MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Recent Advances in Distant Speech Recognition

Delcroix, M.; Watanabe, S.

TR2016-115 September 2016

Abstract

Automatic speech recognition (ASR) is being deployed successfully more and more in products such as voice search applications for mobile devices. However, it remains challenging to perform recognition when the speaker is distant from the microphone, because of the presence of noise, attenuation, and reverberation. Research on distant ASR has received increased attention, and has progressed rapidly due to the emergence of 1) deep neural network (DNN) based ASR systems, 2) the launch of recent challenges such as CHiME series, REVERB, ASpIRE, and DIRHA, and 3) the development of new products such as the Microsoft Kinect and the AMAZON Echo. This tutorial will review the recent progresses made in the field of distant speech recognition in the DNN era, including single and multi-channel speech enhancement front-ends, and acoustic modeling techniques for robust back-ends. The tutorial will also introduce practical schemes for building distant ASR systems based on the expertise acquired from past challenges.

2016 Interspeech Tutorials

Interspeech 2016 tutorial:

Recent advances in distant speech recognition

Marc Delcroix
NTT Communication Science Laboratories

Shinji Watanabe
Mitsubishi Electric Research Laboratories (MERL)

Table of contents

List of abbreviations

ASR	Automatic Speech Recognition
AM	Acoustic Model
BF	Beamformer
BLSTM	Bidirectional LSTM
CMLLR	Constrained MLLR (equivalent to fMLLR)
CNN	Convolutional Neural Network
CE	Cross Entropy
DAE	Denoising Autoencoder
DNN	Deep Neural Network
DOC	Damped Oscillator Coefficients
DSR	Distant Speech Recognition
D&S	Delay and sum (Beamformer)
fDLR	Feature space Discriminative Linear Regression
fMLLR	Feature space MLLR (equivalent to CMLLR)
GCC-PHAT	Generalized Cross Correlation with Phase Transform
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IRM	Ideal Ratio Mask
KL	Kullback–Leibler (divergence/distance)
LCMV	Linear Constrained Minimum Variance
LDA	Linear Discriminant Analysis
LIN	Linear Input Network
LHN	Linear Hidden Network
LHUC	Learning Hidden Unit Contribution
LM	Language Model
LP	Linear Prediction
LSTM	Long Short-Term Memory (network)
MAP	Maximum A Posterior
MBR	Minimum Bayes Risk
MCWF	Multi-Channel Wiener Filter
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MLLT	Maximum Likelihood Linear Transformation
MMeDuSA	Modulation of Medium Duration Speech Amplitudes
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
MVDR	Minimum Variance Distortionless Response (Beamformer)
NMF	Non-negative Matrix Factorization
PNCC	Power-Normalized Cepstral Coefficients
RNN	Recurrent Neural Network
SE	Speech Enhancement
sMBR	state-level Minimum Bayes Risk
SNR	Signal-to-Noise Ratio
SRP-PHAT	Steered Response Power with the PHAse Transform
STFT	Short Time Fourier Transform
TDNN	Time Delayed Neural Network
TDOA	Time Difference Of Arrival
TF	Time-Frequency
VTLN	Vocal Tract Length Normalization
VTS	Vector Taylor Series
WER	Word Error Rate
WPE	Weighted Prediction Error (dereverberation)

Notations

Basic notation	
a	Scalar
$\mathbf{a}$	Vector
$\mathbf{A}$	Matrix
Signal processing	
A	Sequence
$x[n]$	Time domain signal at sample n
$X(t,f)$	Frequency domain coefficients at frame t and frequency bin f
ASR	
$\mathbf{o}_t$	Speech feature vector at frame t
$O \equiv \{\mathbf{o}_t \mid t = 1, \dots, T\}$	T-length sequence of speech features
w_n	Word at n^{th} position
$W \equiv \{w_n \mid n = 1, \dots, N\}$	N-length word sequence

Notations

operation	
a^*	Complex conjugate
$\mathbf{A}^{\mathrm{T}}$	Transpose
$\mathbf{A}^{\mathrm{H}}$	Hermitian transpose
$\mathbf{a} \circ \mathbf{b}$ or $\mathbf{A} \circ \mathbf{B}$	Elementwise multiplication
$\sigma()$	Sigmoid function
$\mathrm{softmax}()$	Softmax function
$\tanh()$	Tanh function

1. Introduction

1.1 Evolution of ASR

From pattern matching to probabilistic approaches

(Juang'04)

- 50s-60s
  - Initial attempts with template matching
  - Recognition of digits or few phonemes
- 70s
  - Recognition of 1000 words
  - First National projects (DARPA)
  - Introduction of beam search
- 80s
  - Introduction of probabilistic model approaches (**n-gram language models**, **GMM-HMM** acoustic models)
  - First attempts with Neural Networks
  - Launch of initial dictation systems (Dragon Speech)

