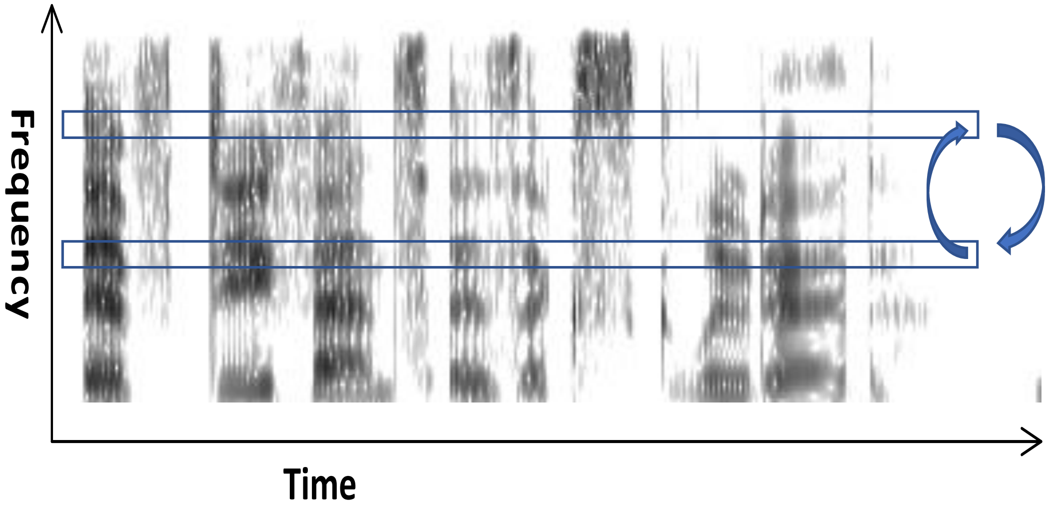
How DNN and (LSTM-)RNN Process an Utterance

- No impact when two LFBs are switched.

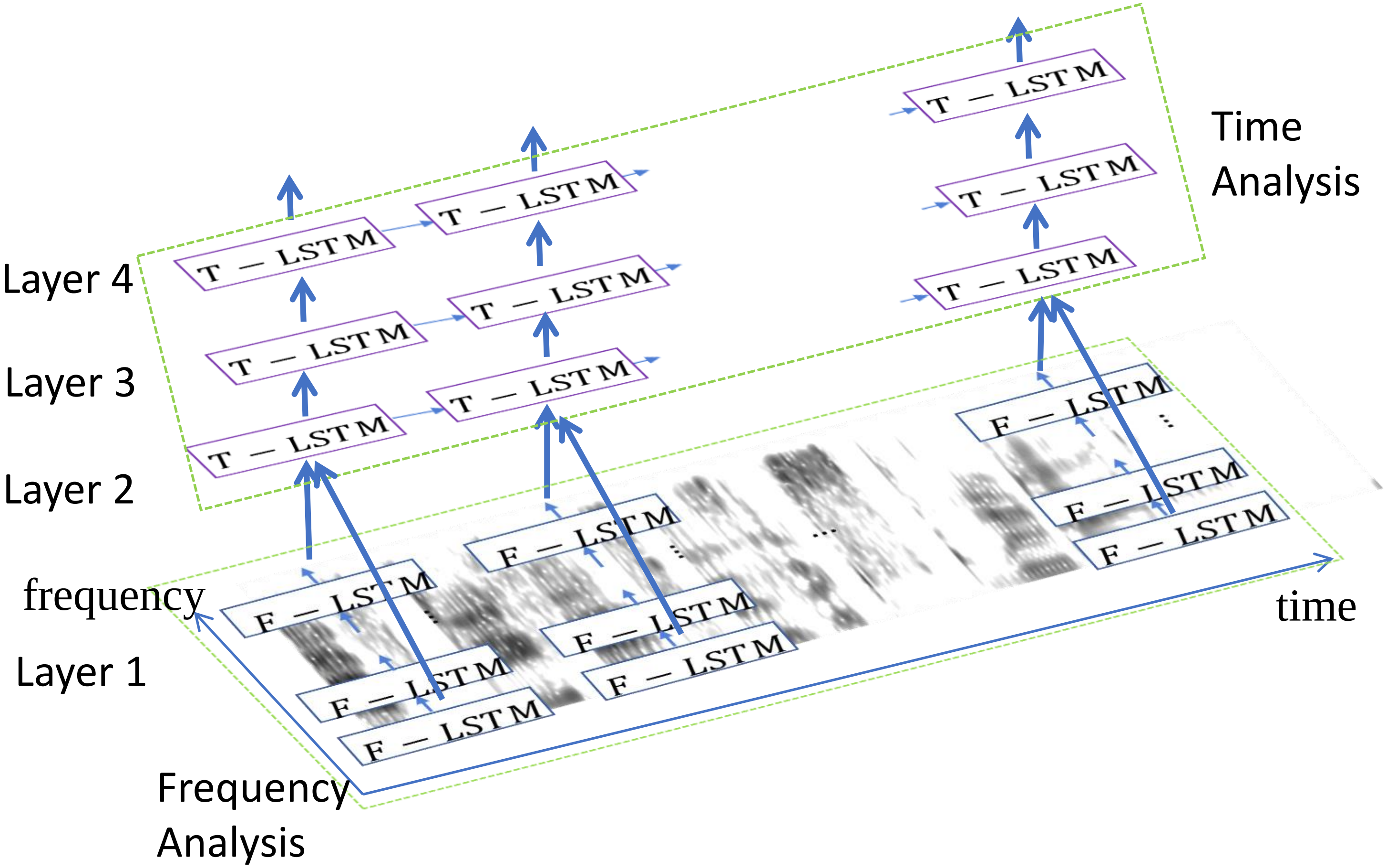


Human Read Spectrum by Using the Correlation across Time and Frequency

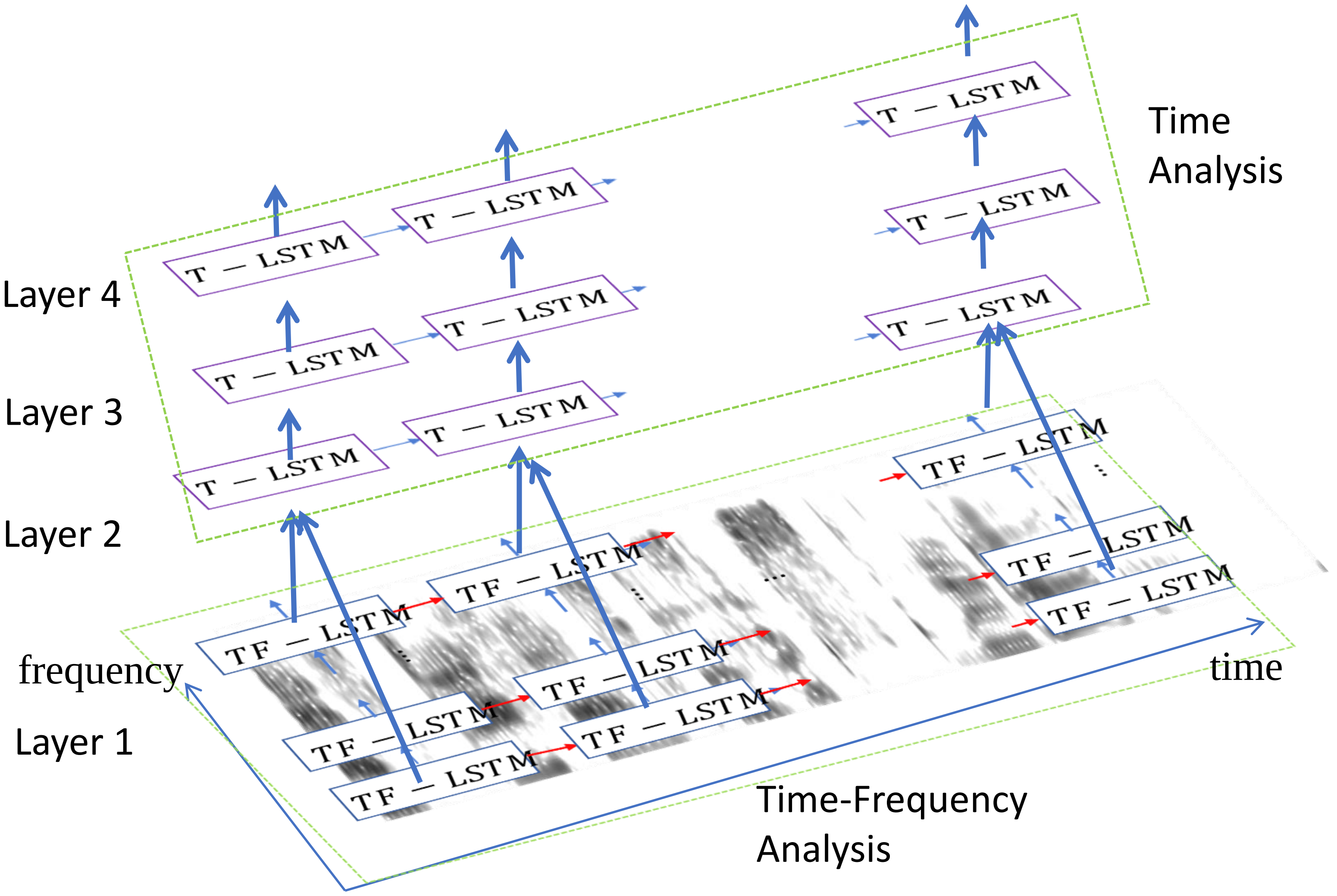
- Big impact when two LFBs are switched.



