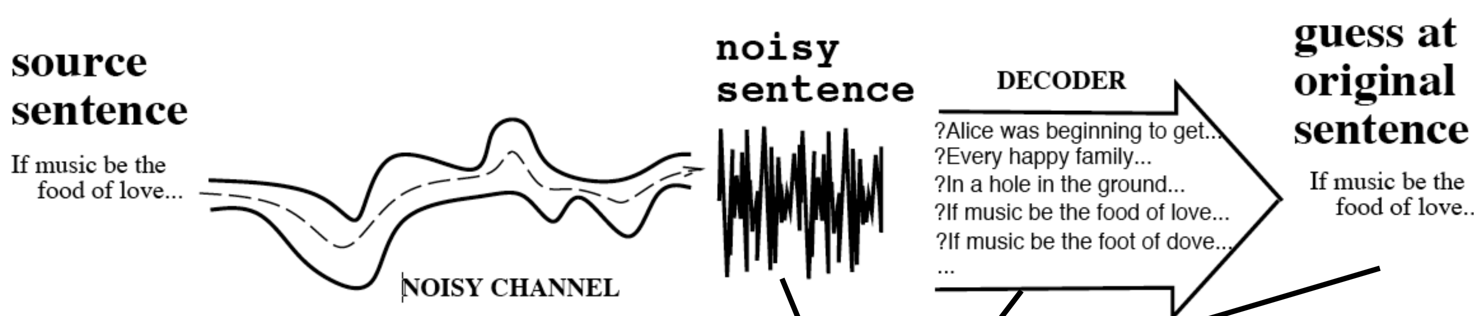
# Channel Mismatch Adaptation for DNNs

Graduate Research Proposal

Animesh Prasad
Advisor: Khe Chai SIM

# Statistical Automatic Speech Recognition (ASR) Formulation

Noisy Channel Model




$$\hat{W} = \underset{W \in L}{\arg\max}\, P(W \mid O)$$

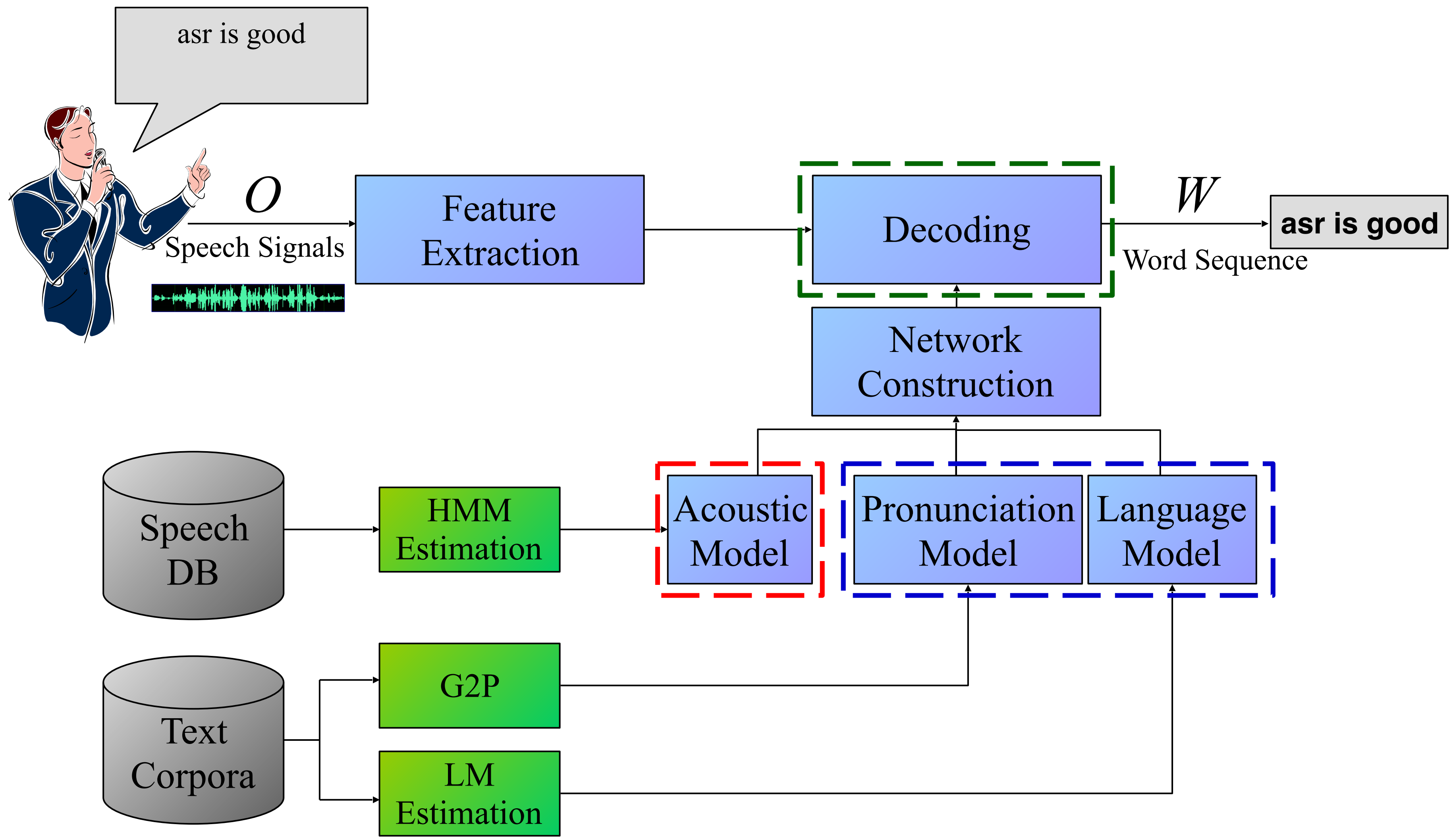
$$\hat{W} = \underset{W \in L}{\arg\max}\, P(O \mid W) P(W)$$