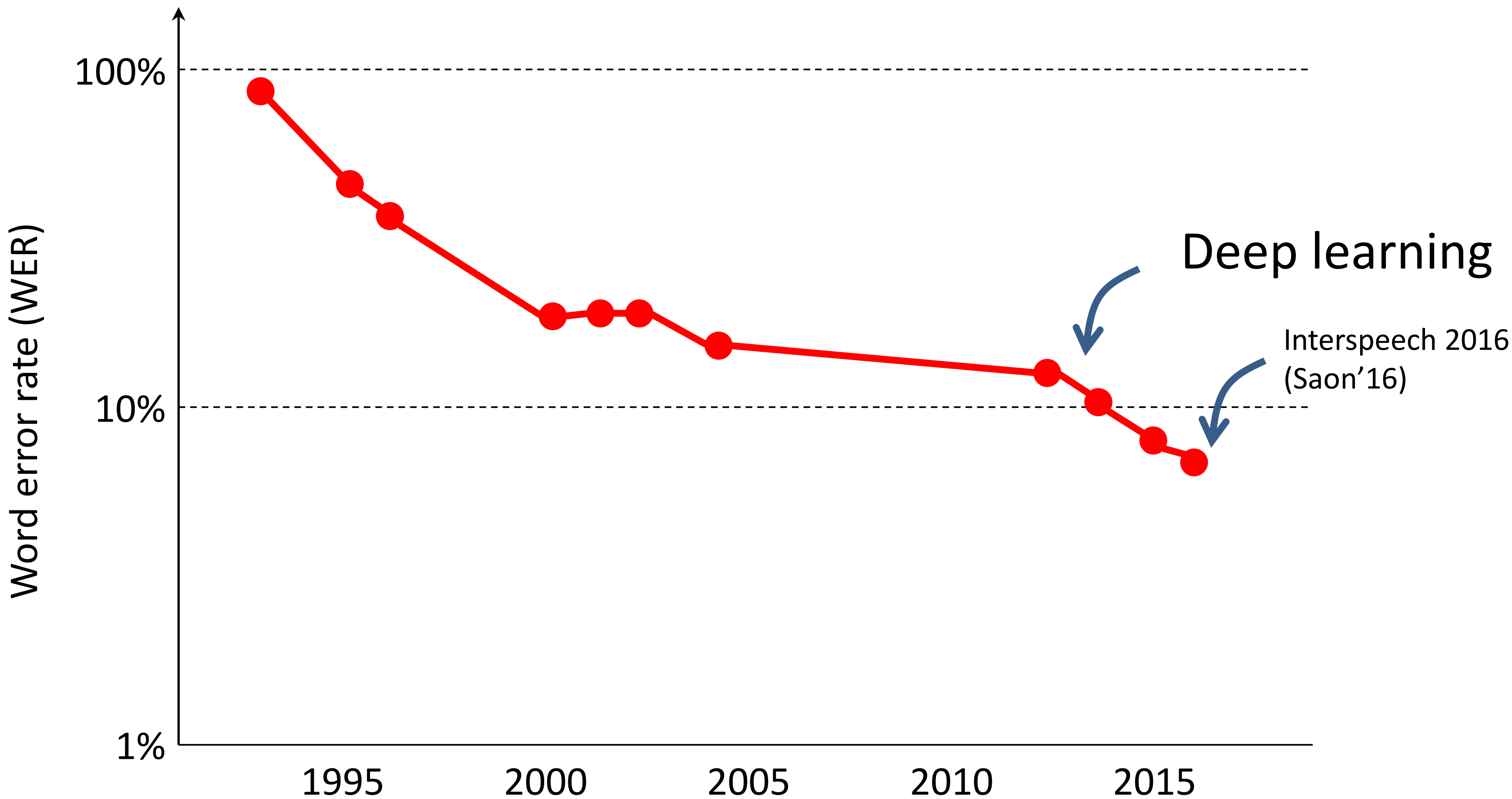
From research labs to outside world

(Juang'04)

- 90s
  - **Discriminative training** for acoustic models, **MLLR** adaptation, **VTS**
  - Development of Common toolkits (**HTK**)
- 2000s
  - Less breakthrough technologies
  - New popular toolkits such as **KALDI**
  - Launch of large scale applications (Google Voice search)
- 2010s
  - Introduction of **DNNs**, RNN-LMs
  - ASR used in more and more products (e.g. SIRI…)

Evolution of ASR performance

(Pallett'03, Saon'15, Saon'16)



Switchboard task (Telephone conversation speech)

Impact of deep learning

- Great performance improvement
  - DNNs are more robust to input variations

  → bring improvements for all tasks (LVCSR, DSR, …)

- Robustness is still an issue (Seltzer'14, Delcroix'13)
  - Speech enhancement/adaptation improve performance

    Microphone array, fMLLR, …

- Reshuffling the cards
  - Some technologies relying on GMMs became obsolete,

    VTS, MLLR …
  - Some technologies became less effective,

    VTLN, Single channel speech enhancement, …
  - New opportunities,
    - Exploring long context information for recognition/enhancement
    - Front-end/back-end joint optimization, …

Towards distant ASR (DSR)

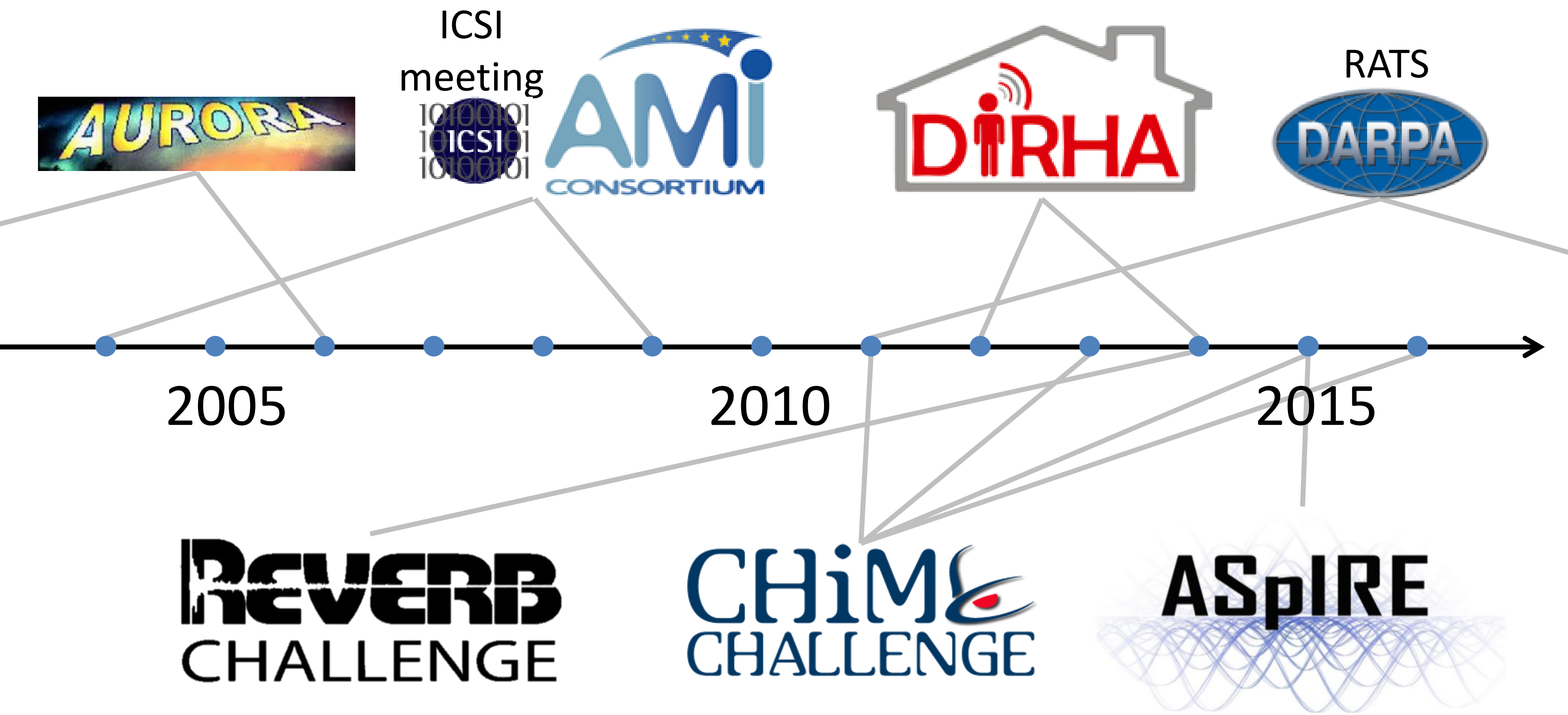
Close-talking microphone

e.g., voice search

Distant microphone

e.g., Human-human comm.,
Human-robot comm.