Frequency-LSTM



Time-Frequency-LSTM



TF-LSTM Results

Models: trained from the 375hr Cortana task

Test set: Cortana

Model	WER (%)	Number of parameters
4-layer T-LSTM	15.35	19.8 M
TF-LSTM + 3-layer T-LSTM	15.09	17.0 M
TF-LSTM + 4-layer T-LSTM	14.83	21.6 M

Invariance Properties

Models: trained from the 375hr Cortana task

Test set: Aurora 4

Model	A	B	C	D	Avg.
4-layer T-LSTM	6.37	14.25	9.14	23.90	17.46
TF-LSTM + 4-layer T-LSTM	5.45	12.07	8.07	20.69	15.01

14.2% WERR

Enable Languages with Limited Training Data

[Huang13]

Motivation

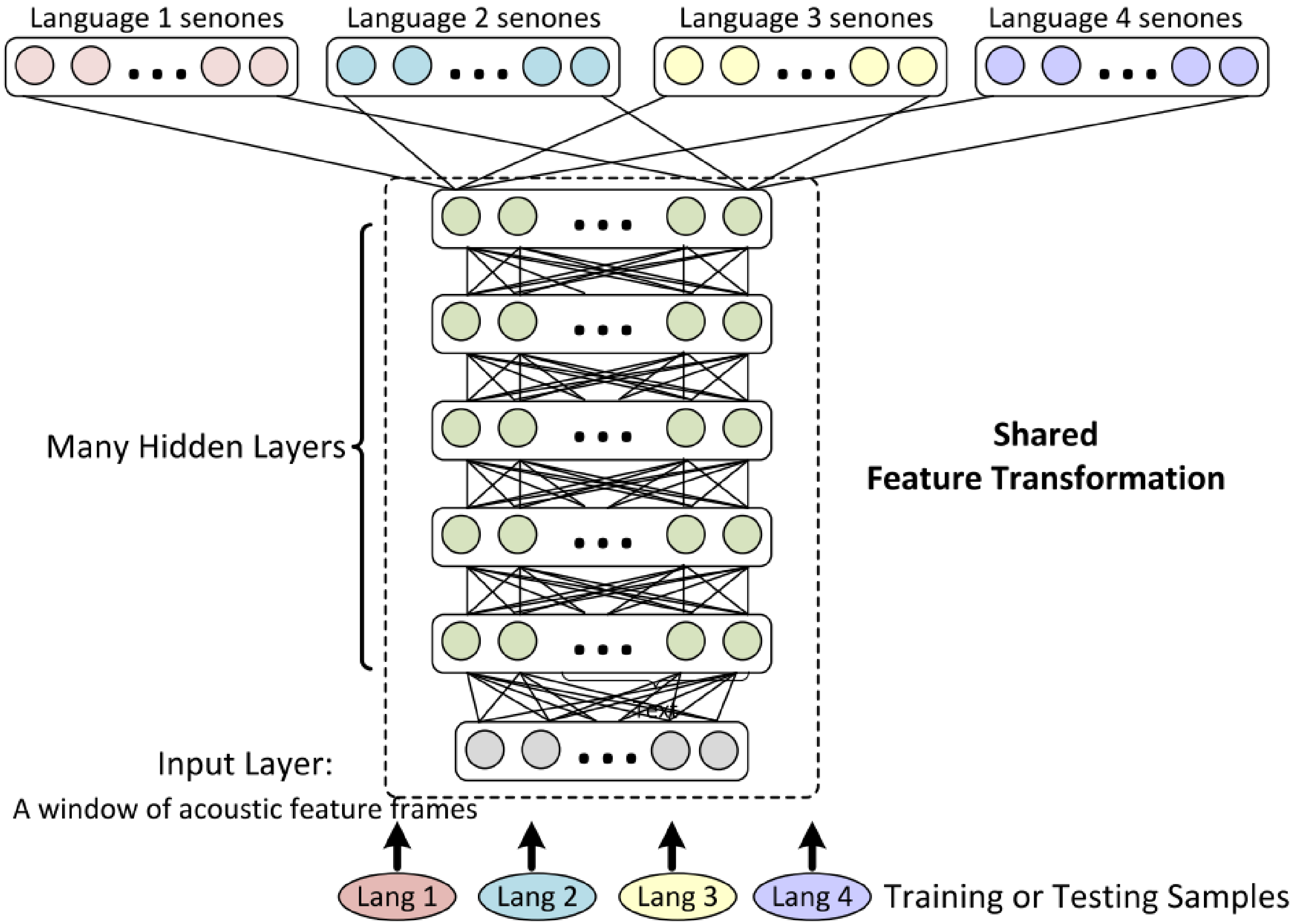
- Develop a new language in new scenario with small amount of training data.

Solution

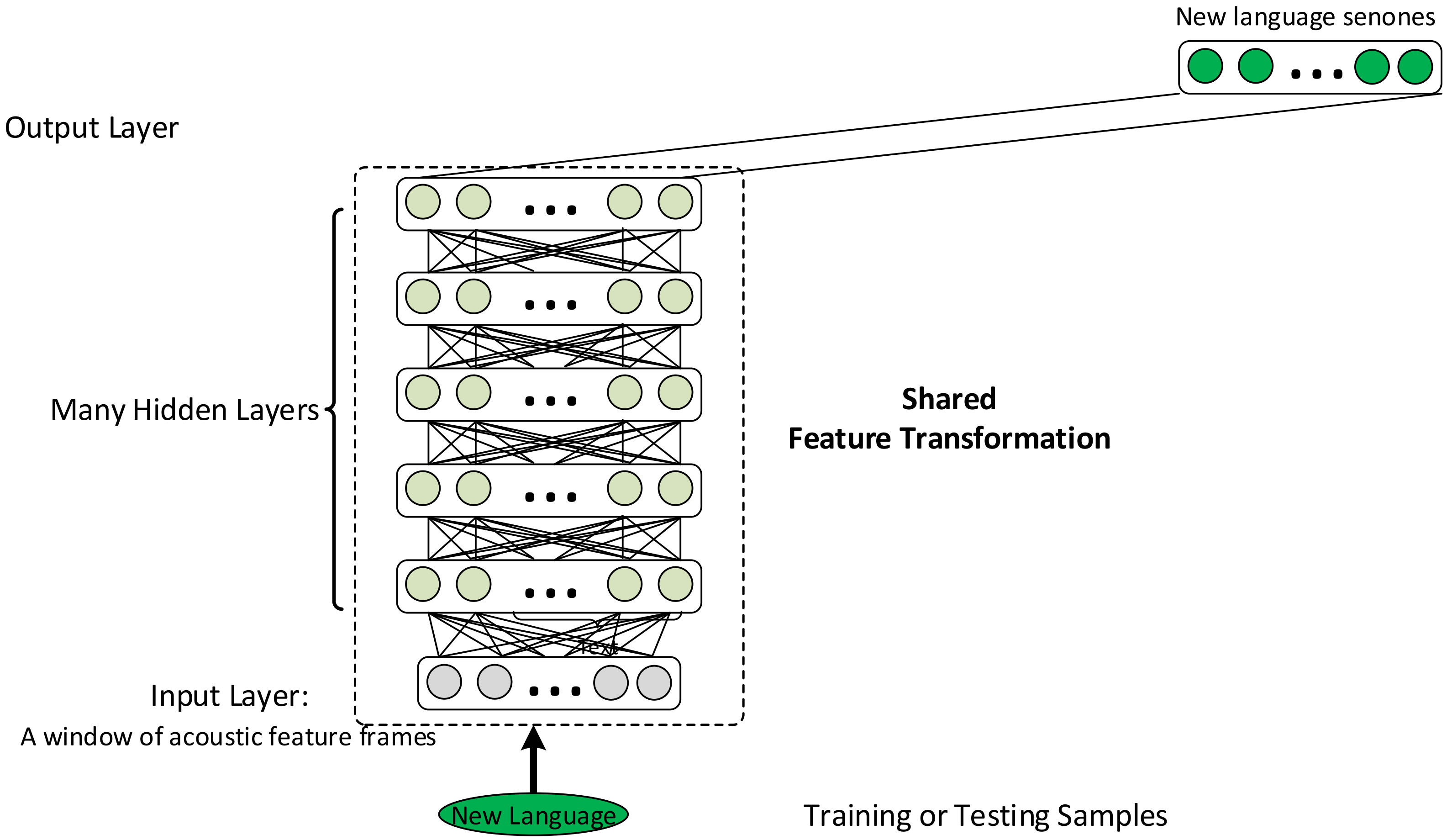
- Develop a new language in new scenario with small amount of training data.

- Leverage the resource-rich languages to develop high-quality ASR for resource-limited languages.

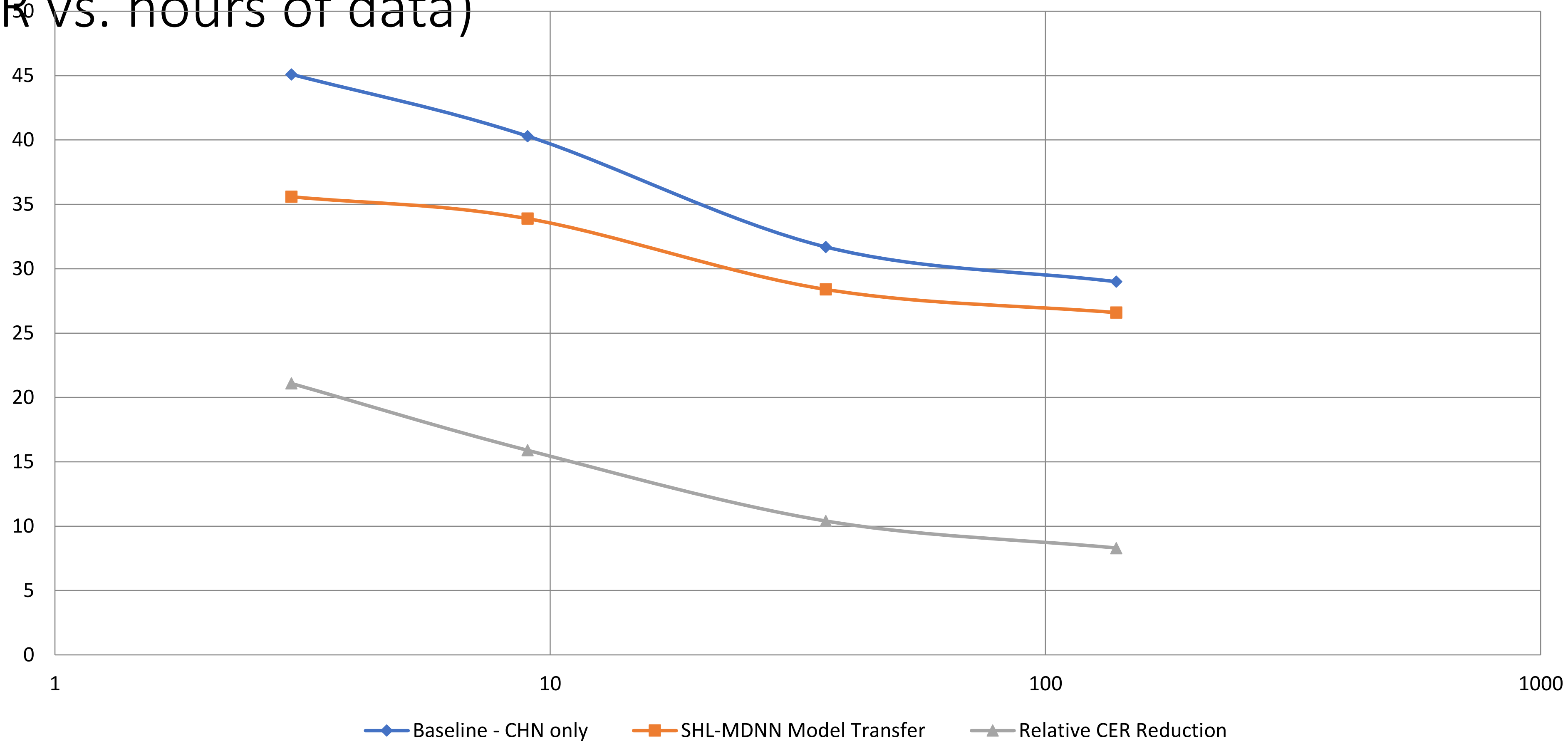
Shared Hidden Layer Multi-Lingual DNN



Adapting to New Language



DNN data reuse: 10-20% WER reduction with data from non-native languages
(WER vs. hours of data)



Target language: zh-CN
Non-native source languages: FRA: 138 hours, DEU: 195 hours, ESP: 63 hours, and ITA: 93 hours of speech.