# Statistical ASR Formulation

$$\hat{W} = \underset{W \in L}{\arg\max}\, P(O \mid W)\, P(W)$$

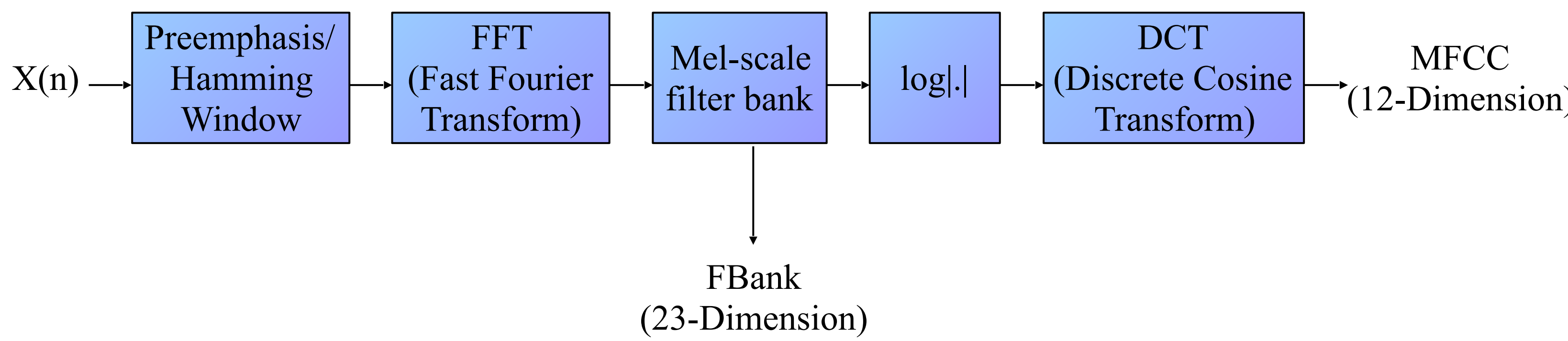
Acoustic Model ⟵ Lexicon ⟶ Language Model