Interest for DSR - Academia



Interest for DSR - Industry

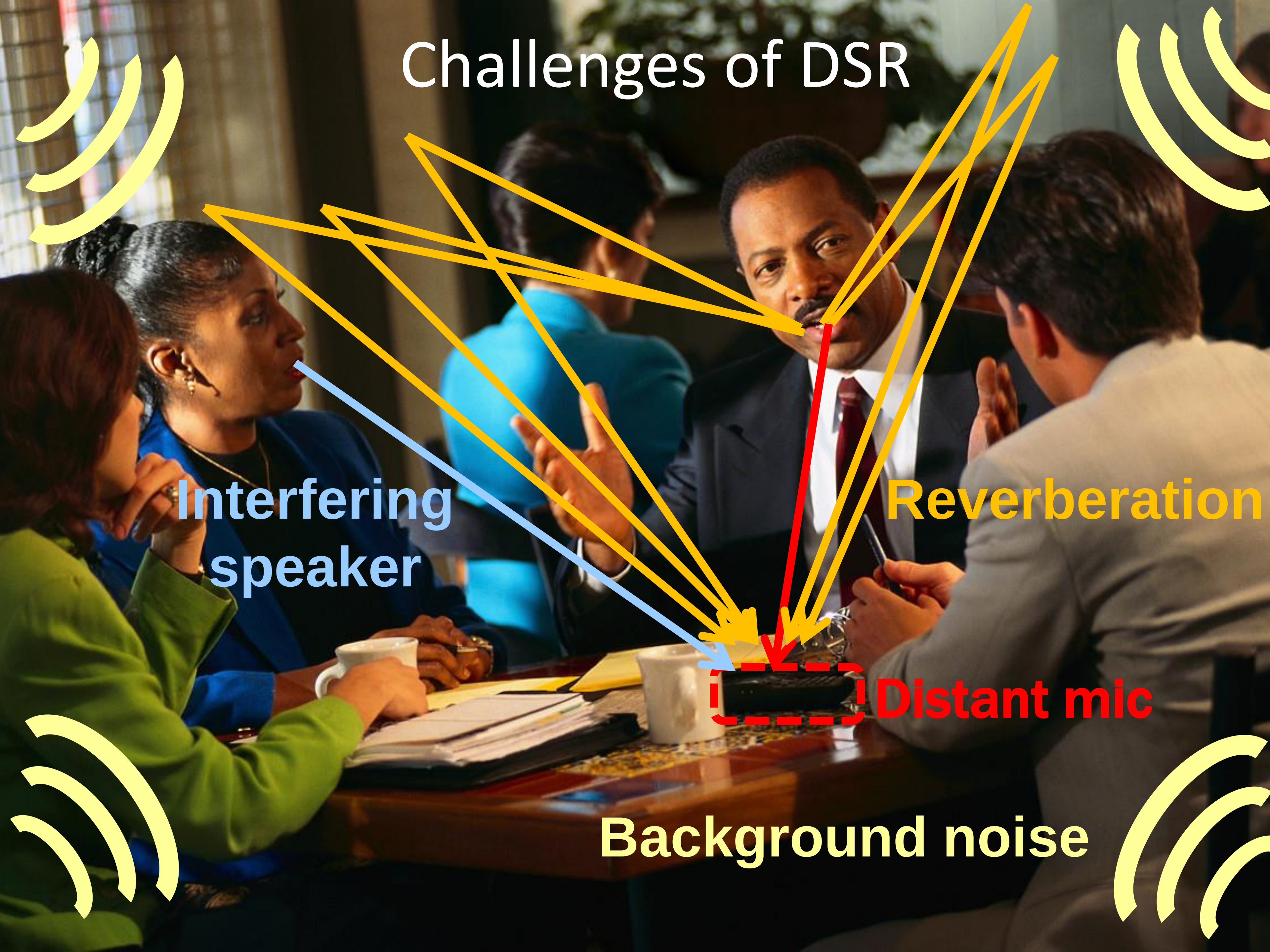
Home assistants

Game consoles

Robots

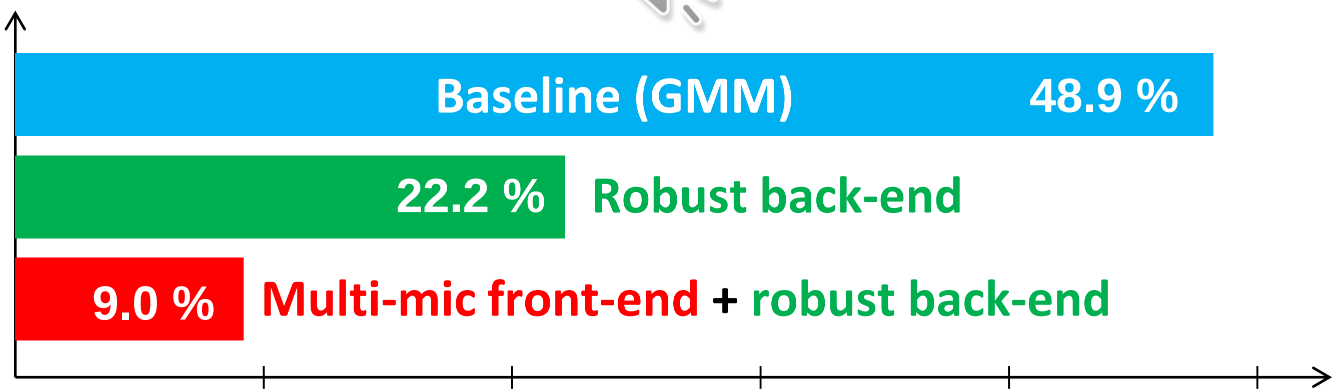
Voiced controlled appliances

1.2 Challenges of DSR

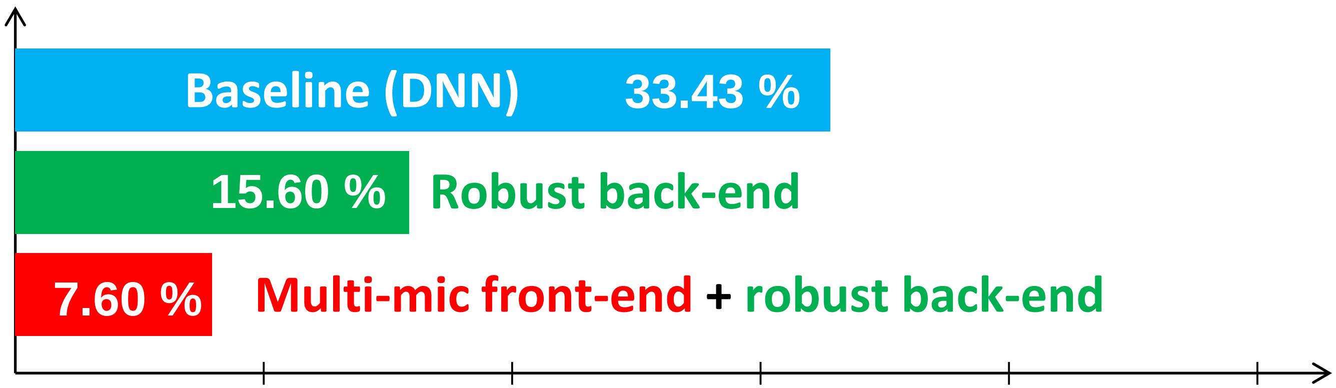

Challenges of DSR
Interfering speaker
Reverberation
Distant mic
Background noise

Recent achievements

- REVERB 2014 (WER)