Reduce Accuracy Gap between Large and Small Deep Networks

[Li14]

To Deploy DNN on Server

- SVD matrices are used to reduce the number of DNN parameters and CPU cost.
- Quantization for SSE evaluation is used for single instruction multiple data processing.
- Frame skipping is used to remove the evaluation of some frames.

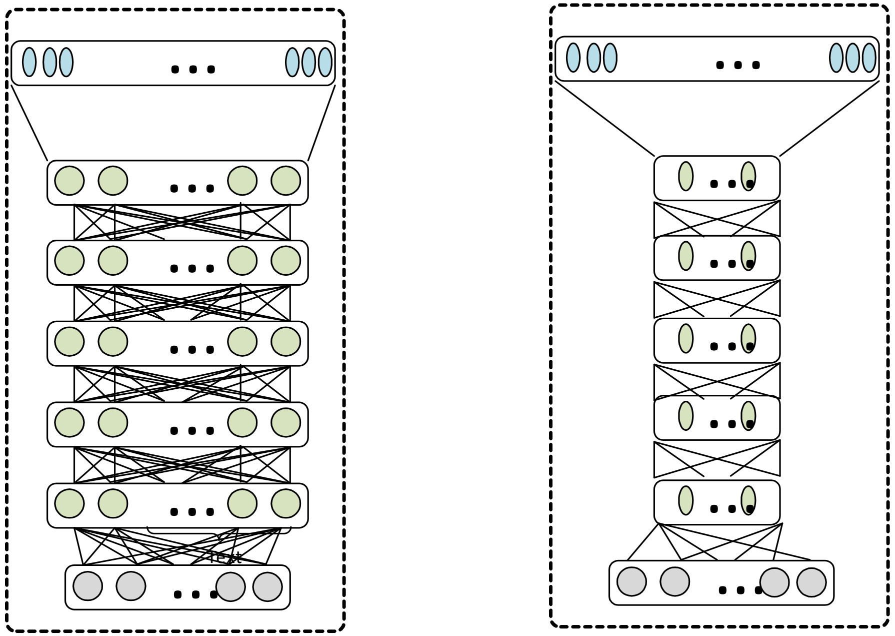
To Deploy DNN on Device

- Even with the technologies mentioned above, the large computational cost is still very challenging due to the limited processing power of devices.

- A common way to fit CD-DNN-HMM on devices is to reduce the DNN model size by
  - reducing the number of nodes in hidden layers
  - reducing the number of targets in the output layer

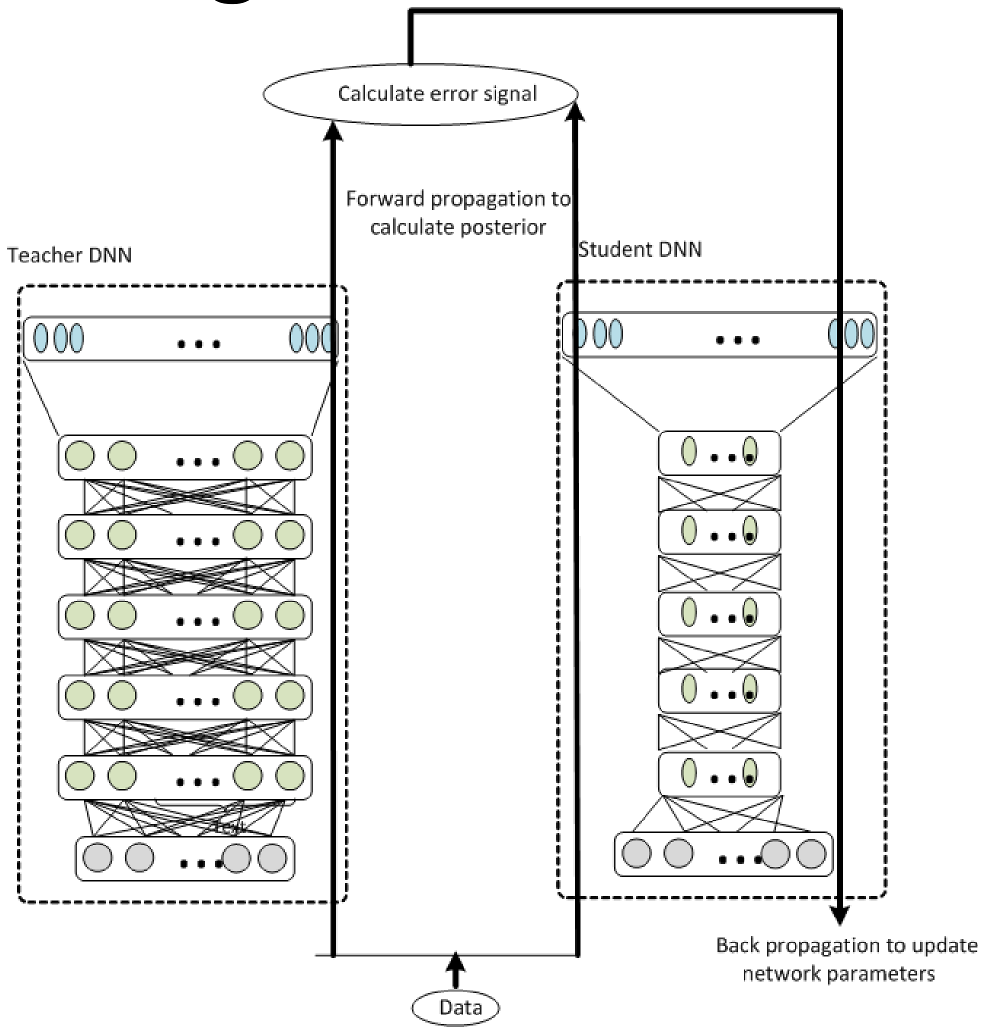
Significant Accuracy Loss when DNN Size Is Significantly Reduced

- Better accuracy is obtained if we use the output of large-size DNN for acoustic likelihood evaluation
- The output of small-size DNN is away from that of large-size DNN, resulting in worse recognition accuracy
- The problem is *solved* if the small-size DNN can generate similar output as the large-size DNN



Teacher-Student Learning

- Minimize the KL divergence between the output distribution of the student DNN and teacher DNN with large amount of un-transcribed data



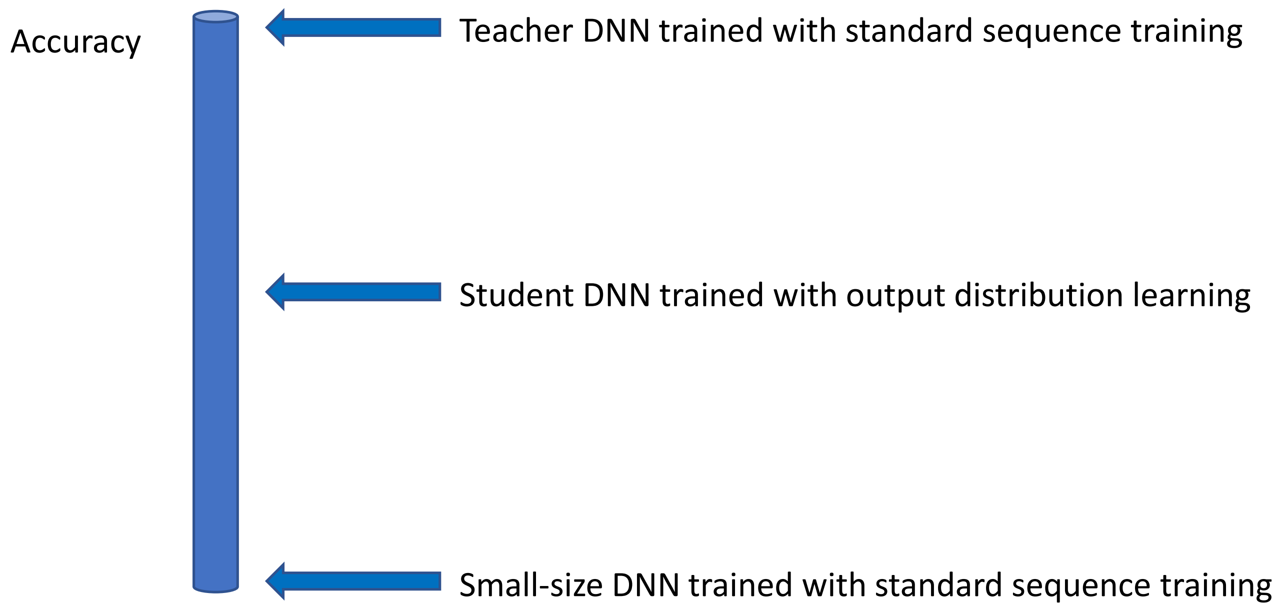
Learning with Soft Targets

teacher-student learning [1]	**knowledge distillation [2]**
$-\sum_f \sum_i P_T(s_i \mid x_{src,f}) log P_S(s_i \mid x_{tgt,f})$	$-(1-\lambda)\sum_f \sum_i P_T(s_i \mid x_{src,f}) log P_S(s_i \mid x_{tgt,f})$ $-\lambda \sum_f log P_S(s_i \mid x_{tgt,f})$
Pure soft target learning	Soft target regularized with hard label from transcription
Can use all available untranscribed data	Limited to available transcribed data

[1] Li, J., Zhao, R., Huang, J.T. and Gong, Y., Learning small-size DNN with output-distribution-based criteria. In Proc. Interspeech, 2014.
[2] Hinton, G., Vinyals, O. and Dean, J., Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

Production Setup

- 2 Million parameter for small-size DNN, compared to 30 Million parameters for teacher DNN.
- The footprint is further reduced to 0.5 million parameter when combining with SVD.