# ASR Pipeline




$$\hat{W} = \arg\max_{W \in L} P(O \mid W) P(W)$$

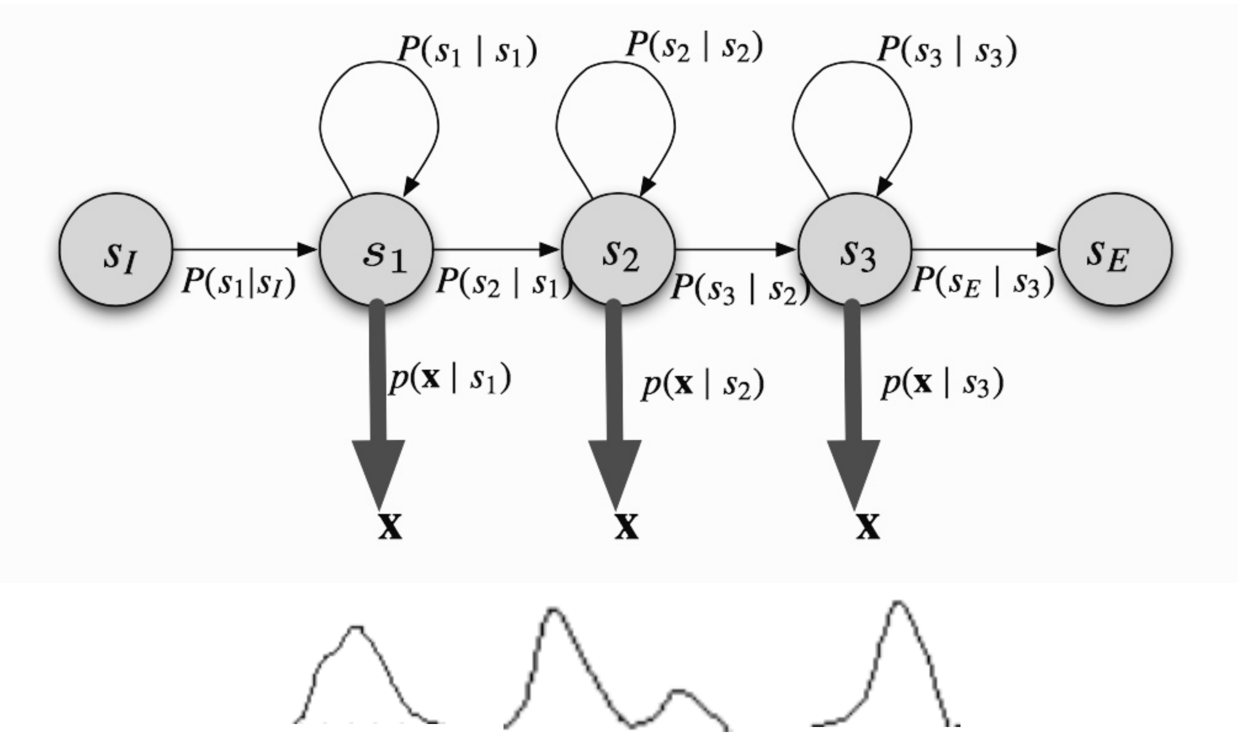
# Feature Extraction



Acoustic Model

# Acoustic Modelling

## Hidden Markov Model

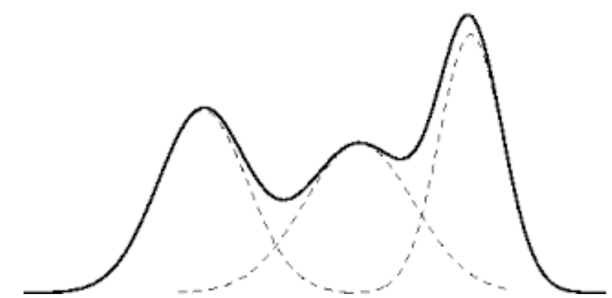


Acoustic Model

# Modelling State

## Gaussian Mixture Model (GMMs)

$$b_j(x_j) = \sum_{k=1}^{K} c_{jk} N(x_t; \mu_{jk}, \Sigma_{jk})$$

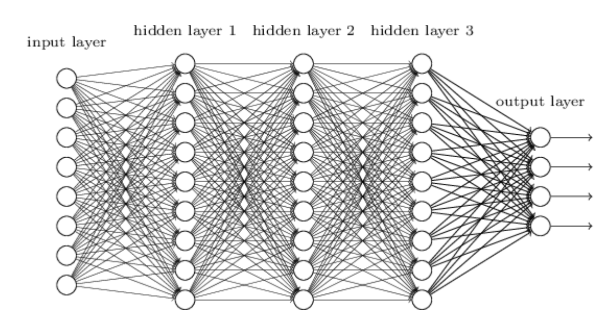


## Deep Neural Network (DNNs)

Posterior

$$b_j(x_j) = p(x_t|s_t = s) = \frac{p(s_t = s|x_t)p(x_t)}{p(s)}$$

Likelihood



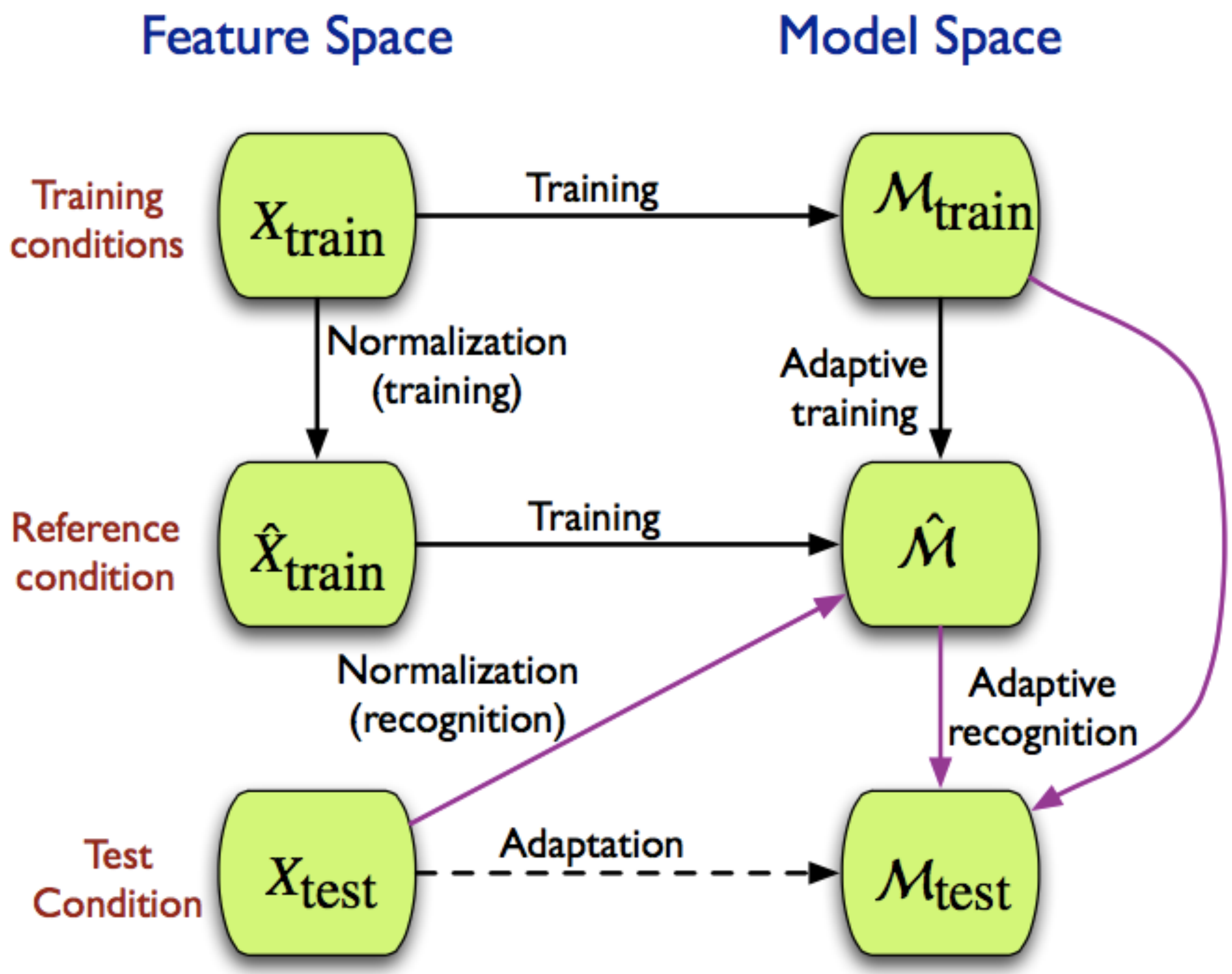
# Need of Adaptation

Training and testing condition mismatch

Speaker, speaking rate, background noise, reverberation, channel (speaker microphone distance), etc.