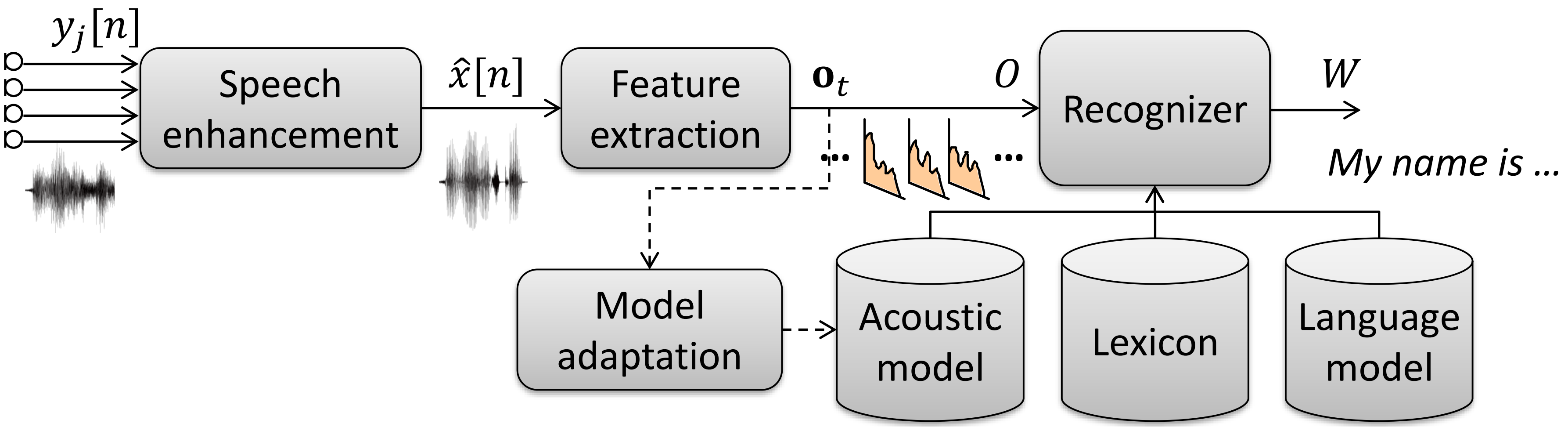


- CHiME-3 2015 (WER)



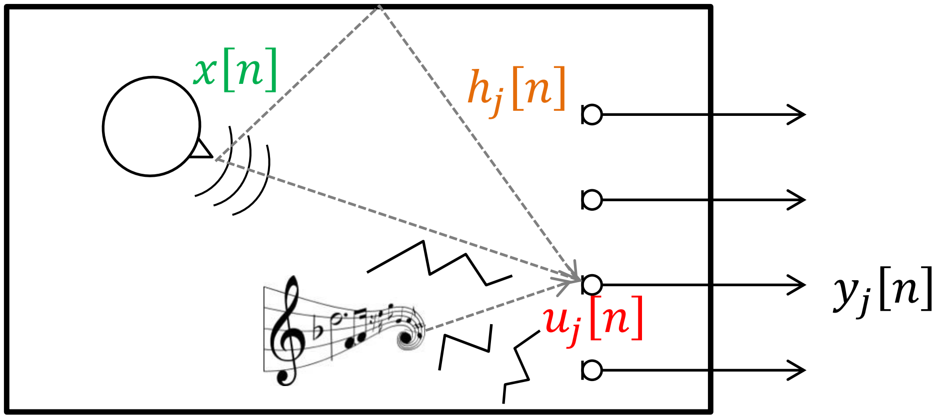
1.3 Overview of DSR systems

DSR system


$y_j[n]$
Speech enhancement
$\hat{x}[n]$
Feature extraction
$\mathbf{o}_t$
O
Recognizer
W
My name is …
Model adaptation
Acoustic model
Lexicon
Language model

Signal model – Time domain

- Speech captured with a distant microphone array



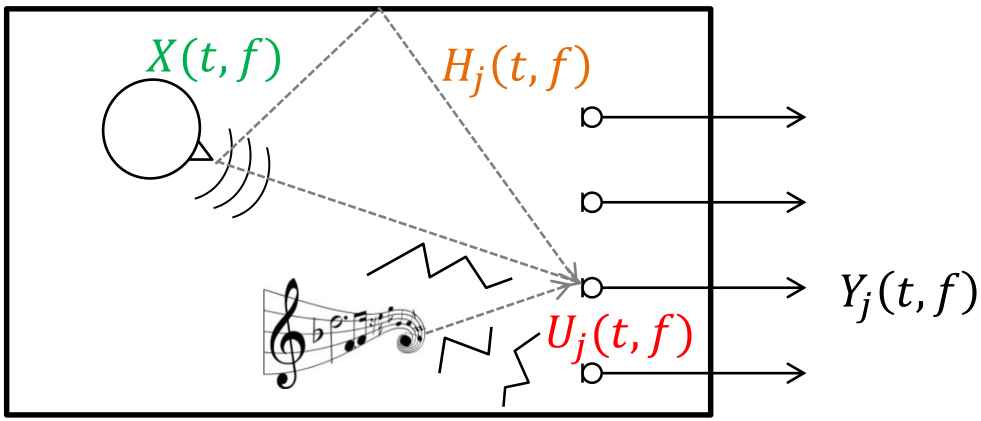
- Microphone signal at j^{th} microphone

$$y_j[n] = \sum_l h_j[l]x[n-l] + u_j[n] = h_j[n] * x[n] + u_j[n]$$

  - $x[n]$ Target clean speech
  - $h_j[n]$ Room impulse response
  - $u_j[n]$ Additive noise (background noise, …)
  - n Time index

Signal model - STFT domain

- Speech captured with a distant microphone array



- Microphone signal at j^{th} microphone:

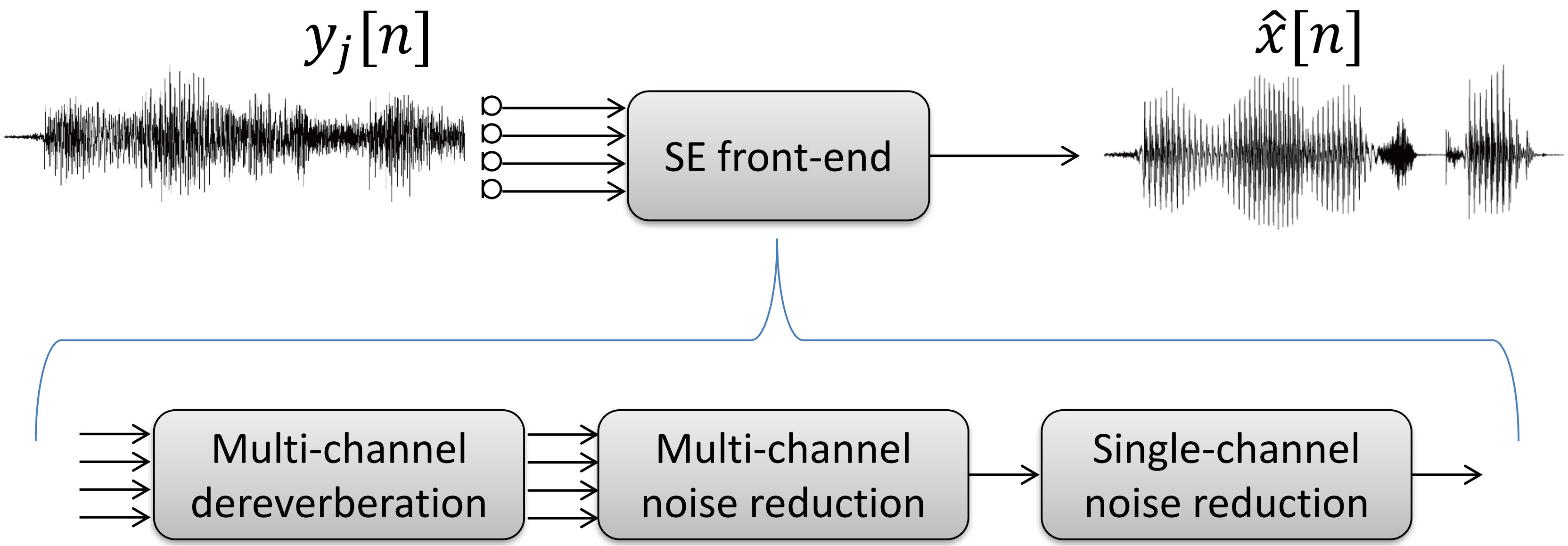
$$Y_j(t,f) \approx \sum_m H_j(m,f) X(t-m,f) + U_j(t,f) = H_j(t,f) * X(t,f) + U_j(t,f)$$

- $X(t,f)$ Target clean speech
- $H_j(t,f)$ Room impulse response
- $U_j(t,f)$ Additive noise
- (t,f) time frame index and frequency bin index

Approximate a long-term convolution in the time domain as a convolution in the STFT domain, because $h_i[n]$ is longer than the STFT analysis window

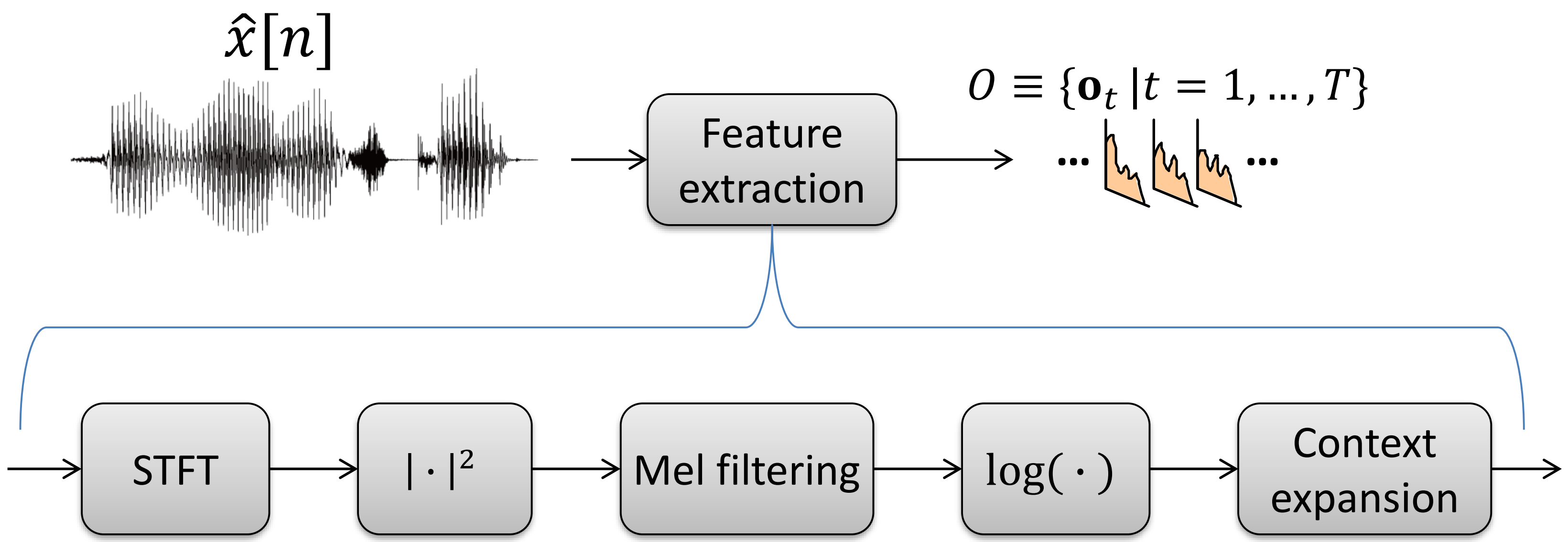
Speech enhancement (SE) front-end

- Reduce mismatch between the observed signal and the acoustic model caused by noise and reverberation



Feature extraction

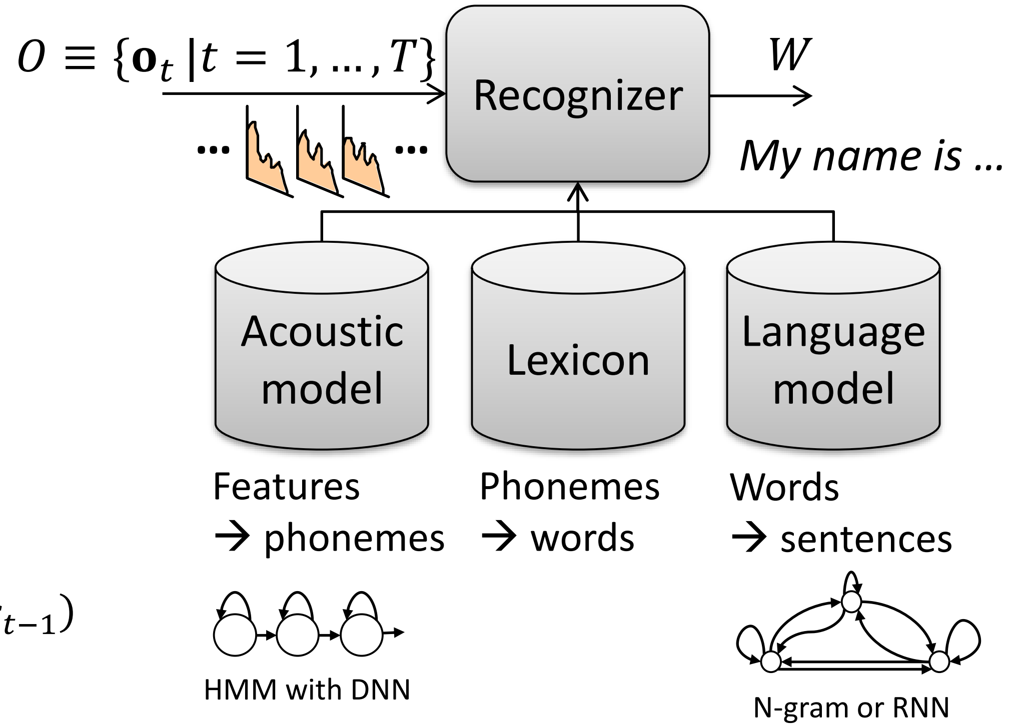
- Converts a speech signal to a sequence of speech features more suited for ASR, typically log mel filterbank coefficients
- Append left and right context