New Domain Adaptation with Parallel Data

[Li17]

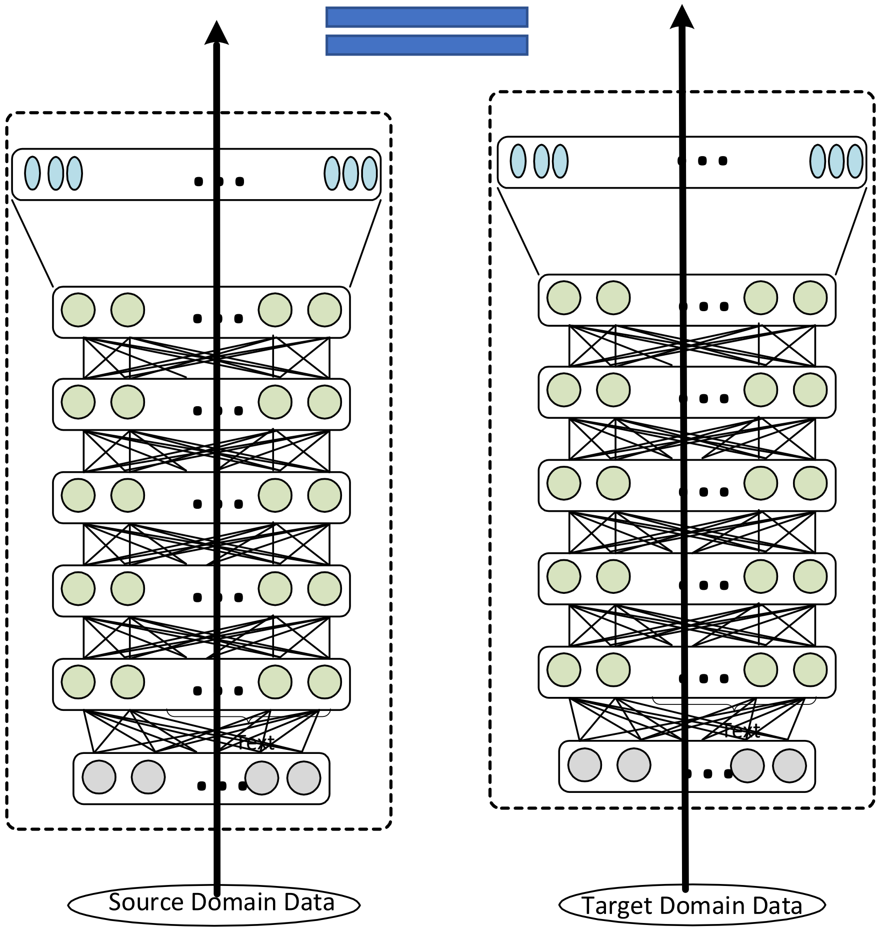
Domain Adaptation

- The success of deep learning relies on a large amount of **transcribed** data
  - The training data is assumed to originate from the distribution as the test data
  - Performance degrades when exposed to test data from a new domain

- It is very expensive to transcribe large amounts of data for a new domain
  - Domain-adaptation approaches have been proposed to bootstrap the training of a new model using an existing well-trained model
    - **Supervised adaptation:** only limited transcribed data is available in new domain
    - **Semi-supervised adaptation:** Estimated hypotheses are typically unreliable in the new domain
    - **Unsupervised adaptation:** does not rely on transcription

How to Train a Good Target Model

- Good accuracy is obtained if we use the output of source-domain DNN with source data for acoustic likelihood evaluation
- The output of target-domain DNN with target data is away from that of source-domain DNN with source data, resulting in worse recognition accuracy
- The problem is *solved* if target-domain DNN with target data can generate similar output as the source-domain DNN with source data

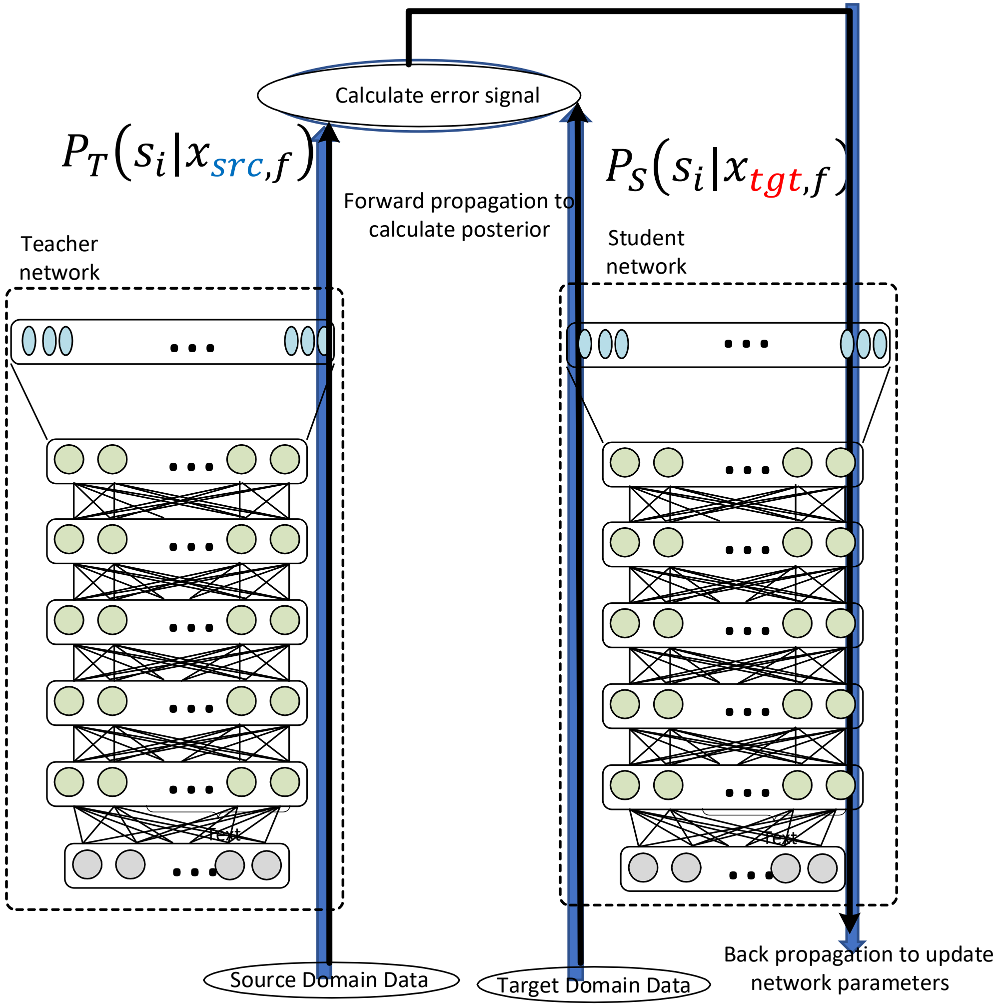


Teacher-Student Learning with Parallel Data

- The behavior of student DNN with target data should be similar to that of the teacher DNN with source data
- Objective function: minimize the KL distance between the teacher and student distributions

$$-\sum_f \sum_i P_T(s_i|x_{src,f}) \log P_S(s_i|x_{tgt,f})$$

- No transcriptions required



Application Scenarios

Source domain	Target domain	How to simulate?
Clean speech	Noisy speech	Add noise
Close-talk speech	Far-field speech	Apply RIR, add noise
Adults	Children	Voice morphing
Original speech	Compressed speech	Apply codec
Wideband speech	Narrowband speech	Downsample/filter

Experimental evaluation

- Baseline model: 4-layer LSTM trained with 375 hours of Cortana data (Microsoft's digital assistant available on many platforms)
- Evaluated using 2 new domains
  - Noisy Cortana
  - CHiME-3

Task	Test utterances	Parallel data
Noisy Cortana task	Simulated noisy speech	clean – simulated noisy speech
CHiME-3 task	Real far-talk speech	close – far talk speech

Noisy Cortana Task

Train Teacher	Train Student	noisy WER	original WER
original 375h	none	18.80	15.62
noisy 375h	none	17.34	16.58
original 375h	original + noisy (375h)	16.66	15.32

Noisy Cortana Task

Train Teacher	Train Student	noisy WER	original WER
original 375h	none	18.80	15.62
noisy 375h	none	17.34	16.58
original 375h	original + noisy (375h)	16.66	15.32
original 375h	original + noisy (3400h)	16.11	15.17

Noisy Cortana Task

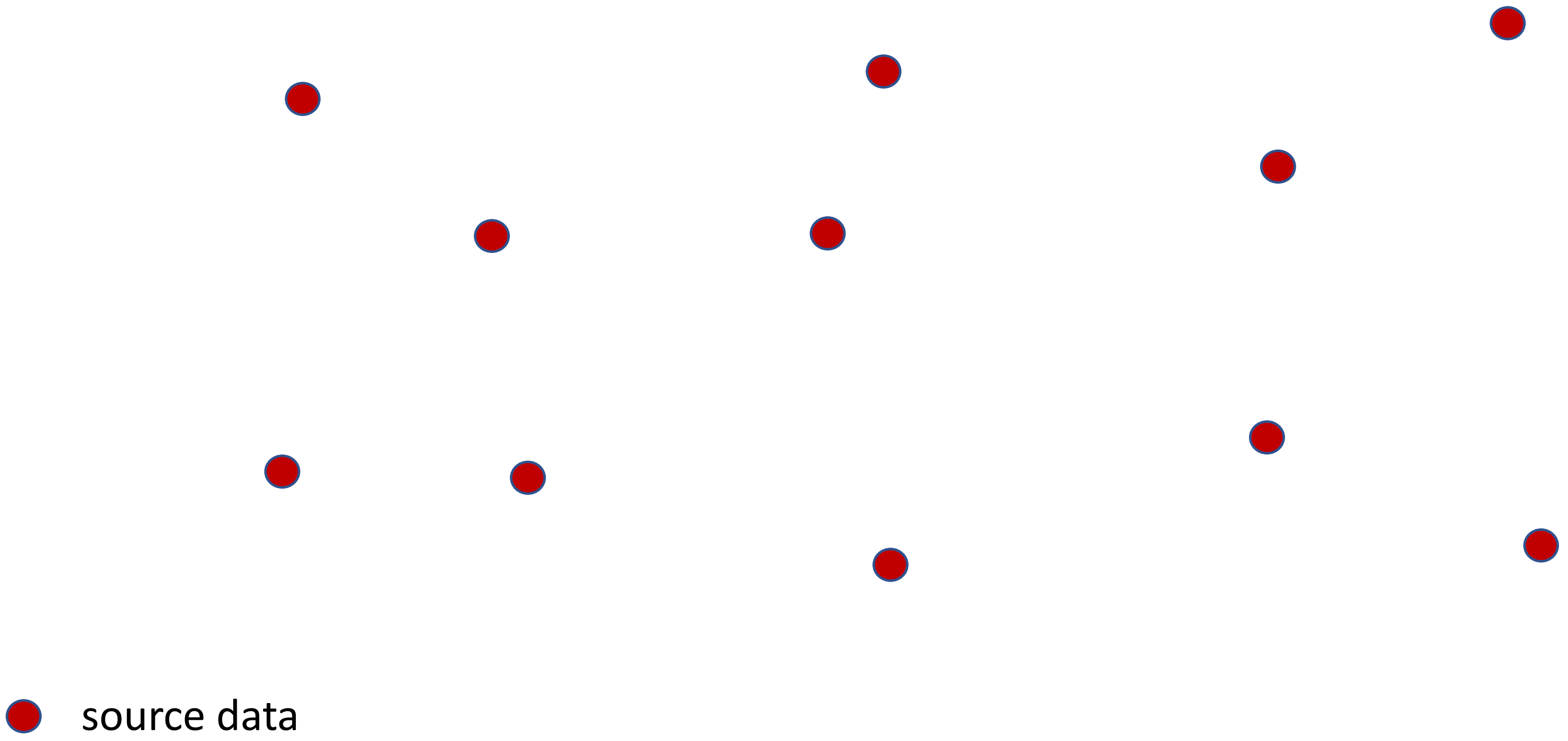
Train Teacher	Train Student	noisy WER	original WER
original 375h	none	18.80	**15.62**
noisy 375h	none	17.34	16.58
original 375h	original + noisy (375h)	16.66	15.32
original 375h	original + noisy (3400h)	**16.11**	15.17

Student network in the target domain is approaching performance of teacher network in the source domain