Either bring model close to test condition or vice versa

# Adaptation Schema



# Prior Work

| Adaptation Techniques | Compensation | Applicable On | Applied For |
|---|---|---|---|
| MAP | Model | GMM | Speaker |
| MLLR, cMLLR | Model | GMM | Speaker |
| fMLLR, SAT | Feature | GMM/DNN | Speaker |
| CMV, CVN, CMVN | Feature | GMM/DNN | Speaker, Noise |
| VTLN | Feature | GMM/DNN | Speaker |
| VTS | Model | GMM | Noise |
| RASTA Filtering | Feature | GMM/DNN | Noise, Reverberation, Channel |
| LIN, LON, LHN | Model | DNN | Speaker |
| Retraining | Model | DNN | Speaker, Noise, Channel |
| Regularization | Model | DNN | Speaker |
| Dropout | Model | DNN | Noise |
| Low Rank Approximation | Model | DNN | Speaker |
| Condition Aware Training | Model | DNN | Speaker, Noise, Channel |

# Channel (Speaker Microphone Distance) Adaptation

## Need

Natural interfaces (HCI), application like smart houses

## Current Strategy

Feature Space( eg. Beam-forming)

## Scope

Word Error Rate (WER) : Close talk ASR approx. 10-20, far field ASR approx. 30-40