Recognition

- Speech recognition
  - Bayes decision theory(MAP):

$$\hat{W} = \arg\max_W p(W|O) = \arg\max_W p(O|W)p(W)$$



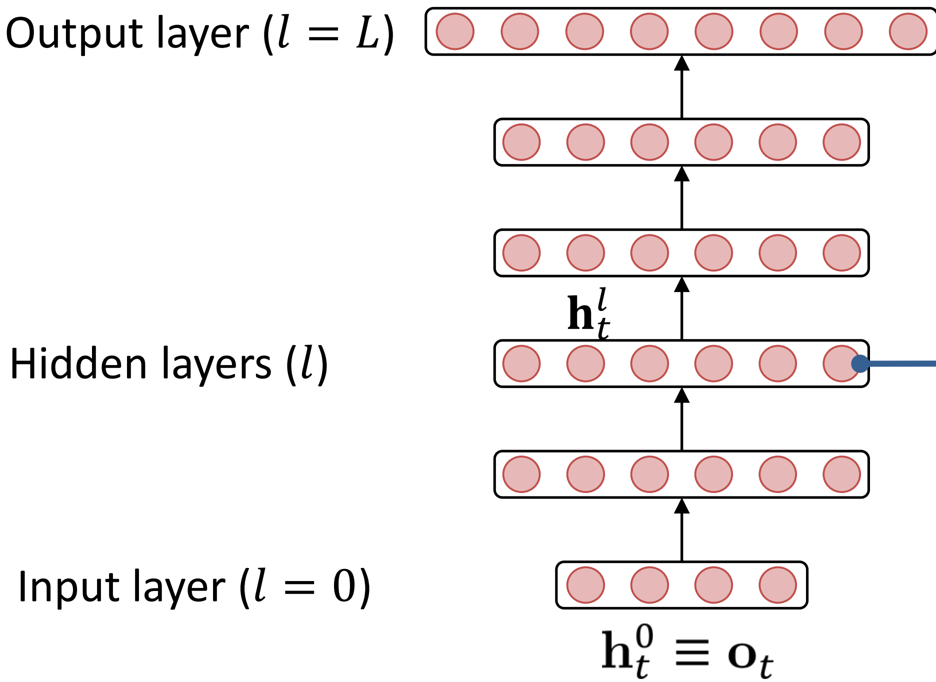
- Acoustic model
  - HMM:

$$p(O|S) = p(\boldsymbol{o}_1|s_1)p(s_1)\prod_{t=2}^{T} p(\boldsymbol{o}_t|s_t)p(s_t|s_{t-1})$$

Where s_t is an HMM state index

  - HMM state emission probability, $p(\boldsymbol{o}_t|s_t)$ obtained as the output of a deep neural network (DNN)

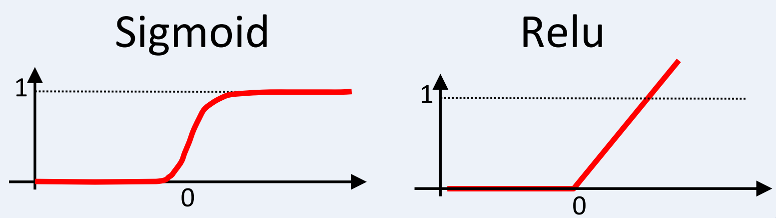
Basics of deep neural networks



$$p(s_t = k|\mathbf{o}_t) = h_{t,k}^L = [\mathrm{softmax}(\mathbf{a}_t^L)]_k$$

$$\mathbf{a}_t^l = \mathbf{W}^l \mathbf{h}_t^{l-1} + \mathbf{b}^l$$
$$\mathbf{h}_t^l = \sigma\left(\mathbf{a}_t^l\right)$$

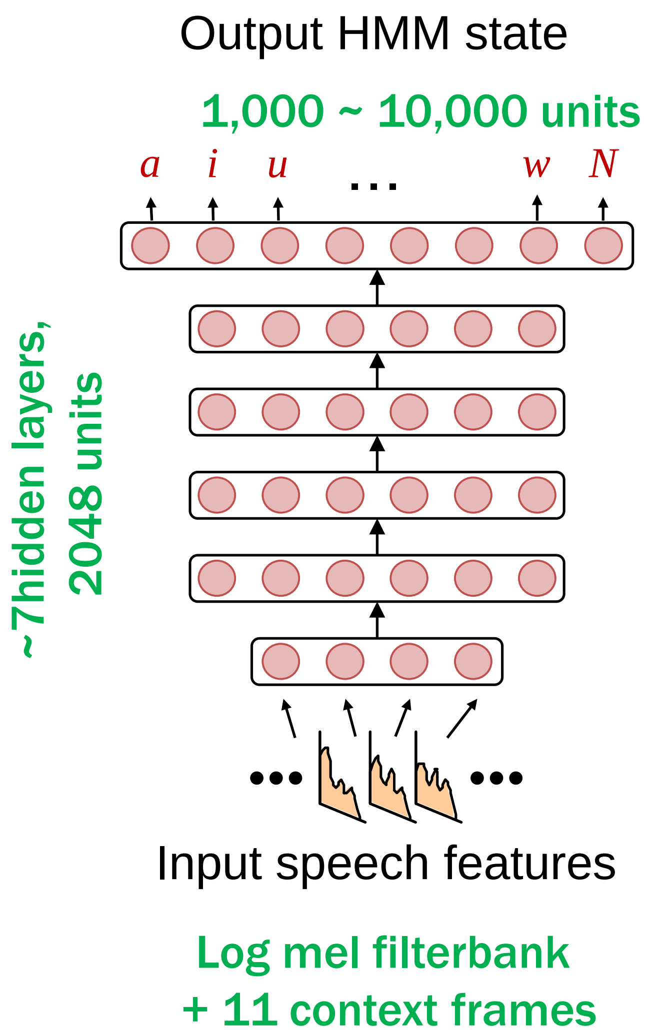
Activation function $\sigma(\cdot)$



- Trained using error back-propagation
- Training criterion, cross entropy, MMSE, State-level MBR, …