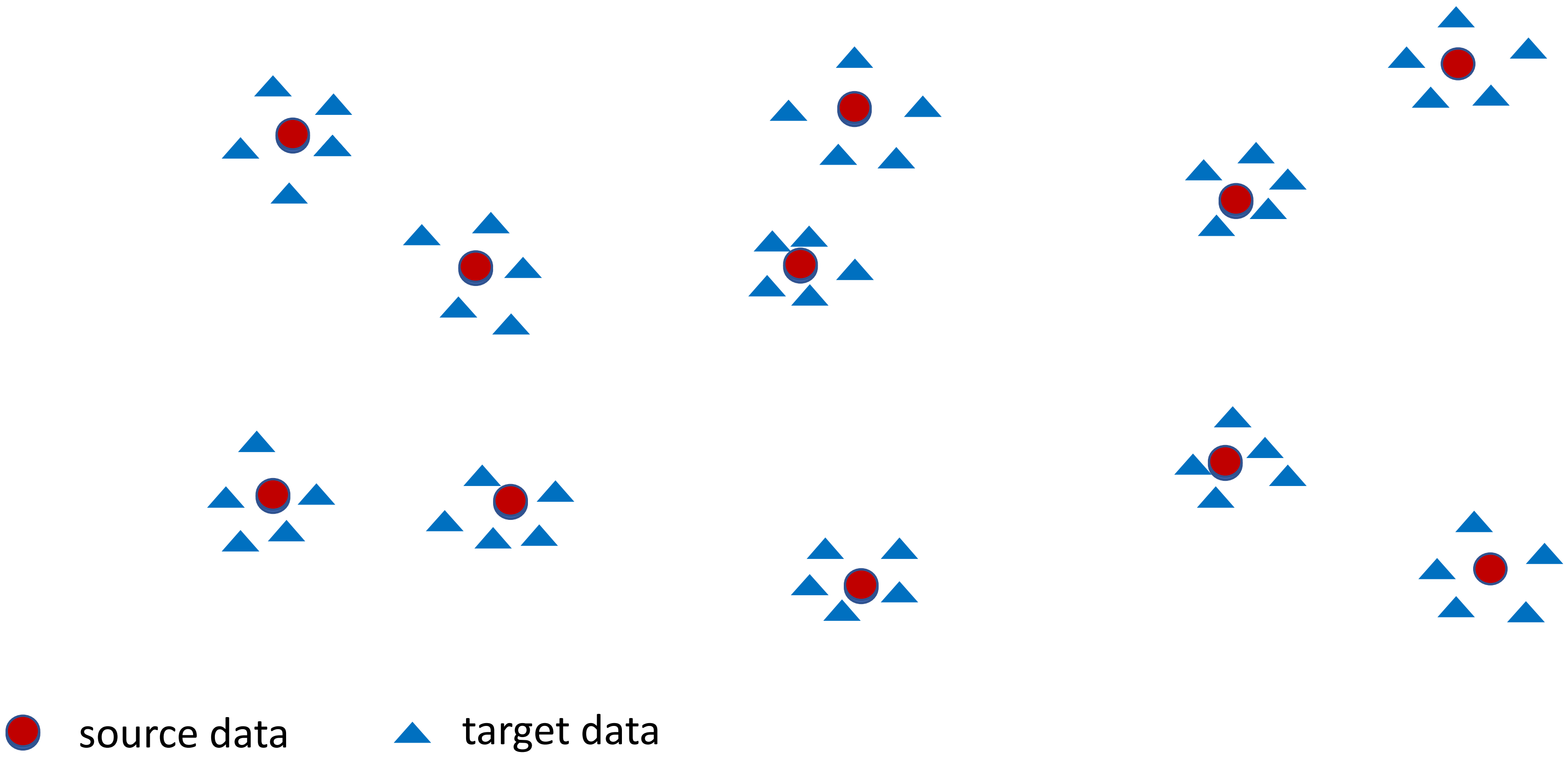
How to Effectively Simulate Data

- Example: Assume we want to use 5X data
- Compare two approaches:
  - Simulate 5 different copies of the transcribed data
  - Simulate 1 copy of 5X larger untranscribed data

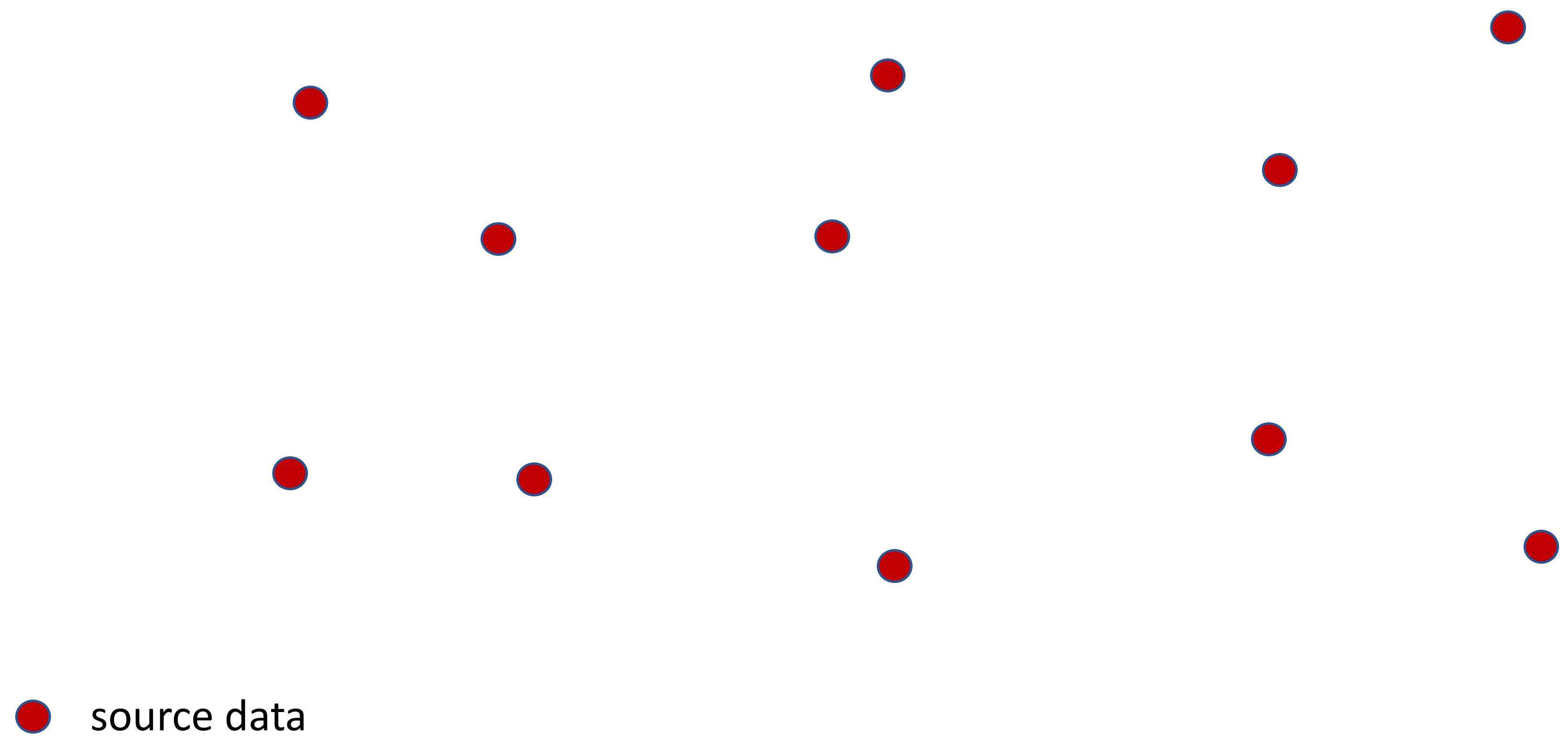
Space of Original Transcribed Data



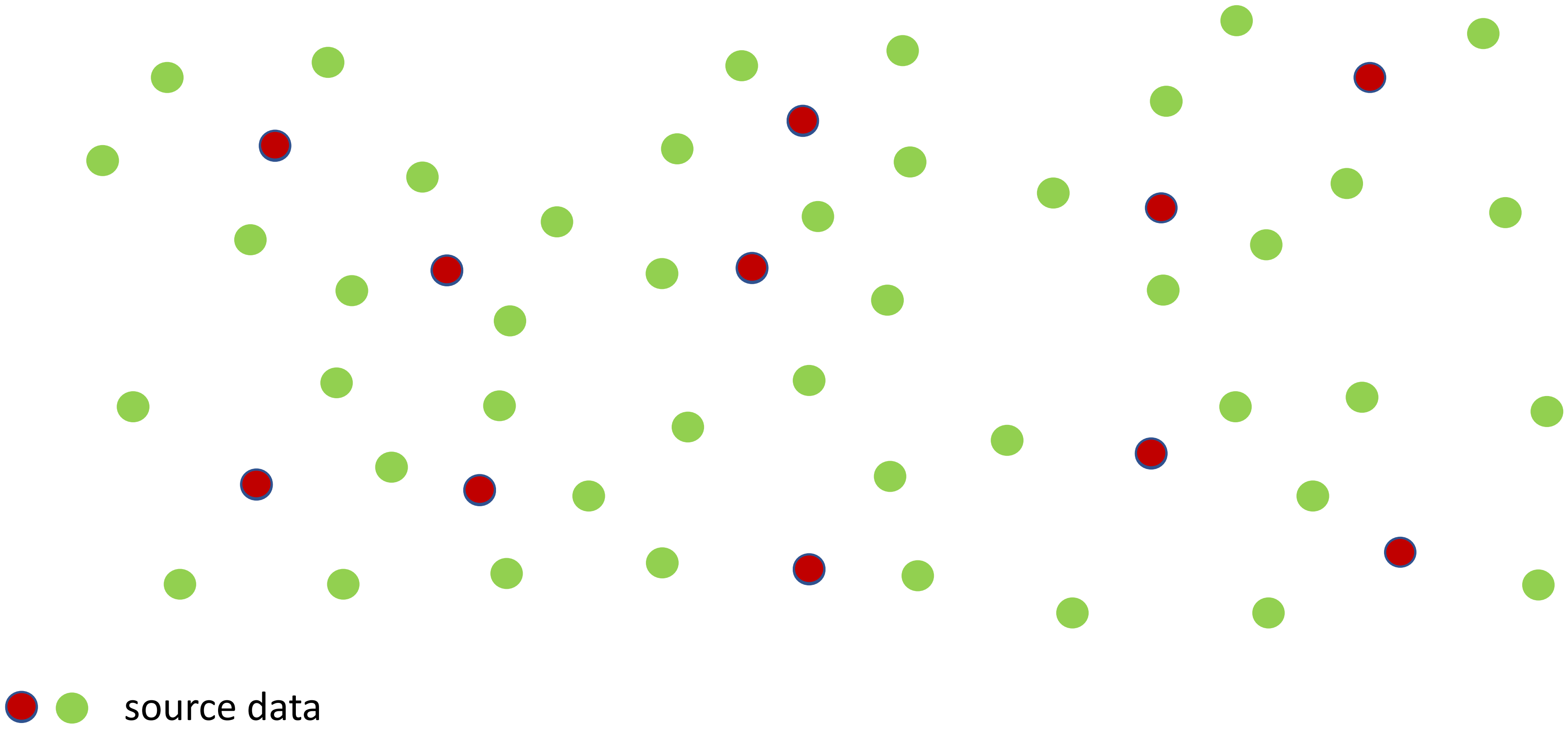
Simulate 5 Copies of the Transcribed Data