## Consideration

During testing the source distance might be know or unknown,

# Data Preparation

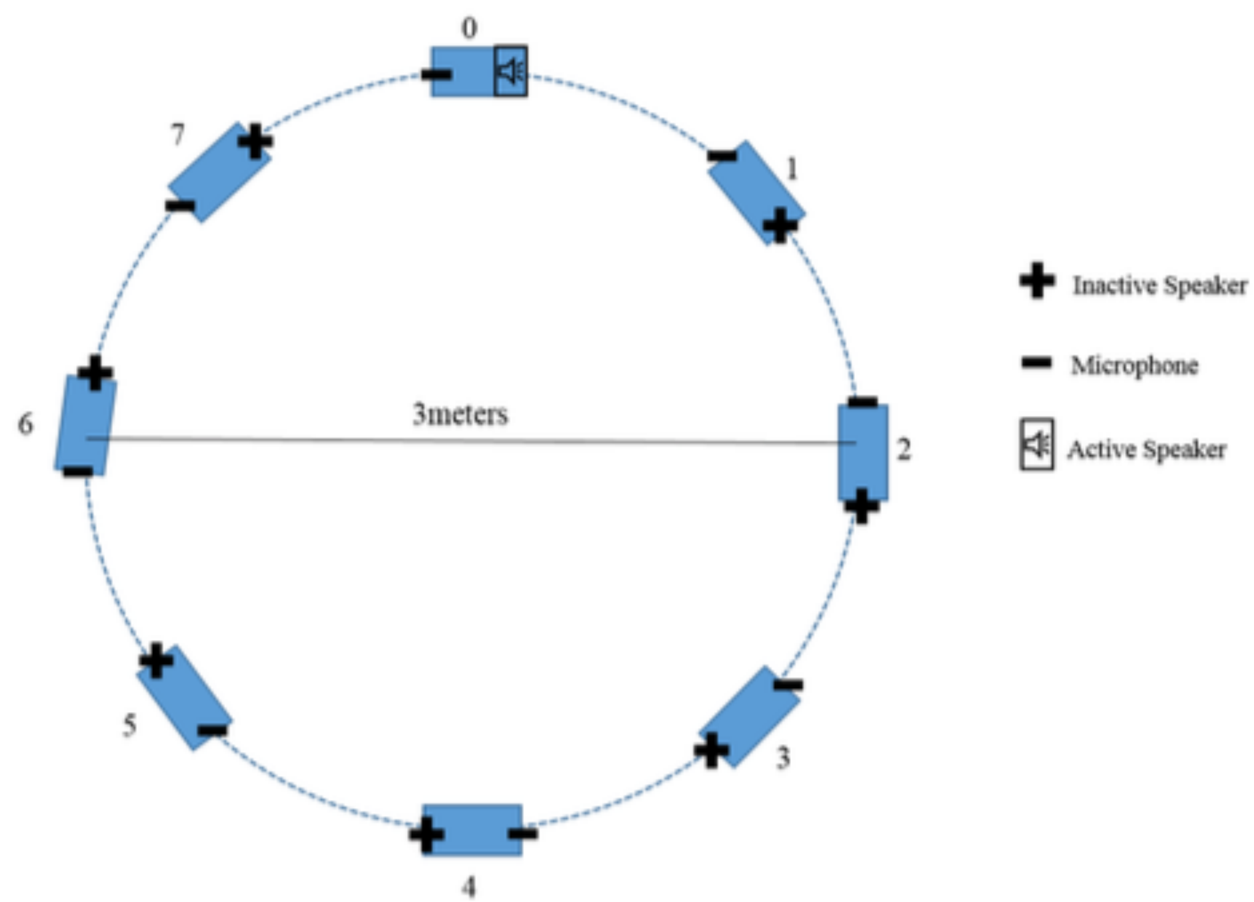
## Basic features of data

Multichannel version of WSJ0

7128 training, 330 test utterances

83 speakers in train, 12 speaker in te

8 times the original data




## New features of data

Inter Channel Variation over large distance

Device characteristic Nullified

Precise distance sampling of speech w.r.t human speaker

# Baseline Systems

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_0$ | 20.61 | 22.49 | 31.66 | 39.25 | 44.38 | 46.84 | 43.93 | 38.8 |
| $Model_1$ | 22.59 | 21.86 | 27.7 | 34.58 | 30.68 | 39.66 | 41.6 | 32.93 |
| $Model_2$ | 44.22 | 30.1 | 26.55 | 32.71 | 30.14 | 34.48 | 35.66 | 30.15 |
| $Model_3$ | 59.03 | 38.2 | 28.3 | 29.91 | 26.03 | 35.34 | 35.92 | 30.67 |
| $Model_4$ | 71.44 | 45.84 | 29.24 | 31.78 | 26.85 | 35.63 | 34.88 | 30 |
| $Model_5$ | 65.05 | 39.98 | 29.7 | 29.91 | 27.4 | 32.18 | 36.95 | 31.5 |
| $Model_6$ | 71.9 | 46.5 | 30.23 | 29.91 | 26.58 | 37.07 | 32.82 | 30.52 |
| $Model_7$ | 45.34 | 36.17 | 29.87 | 31.78 | 27.95 | 36.21 | 32.56 | 29.76 |

Table 3.1: GMM speaker independent model

standard deviation 10.22

# Baseline Systems

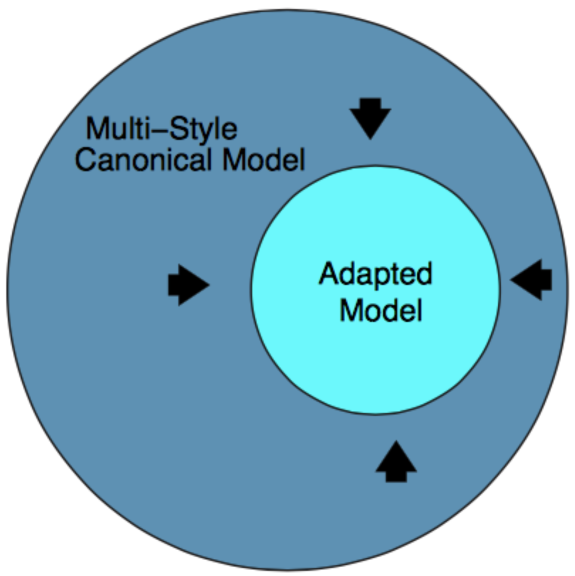
| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_0$ | 17.82 | 22.34 | 44.85 | 66.49 | 72.76 | 77.1 | 63.42 | 43.68 |
| $Model_1$ | 20.72 | 18.76 | 27.03 | 39.31 | 43.21 | 46.35 | 39.31 | 34.62 |
| $Model_2$ | 31.66 | 23 | 21.58 | 25.39 | 26.27 | 25.89 | 25.18 | 24.4 |
| $Model_3$ | 32.67 | 26.64 | 22.6 | 23.5 | 24.66 | 23.97 | 23.8 | 24.34 |
| $Model_4$ | 48.76 | 29.72 | 22.81 | 23.73 | 23.33 | 23.54 | 23.39 | 24.15 |
| $Model_5$ | 53.6 | 30.58 | 23.84 | 25.41 | 24.92 | 24.1 | 24.1 | 24.64 |
| $Model_6$ | 55.54 | 32.62 | 24.25 | 24.94 | 24.36 | 24.45 | 23.2 | 23.3 |
| $Model_7$ | 31.81 | 26.96 | 24.55 | 26.81 | 25.89 | 27.16 | 24.86 | 22.9 |