DNN-based acoustic modeling

(Hinton'12, Mohamed'12)

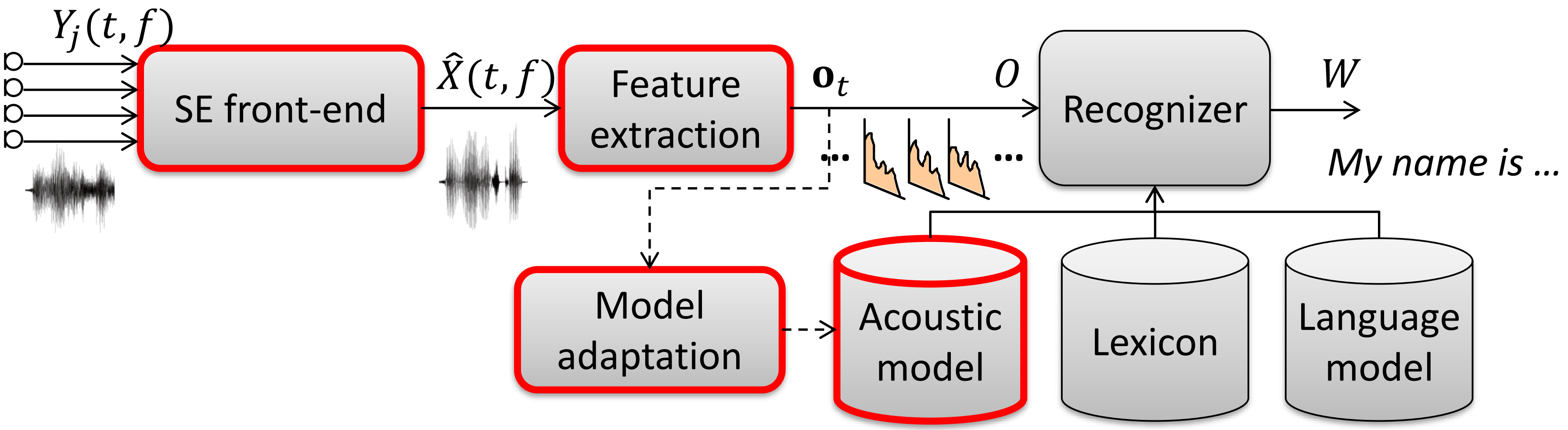


- Minimize cross entropy

$$J(\theta) = -\sum_{t}\sum_{k} \tau_{t,k} \log h_{t,k}^{L}(\theta)$$

  - $\tau_{t,k}$ Target label
  - $h_{t,k}^{L}$ Network output
  - θ Network parameters
- Optimization using error backpropagation
- Use large amount of speech training data with the associated HMM state alignments

Content of the tutorial



In this tutorial we describe some representative approaches for each of the main components of a DSR system

Topics not covered in this tutorial

- Voice activity detection
- Keyword spotting
- Multi-speaker / Speaker diarization
- Online processing
- Data simulation
- Lexicon, Language modeling and decoding

1.4 Overview of related tasks

Robust ASR tasks

CHiME CHALLENGE

REVERB CHALLENGE

AMI CONSORTIUM

AURORA

ASpIRE

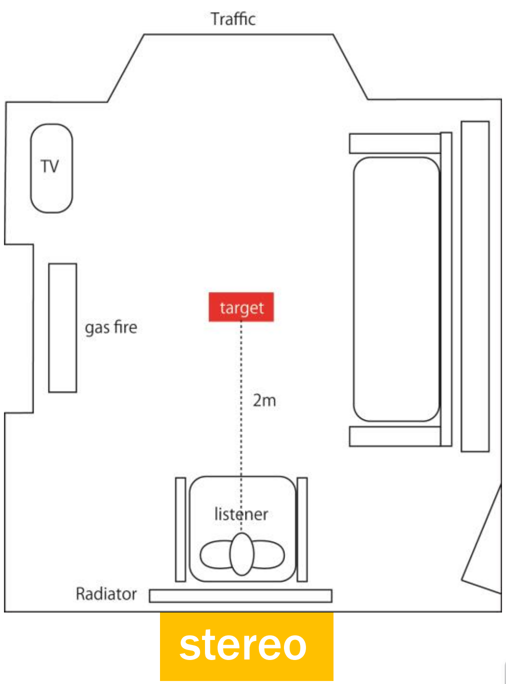
DIRHA

CHiME 1, 2

(Barker'13, Vincent'13)

- Distant speech recognition in living room
  - Acoustic conditions
    - Simulated distant speech
    - SNR: -6dB to - 9dB
  - # mics : 2
  - CHiME 1: Command (Grid corpus)
    + noise (living room)
  - CHiME 2 (WSJ): WSJ (5k) + noise (living room)

http://spandh.dcs.shef.ac.uk/chime_challenge



CHiME 3, 4

(Barker'15)

- Noisy speech recognition using a tablet
  - Recording conditions
    - Noise types: Bus, Café, Street, Pedestrian
    - # mics: 6 (CHiME3); 1, 2, 6 (CHiME4)
    - Simulated and real recordings
  - Speech
    - Read speech (WSJ (5k))

http://spandh.dcs.shef.ac.uk/chime_challenge

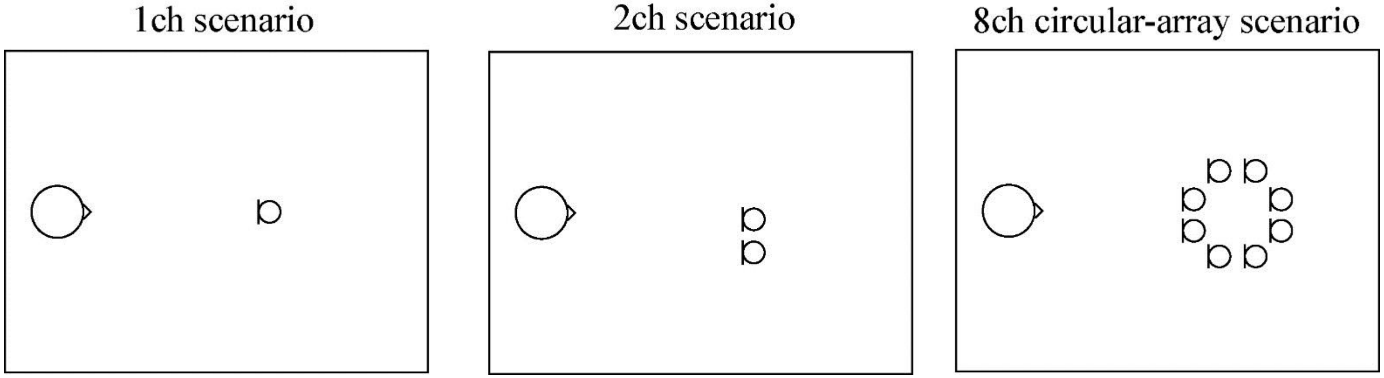
REVERB CHALLENGE

REVERB

(Kinoshita'13, Lincoln'05)