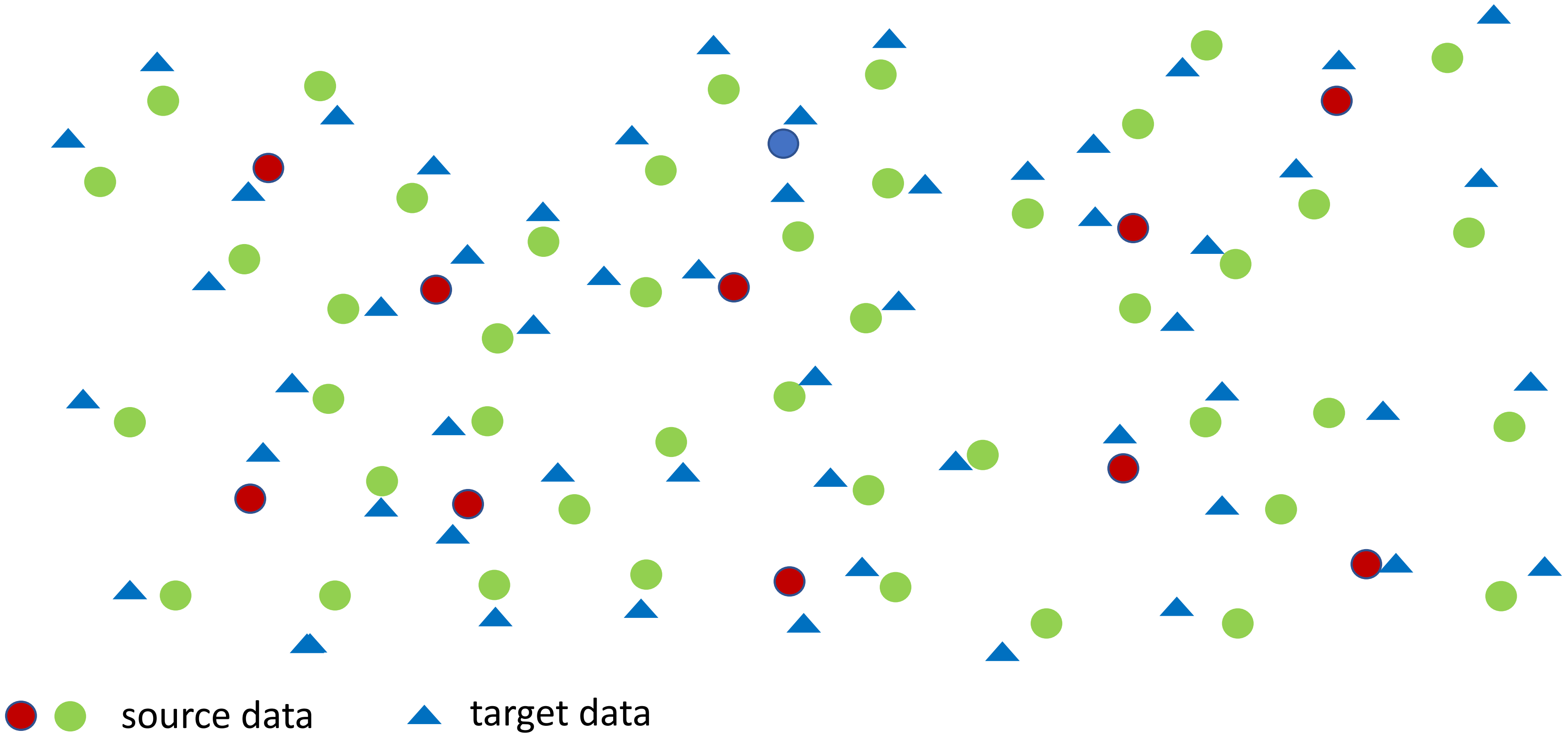
Space of Original Transcribed Data



Space of 5x Untranscribed Data



Simulate 1 Copy of 5x Un-transcribed Data



Chime-3 Task

- Test data more severely mismatched to training data
  - Topic/content mismatched (personal assistant vs. WSJ)
  - Noises/conditions mismatched to adaptation data

Train Teacher	Train Student	Chime-3 WER
original 375h	none	23.16
noisy 375h	none	24.51
original 375h	original + noisy (375h)	23.67
original 375h	original + noisy (3400h)	**19.89**

- Increasing the amount of parallel training data helps the student model more of the acoustic space

Chime-3 with Smaller Well-matched Parallel Corpus

- Matched real data significantly improves the performance of T/S learning

The noisy data in the pair comes from				
Real channel 5	Simulated channel 5	Other real channels	Simulated other channels	**WER**
Y	N	N	N	15.88

Chime-3 with Smaller Well-matched Parallel Corpus

- Matched simulated data also improves the performance of T/S learning

The noisy data in the pair comes from				
Real channel 5	Simulated channel 5	Other real channels	Simulated other channels	WER
Y	N	N	N	15.88
N	Y	N	N	15.73

Chime-3 with Smaller Well-matched Parallel Corpus

- With both real and simulated data, T/S learning can get further improved.

The noisy data in the pair comes from				
Real channel 5	Simulated channel 5	Other real channels	Simulated other channels	**WER**
Y	N	N	N	15.88
N	Y	N	N	15.73
Y	Y	N	N	13.77

Chime-3 with Smaller Well-matched Parallel Corpus

- More data gives better performance
  - Significantly better than feature mapping and mask learning [3]

The noisy data in the pair comes from				
Real channel 5	Simulated channel 5	Other real channels	Simulated other channels	WER
Y	N	N	N	15.88
N	Y	N	N	15.73
Y	Y	N	N	13.77
Y	Y	Y	Y	12.99

[3] Z. Chen, Y. Huang, J. Li, and Y. Gong, "Improving mask learning based speech enhancement system with restoration layers and residual connection," in Proc. Interspeech, 2017.

When Baseline Model is Trained with 3400hr Transcribed Data

- Evaluated with multiple scenarios – real test utterances

Model	Test0	Test1	Test2	Test3	Test4	Test5
3.4k hour-transcribed Teacher	62.36					

When Baseline Model is Trained with 3400hr Transcribed Data

- Evaluated with multiple scenarios – real test utterances: T/S learning with simulation works very well for real target-domain speech

Model	Test0	Test1	Test2	Test3	Test4	Test5
3.4k hour-transcribed Teacher	62.36					
T/S with 3.4k hour paired data	17.22	12.78	9.19	14.65	13.89	25.90

When Baseline Model is Trained with 3400hr Transcribed Data

- Evaluated with multiple scenarios – real test utterances: T/S learning with simulation works very well for real target-domain speech

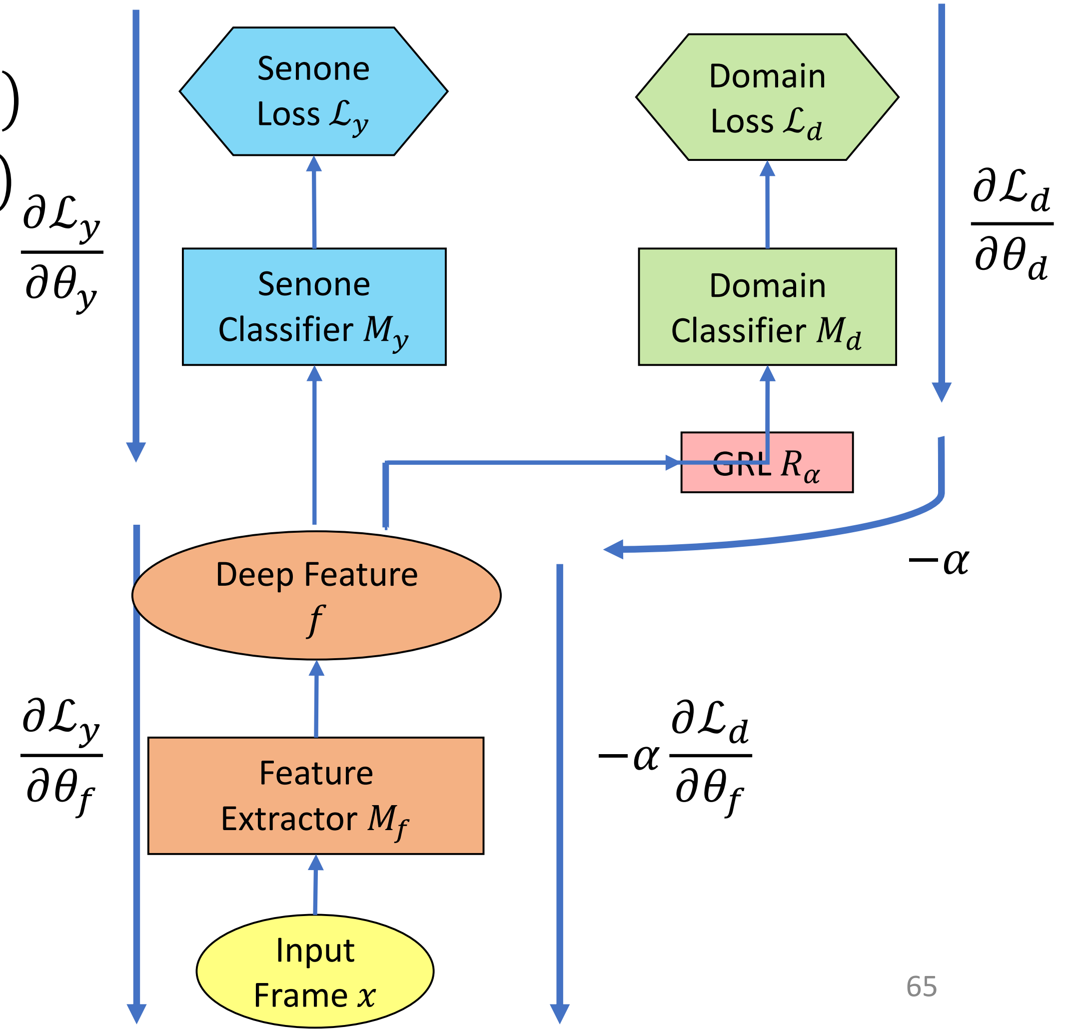
Model	Test0	Test1	Test2	Test3	Test4	Test5
3.4k hour-transcribed Teacher	62.36					
T/S with 3.4k hour paired data	17.22	12.78	9.19	14.65	13.89	25.90
T/S with 25k hour paired data	15.66	12.35	8.95	12.90	12.23	20.79