Table 3.3: DNN model after borrowing the clustering tree form $Model_0$

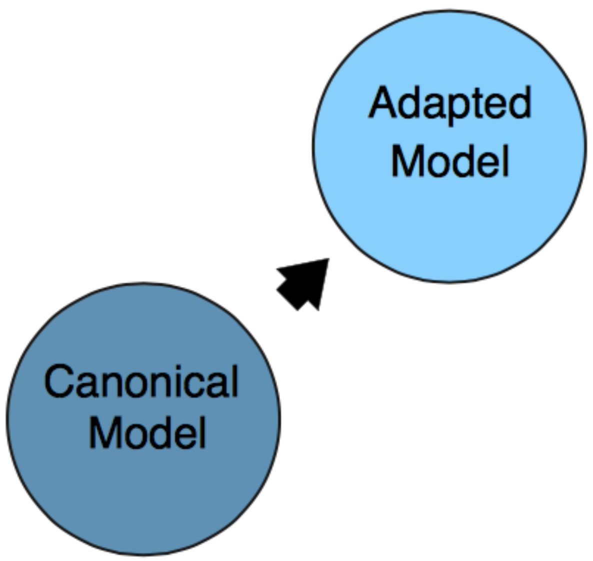
standard deviation 12.82

# Adaptation

Canonical Model Selection




(a) Multi-Style System




(b) Adaptive System

# Our Approach: Representational Mixing

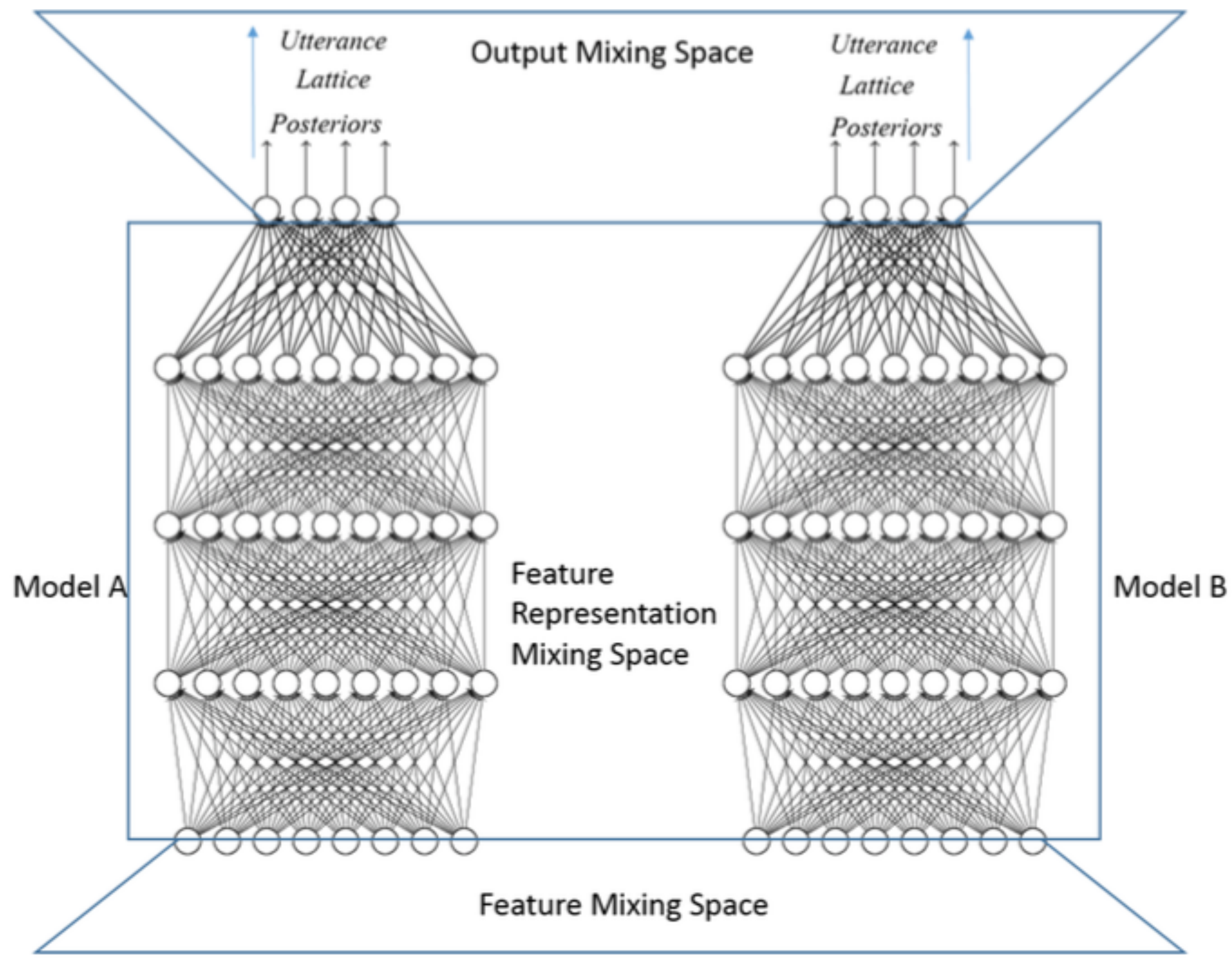
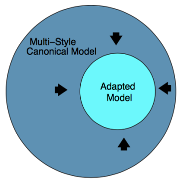



Figure 3.2: Level of abstraction for mixing the DNNs

$$Model_i(x) = \alpha Representation_A(x) + \beta Representation_B(x)$$

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_{40}$ | 19.39 | 22.70 | 23.00 | 24.36 | 24.01 | 24.53 | 24.27 | 24.75 |

Table 3.5: Model trained with $Data_0$ and $Data_4$ pooled together



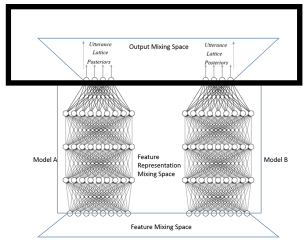
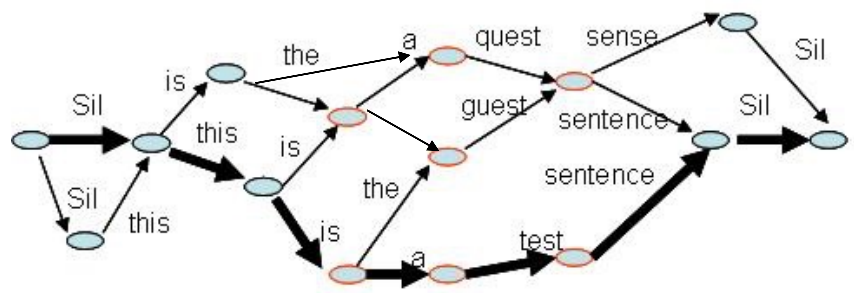
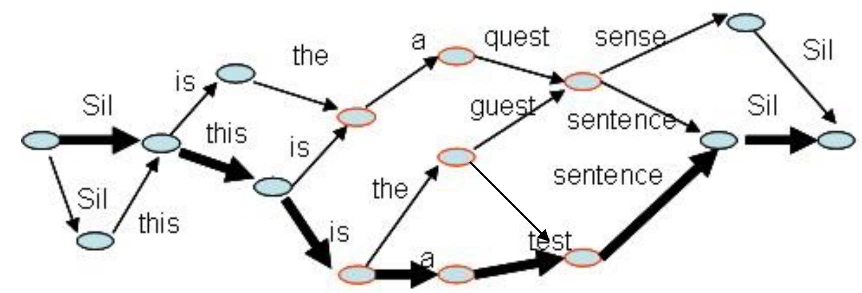