New Domain Adaptation without Parallel Data

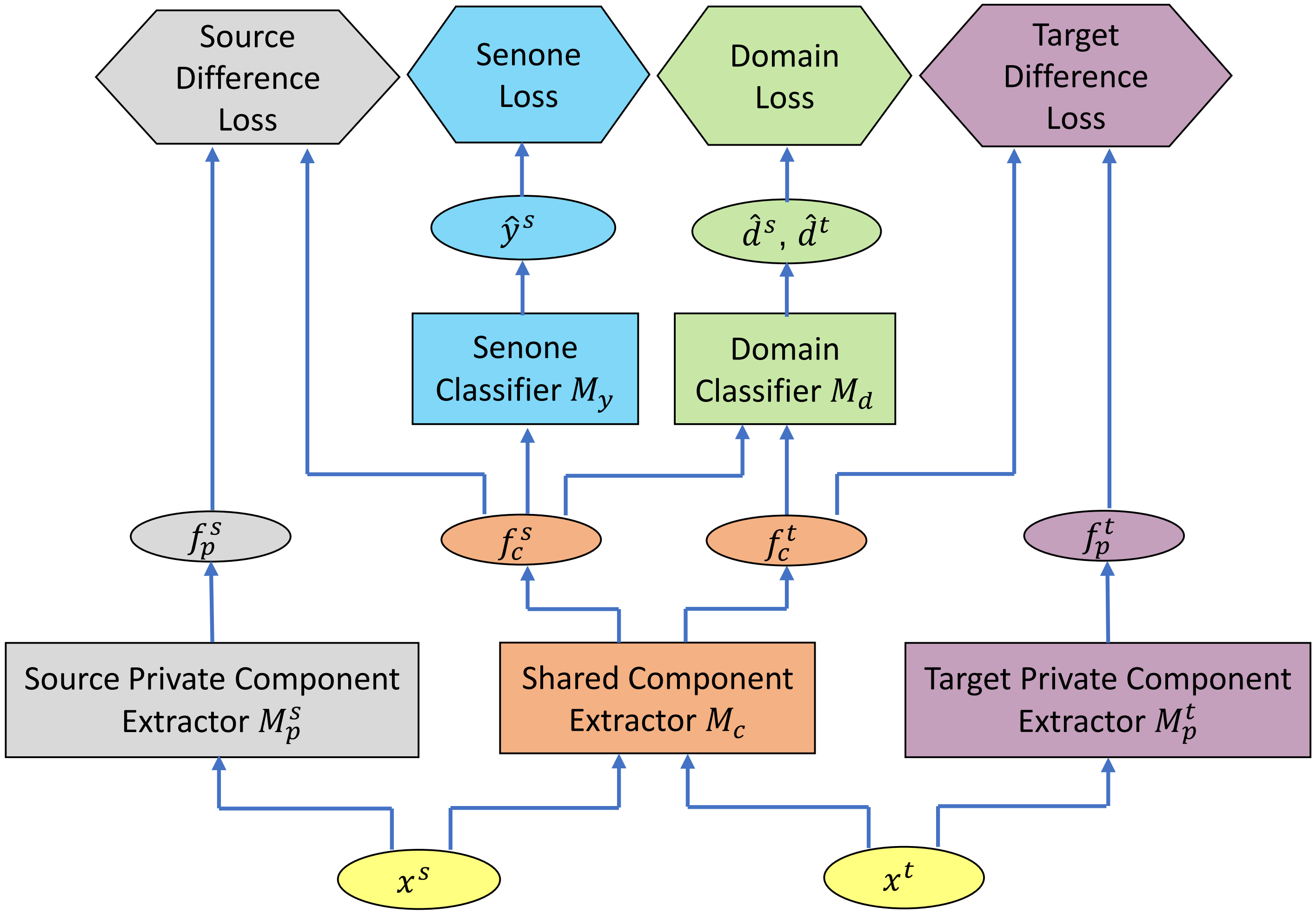
[Meng17]

Domain-Invariant Training of Acoustic Model: Gradient Reversal Layer Network (GRLN)

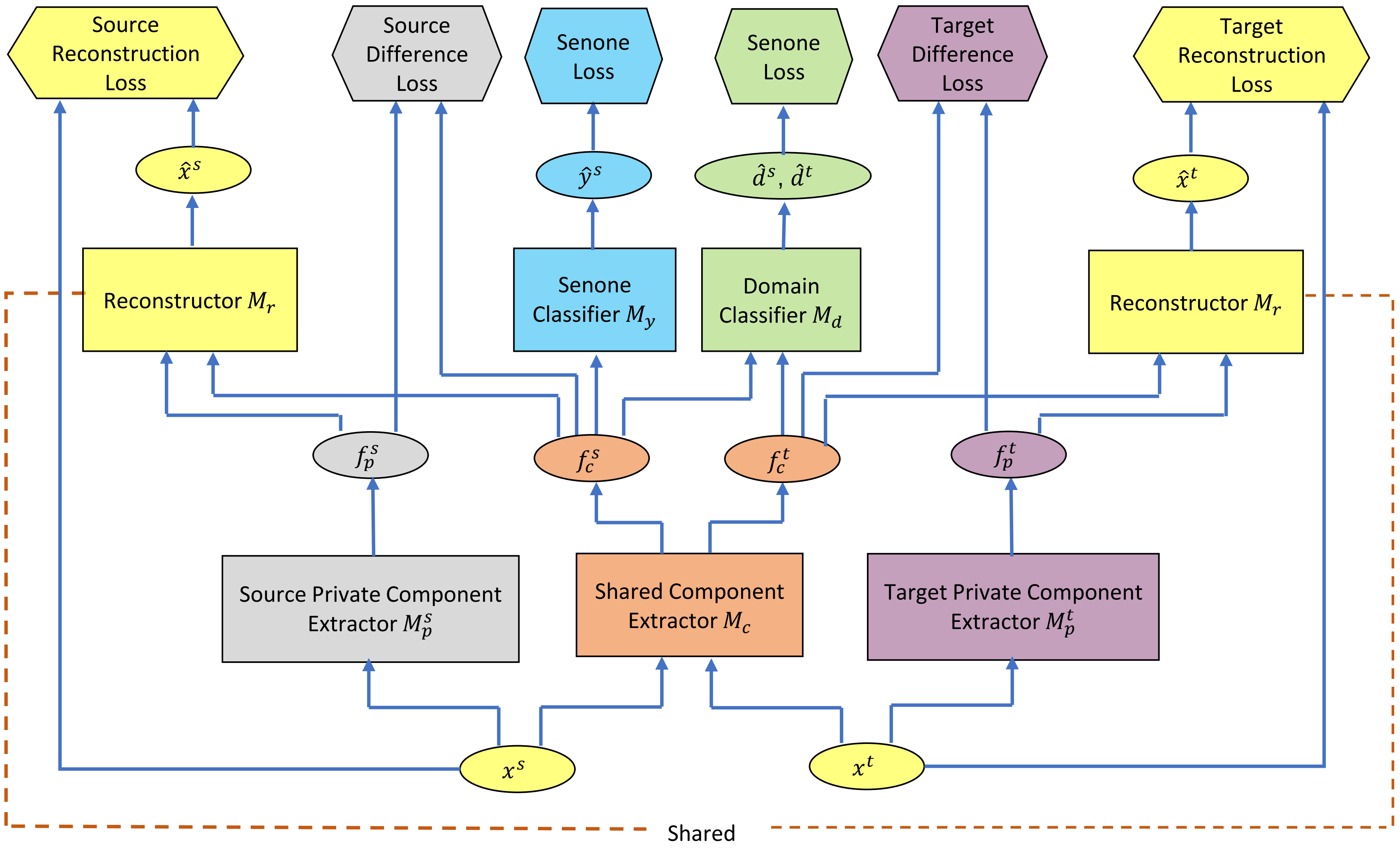
- Adversarial Multi-Task Learning
  - Senone-discriminative: $\min\limits_{\theta_y, \theta_f} \mathcal{L}_y(\theta_y, \theta_f)$
  - Domain-invariant: $\max\limits_{\theta_f} \min\limits_{\theta_d} \mathcal{L}_d(\theta_d, \theta_f)$
  - Multi-task: $\max\limits_{\theta_f} \min\limits_{\theta_y, \theta_d} \left[\mathcal{L}_y(\theta_y) + \alpha \mathcal{L}_d(\theta_d, \theta_f)\right]$
- Stochastic Gradient Decent
  - $\theta_y \leftarrow \theta_y - \mu \frac{\partial \mathcal{L}_y}{\partial \theta_y}$
  - $\theta_f \leftarrow \theta_f - \mu \left[\frac{\partial \mathcal{L}_y}{\partial \theta_f} - \alpha \frac{\partial \mathcal{L}_d}{\partial \theta_f}\right]$
  - $\theta_d \leftarrow \theta_d - \mu \frac{\partial \mathcal{L}_d}{\partial \theta_d}$
- Gradient Reversal Layer R_α
  - Forward pass: $R_\alpha(f) = f$
  - Backward pass: $\frac{\partial R_\alpha(f)}{\partial f} = -\alpha I$
  - I is the identity matrix