Figure 3.2: Level of abstraction for mixing the DNNs

# Lattice Mixing



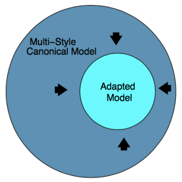

Links get re-weighted for better probability score

+ Only 1 parameter to estimate

+ No distance information required

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_{40}$ | 36.37 | 26.72 | 22.72 | 22.91 | 23.55 | 23.10 | 23.90 | 24.72 |
| $Model_{62}$ | 31.66 | 24.64 | 22.01 | 24.01 | 24.83 | 24.81 | 23.48 | 23.41 |

Table 3.4: Lattice Interpolation



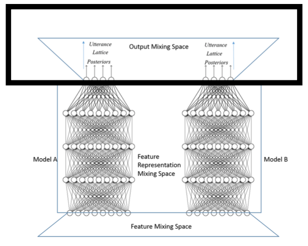




Figure 3.2: Level of abstraction for mixing the DNNs

# Utterance/Frame Oracle

Select best shot text/posterior per utterance/frame

+ No parameters, No distance information required

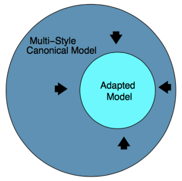
- Realtime pseudo-transcript required from another canonical model

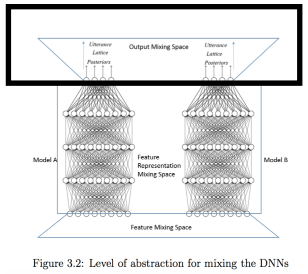
| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_{40}$ | 23.33 | 24.62 | 26.74 | 26.90 | 25.75 | 27.16 | 27.40 | 25.03 |

Table 3.6: Model selecting decoded utterance from $Model_0$ and $Model_4$

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_{40}$ | 26.88 | 25.65 | 32.86 | 28.38 | 33.44 | 29.37 | 34.34 | 29.18 |

Table 3.7: Model selecting posterior per frame from $Model_0$ and $Model_4$



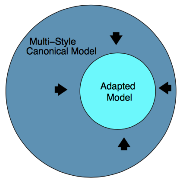

# Product of Experts

Instead of selecting the posteriors learn weights to interpolate the posteriors unseen condition

+ One parameters per expert, No distance information required

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_{23}$ | 29.37 | 23.80 | 23.22 | 24.23 | 25.59 | 24.51 | 23.93 | 24.95 |
| $Model_{40}$ | 33.21 | 26.51 | 42.67 | 43.74 | 44.29 | 44.3 | 43.32 | 39.72 |

Table 3.8: Product of Experts



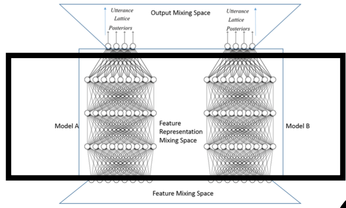




Figure 3.2: Level of abstraction for mixing the DNNs

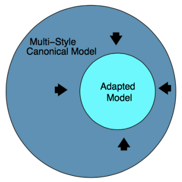
# Cluster Adaptive Training

- CAT or Multi-basis training is motivated by the representation learning capabilities of DNN.

$$z_x^L = W^L \left( \sum_{k=1}^{K} \lambda_k h_k^L(x) \right) + b^L = W^L H(x) \lambda + b^L$$

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_{40}$ | 19.45 | 23.12 | 23.70 | 25.36 | 25.32 | 23.11 | 25.32 | 24.98 |

Table 3.9: Cluster Adaptive Training using $Model_0$ and $Model_4$ as basis



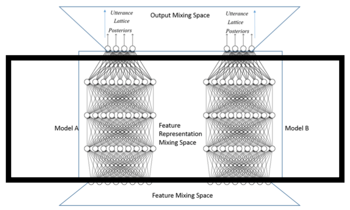




Figure 3.2: Level of abstraction for mixing the DNNs

# Multitask Training

- Instead of specifying the explicit nature of mixing of representation, the mixing is controlled by the secondary task

$$J_{Multitask}(W,b) = J_{Primarytask}(W,b) + \lambda J_{Secondarytask}(W,b)$$

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_{40}$ | 21.93 | 22.45 | 22.55 | 22.75 | 22.16 | 23.02 | 22.40 | 22.72 |

Table 3.10: Multitask Training

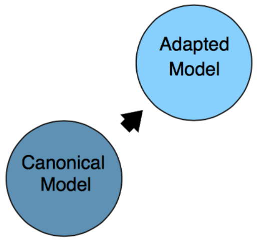
# Analysis: Representational Mixing

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_{40}$ | 19.39 | 22.70 | 23.00 | 24.36 | 24.01 | 24.53 | 24.27 | 24.75 |

Table 3.5: Model trained with $Data_0$ and $Data_4$ pooled together

| | | |
|---|---|---|
| $Data_0$ and $Data_4$ | 23.38 | |
| Lattice interpolation | $Model_{40}$ 25.5 | $Model_{62}$ 24.86 |
| Frame selection | $Model_{40}$ 30 | |
| Utterance selection | $Model_{40}$ 25.86 | |
| PoE | $Model_{40}$ 39.72 | $Model_{23}$ 24.95 |
| CAT | $Model_{40}$ 24.39 | |
| Multitask learning | $Model_{40}$ 22.5 | |

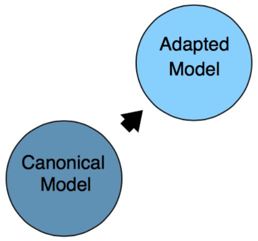
Multitask Learning: Variance 0.10




# Feature Space Normalisation

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_0$ | 17.82 | 19.88 | 26.60 | 31.81 | 32.36 | 32.36 | 29.52 | 28.10 |
| $Model_1$ | 18.57 | 18.76 | 23.30 | 28.47 | 29.11 | 27.98 | 26.94 | 26.17 |
| $Model_2$ | 21.22 | 20.85 | 21.58 | 24.30 | 25.16 | 25.16 | 24.40 | 23.30 |
| $Model_3$ | 22.08 | 21.99 | 22.64 | 23.50 | 24.42 | 24.06 | 23.39 | 23.91 |
| $Model_4$ | 22.42 | 21.78 | 22.32 | 23.22 | 23.33 | 23.58 | 22.73 | 23.69 |
| $Model_5$ | 23.78 | 23.50 | 23.65 | 24.72 | 24.88 | 24.10 | 24.19 | 24.47 |
| $Model_6$ | 22.38 | 22.55 | 23.48 | 24.19 | 23.78 | 24.64 | 23.20 | 23.31 |
| $Model_7$ | 21.97 | 22.83 | 23.67 | 25.52 | 25.01 | 25.46 | 24.04 | 22.90 |

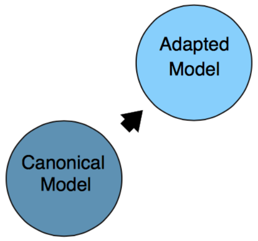
Table 3.11: DNN after applying the exact transform from correct Relative Position




# Feature Space Normalisation

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_0$ | 18.64 | 20.31 | 28.13 | 31.87 | 32.17 | 33.64 | 31.40 | 28.97 |
| $Model_1$ | 18.57 | 20.27 | 25.44 | 29.20 | 31.29 | 30.92 | 29.82 | 28.13 |
| $Model_2$ | 23.03 | 22.68 | 24.72 | 26.86 | 28.19 | 28.02 | 28.12 | 27.07 |
| $Model_3$ | 24.23 | 24.30 | 25.31 | 27.33 | 27.87 | 28.02 | 28.26 | 27.65 |
| $Model_4$ | 25.18 | 25.50 | 25.67 | 27.78 | 27.54 | 28.53 | 28.81 | 27.57 |
| $Model_5$ | 26.06 | 26.45 | 27.74 | 28.23 | 29.25 | 28.94 | 29.01 | 28.73 |
| $Model_6$ | 25.46 | 25.76 | 26.43 | 27.89 | 28.26 | 28.30 | 28.15 | 27.22 |
| $Model_7$ | 23.59 | 24.08 | 26.23 | 28.13 | 28.58 | 28.69 | 28.94 | 26.27 |

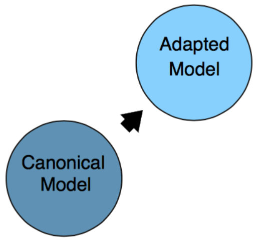
Table 3.12: DNN after applying the estimated transform per utterance




# Feature Space Normalisation

| DNN | $Data_0$ | $Data_1$ | $Data_2$ | $Data_3$ | $Data_4$ | $Data_5$ | $Data_6$ | $Data_7$ |
|---|---|---|---|---|---|---|---|---|
| $Model_0$ | 18.18 | 19.82 | 26.99 | 30.54 | 30.36 | 31.35 | 29.83 | 27.46 |
| $Model_1$ | 18.23 | 19.32 | 23.25 | 27.89 | 29.45 | 28.64 | 27.6 | 27.22 |
| $Model_2$ | 22.21 | 21.54 | 24.02 | 24.73 | 25.45 | 25.12 | 24.87 | 24.13 |
| $Model_3$ | 22.45 | 22.33 | 23.42 | 24.64 | 24.26 | 24.72 | 24.46 | 24.17 |
| $Model_4$ | 23.37 | 23.15 | 23.74 | 24.42 | 24.55 | 23.97 | 24.58 | 24.86 |
| $Model_5$ | 23.7 | 23.34 | 24.61 | 24.59 | 24.78 | 24.18 | 25.06 | 24.38 |
| $Model_6$ | 23.32 | 22.89 | 24.54 | 23.91 | 24.39 | 24.85 | 24.56 | 24.25 |
| $Model_7$ | 22.19 | 21.63 | 24.38 | 24.35 | 24.48 | 24.62 | 24.14 | 23.97 |

Table 3.13: DNN after trained on per utterance CMVN




# Summary: Feature Space Normalisation

Table 3.14: Summary of the Results

| Technique | WER | Variance | Min WER |
|---|---|---|---|
| Global CMVN(In Train & Test) | 30.89 | 12.82 | 17.82 |
| Global CMVN(Train) & Known Stats(Test) | 24.14 | 8.01 | 17.82 |
| Global CMVN(Train) & Per Utterance(Test) | 27.08 | 8.43 | 18.57 |
| Per Utterance(Train & Test) | 24.5 | 6.43 | 18.18 |

Per Utterance Trained Model: Variance 6.43

# Conclusion

A new corpus

HMM-GMM systems vs the GMM-DNN systems

Demonstrate the difficulty in adapting to the channel mismatch

We introduce the Multitask learning and CAT as adaptation technique **(1% Absolute Improvement)**

We identify that the reason of degradation of performance in case of mismatch

We propose per utterance CMVN normalised training for better adaptation for channel **(6% Absolute Improvement)**

# Future Work

Per-utterance normalisation solve the problem of inability of mixing

Improvement on the WER on unseen data by applying per-utterance CMVN normalisation, vs degradation on WER for seen data.

If we can model this as a Linear Input Network where the CMVN transform is learned and adapted in the case of mismatch.

Per frame fast adaptation and speaker tracking using model mixing.

Decreasing number of parameters of CAT DNN and improving its performance.

Considering reverberation (noise/speaker maybe on different data) and analysing joint effect on normalisation.

# Question?

Thank You