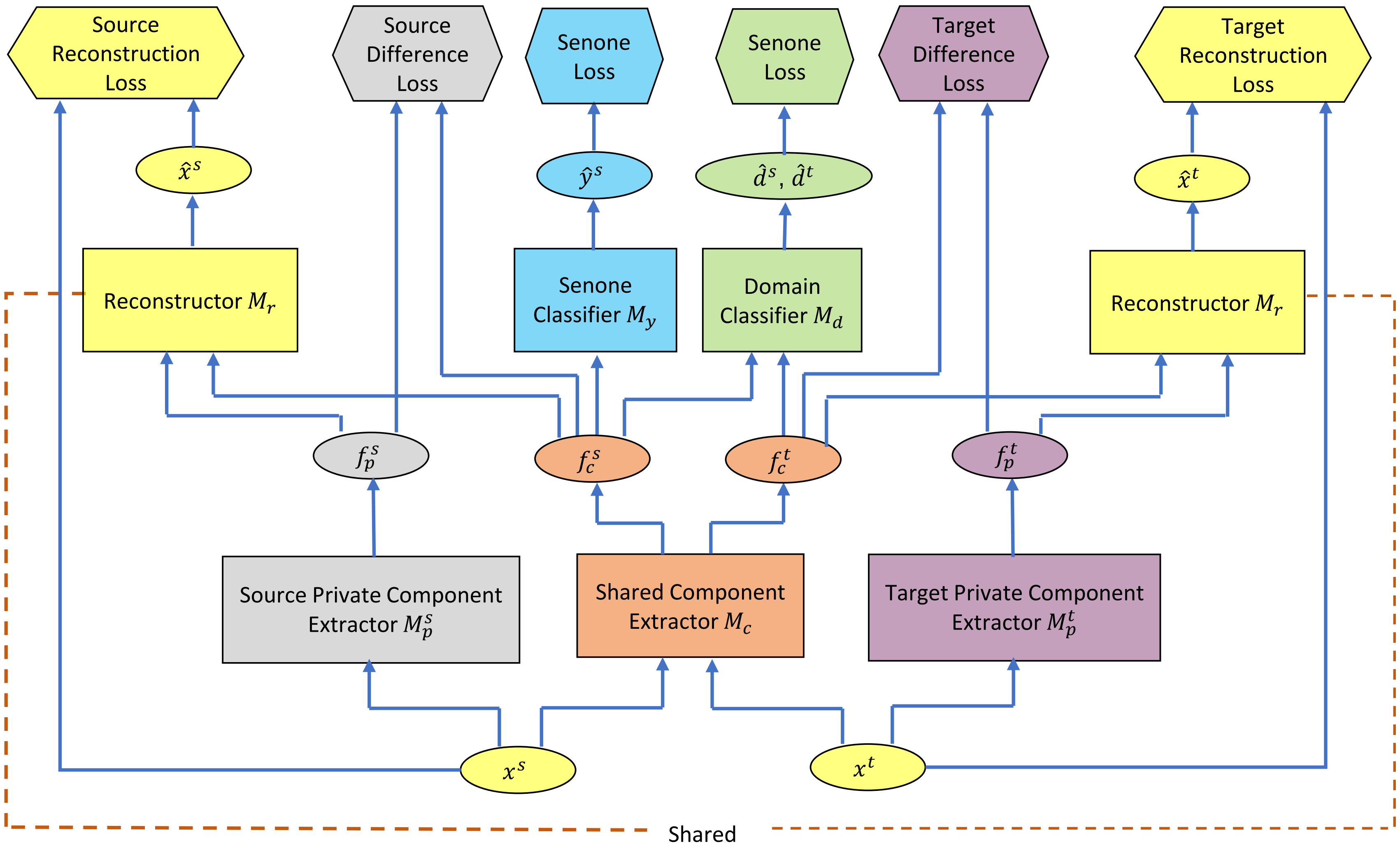
Private Component Extractor



Reconstructor



Adversarial Training of Domain Separation Network



ASR Results of DSN for Unsupervised Environment Adaptation

- Test data: CHiME-3 dev set with 4 noise conditions
- WSJ 5K word 3-gram language model is used for decoding

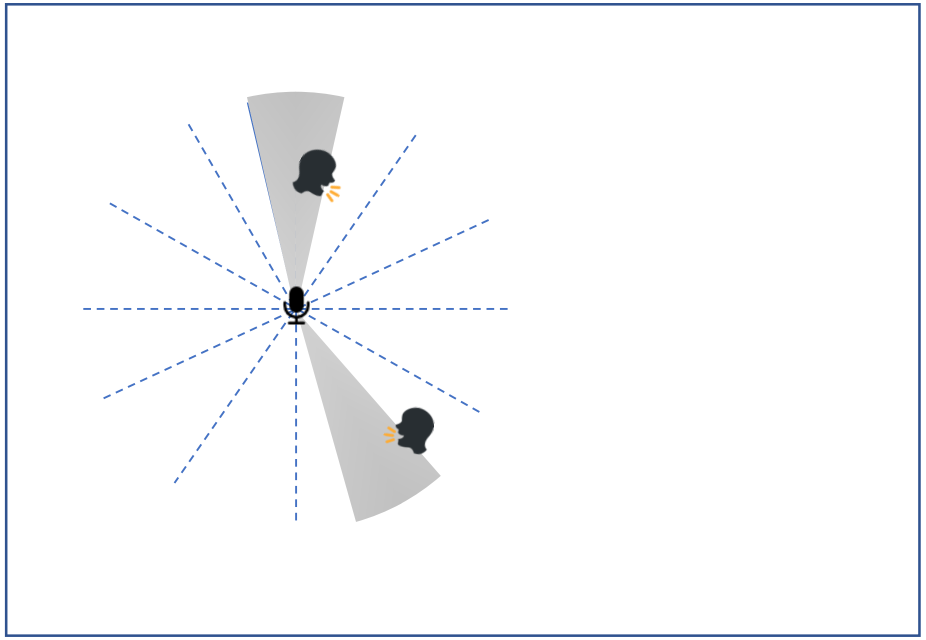
System	Data	BUS	CAF	PED	STR	Avg.
Clean	Real	36.25	31.78	22.76	27.18	29.44
GRL	Real	35.93	28.24	19.58	25.16	27.16
DSN	Real	32.62	23.48	17.29	23.46	**24.15**

Multi-talker Separation

[Chen17]

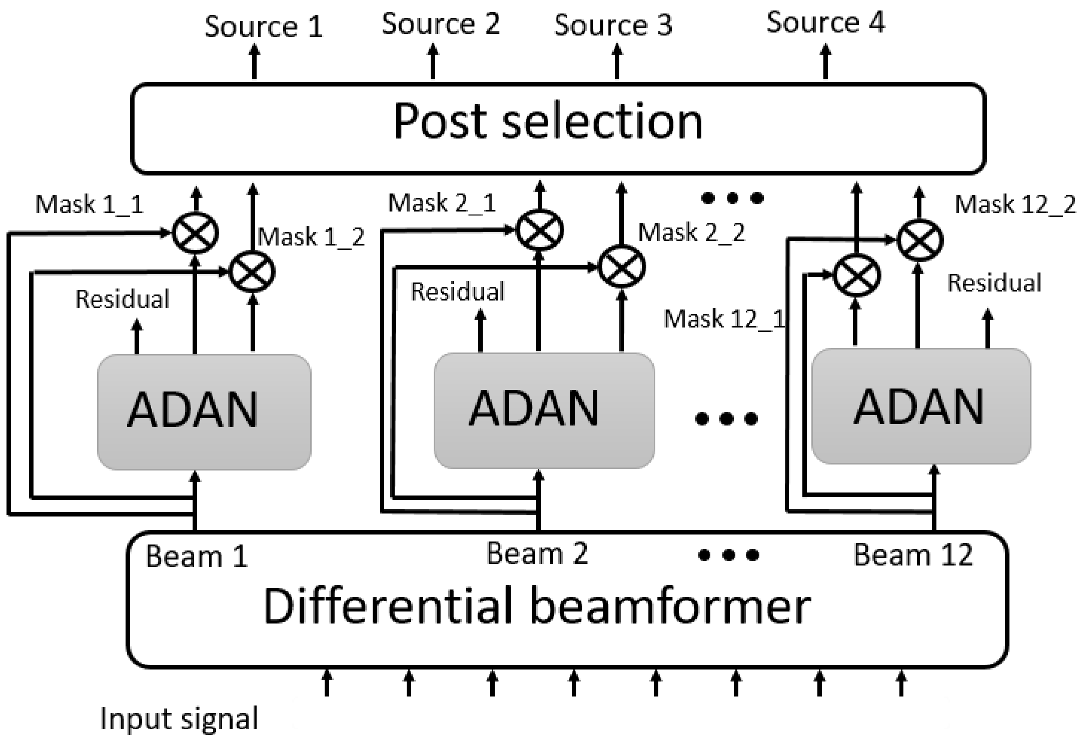
Solving the cocktail problem

- Multi-talker speech separation & recognition
  - Separate and recognize each speaker in highly overlapped environment, e.g. cocktail party
  - The speaker identity and number of speakers are unknown
- Difficulty
  - Tracking multiple speaker largely increase the data & computation complexity
  - Unknown number of speaker is troublesome to neural networks
  - Permutation problem
- Single channel solution
  - Deep clustering/ deep attractor network
  - Permutation Invariation training
- Limitations of single channel processing
  - Performance is still unsatisfactory
  - Difficult to deal with reverberation
  - Multi-channel signal provides spatial clues, which is beneficial for separation



System Architecture

- A fixed set of beamformer
  - 12 fixed differential beamformer, uniformly sample the space
  - A linear operation for beamformer
- Separation network
  - Anchored deep attractor network
  - Pick best two speakers for each beam
  - Additional residual more for noise
- Post selection
  - Selecting each speaker from all 24 outputs
  - Spectral clustering to group the classes
  - Speech quality evaluation to pick best speech for each group



System Architecture

State of the art separation performance

- A new state of the art for multi-talker separation & recognition
  - Similar performance as the ideal ratio mask and the oracle mvdr beamformer
  - Largely improve the single channel system
  - Robustly separating 4 overlapped speakers
  - Significantly improvement for multi-talker speech recognition
- Still a room to further improve
  - Acoustic model retraining/ joint training
  - Mask based beamformer from the separated result
- Example:
  - The sample that has the median performace
  - Mixture:
  - Result:

	Proposed	**IRM**	**OMVDR**	**DAN**
2 speaker	+10.98	+11.05	+12.00	+7.82
3 speaker	+11.54	+11.52	+12.56	+5.16
4 speaker	+11.19	+12.22	+11.82	+4.23

Separation result SDR(Db)

Clean model	Mixture	Top 1	Top 2	Top3	Top4
2 speaker	82.29	29.85	31.38	-	-
3 speaker	93.61	31.8	39.21	44.89	-
4 speaker	95.97	42.31	46.54	53.68	65.67
Far-field model	Mixture	Top 1	Top 2	Top3	Top4
2 speaker	81.96	23.6	26.38	-	-
3 speaker	94.19	27.95	32.64	40.61	-
4 speaker	95.91	37.79	40.29	46.1	57.93

Recognition Result

Reference

- [Chen17] Zhuo Chen, Jinyu Li, Xiong Xiao, Takuya Yoshioka, Huaming Wang, Zhenghao Wang, Yifan Gong, "CRACKING THE COCKTAIL PARTY PROBLEM BY MULTI-BEAM DEEP ATTRACTOR NETWORK", in ASRU, 2017.
- [Huang13] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, cross-language knowledge transfer using multilingual deep neural network with shared hidden layers, in ICASSP, 2013
- [Li14] Jinyu Li, Rui Zhao, Jui-Ting Huang and Yifan Gong, Learning Small-Size DNN with Output-Distribution-Based Criteria, in *Interspeech*, 2014.
- [Li15] Jinyu Li, Abdelrahman Mohamed, Geoffrey Zweig, and Yifan Gong, “LSTM time and frequency recurrence for automatic speech recognition,” in Proc. ASRU, 2015.
- [Li16] Jinyu Li, Abdelrahman Mohamed, Geoffrey Zweig, Yifan Gong, “Exploring Multidimensional LSTMs for Large Vocabulary ASR,” in Proc. ICASSP, 2016.
- [Li17] Jinyu Li, Michael L Seltzer, Xi Wang, Rui Zhao, and Yifan Gong, “Large-scale domain adaptation via teacher student learning,” in *Interspeech*, 2017.
- [Meng17] Zhong Meng, Zhuo Chen, Vadim Mazalov, Jinyu Li, Yifan Gong, "Unsupervised Adaptation with Domain Separation Networks for Robust Speech Recognition“, in ASRU, 2017.
- [Miao16] Yajie Miao, Jinyu Li, Yongqiang Wang, Shi-Xiong Zhang, Yifan Gong, “Simplifying Long Short-term Memory Acoustic Models for Fast Training and Decoding,” in Proc. ICASSP, 2016.
- [Xue13] Jian Xue, Jinyu Li, and Yifan Gong, Restructuring of Deep Neural Network Acoustic Models with Singular Value Decomposition, in *Interspeech*, 2013
- [Xue 14] Jian Xue, Jinyu Li, Dong Yu, Mike Seltzer, and Yifan Gong, Singular Value Decomposition Based Low-footprint Speaker Adaptation and Personalization for Deep Neural Network, in *ICASSP*, 2014