- Reverberant speech recognition
  - Recording conditions
    - Reverberation (RT 0.2 to 0.7 s.)
    - Noise type: stationary noise (SNR ∼20dB)
    - # mics: 1, 2, 8
    - Simulated and real recordings (MC-WSJ-AV)
  - Speech
    - Read speech (WSJ CAM0 (5k))

http://reverb2014.dereverberation.com



AMI

(Carletta'05)

- Meeting recognition corpus
  - Recording conditions
    - Multi-speaker conversations
    - Reverberant rooms
    - # mics: 8
    - Real recordings
  - Speech
    - Spontaneous meetings (8k)

http://corpus.amiproject.org/

AURORA

(Parihar'02)

- Aurora 4
  - Recording conditions
    - Noise types: car, babble, street, airport, train, restaurant
    - SNR: 5-15 dB
    - Channel distortion
    - # mics: 1
    - Simulation
  - Speech
    - Read speech (WSJ (5k))

http://aurora.hsnr.de/index-2.html

ASpIRE

(Harper'15)

- Large vocabulary reverberant speech
  - Recording conditions
    - Reverberant speech
    - 7 different rooms (classrooms and office rooms) with various shapes, sizes, surface properties, and noise sources
    - # mics: 1 or 6
  - Speech
    - Training data: Fisher corpus (2000 h of telephone speech)

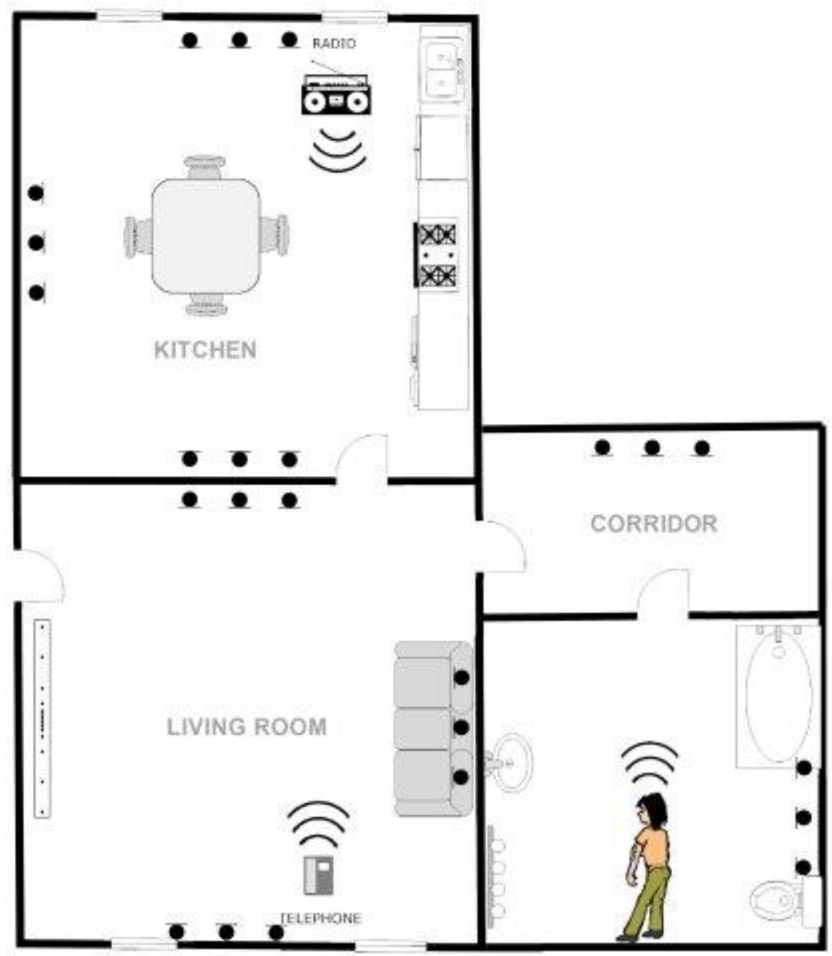
https://www.iarpa.gov/index.php/working-with-iarpa/prize-challenges/306-automatic-speech-in-reverberant-environments-aspire-challenge

DIRHA

(Matassoni'14)

- Multi-microphone and multi-language database
  - Acoustic conditions
    - Noise/reverberation recorded in an apartment
    - # mics: 40
    - Simulation
  - Speech
    - Multi-language (4 languages)
    - Various styles, command, keyword, spontaneous, …

http://dirha.fbk.eu/simcorpora



Summary of tasks

	Vocab size	Amount of training data	Real/ Simu	Type of distortions	# mics	Mic-speaker distance	Ground truth
ASpIRE	100K	~ 2000 h	Real	Reverberation	8/1	N/A	N/A
AMI	11K	75 h	Real	Multi-speaker conversations Reverberation and noise	8	N/A	Headset
Aurora4	5K	7,138 utt. (~ 14 h)	Simu	Additive noise + channel distortion (SNR 5-15dB)	1	N/A	Clean
CHiME1	50	17,000 utt.	Simu	Non-stationary noise recorded in a living room (SNR -6dB – 9dB) Reverberation from recorded impulse responses	2	2m	Clean
CHiME2 (WSJ)	5K	7138 utt. (~ 15 h)	Simu	Same as CHiME1	2	2m	Clean
CHiME3	5K	8738 utt. (~ 18 h)	Simu + Real	Non-stationary noise in 4 environments	6	0.5m	Close talk mic.
CHiME4	5K	8738 utt. (~ 18 h)	Simu + Real	Non-stationary noise in 4 environments	6/2/1	0.5m	Close talk mic.
REVERB	5K	7861 utt.. (~ 15 h)	Simu + Real	Reverberation in different living rooms (RT60 from 0.25 to 0.7 sec.) + stationary noise (SNR ~ 20dB)	8/2/1	0.5 m – 2m	Clean /Headset

References (Introduction)

(Barker’13) Barker, J. et al. “The PASCAL CHiME speech separation and recognition challenge,” Computer Speech&Language (2013).

(Barker’15) Barker, J. et al. “The third ’CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” Proc. ASRU (2015).

(Carletta’05) Carletta, J. et al. “The AMI meeting corpus: A pre-announcement,” Springer (2005).

(Delcroix’13) Delcroix, M. et al. “Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?” in Proc. Interspeech (2013).

(Harper’15) Haper, M. “The Automatic Speech recognition In Reverberant Environments (ASpIRE) challenge,” Proc. ASRU (2015).

(Hinton’12) Hinton, G., et al. “Deep neural networks for acoustic modeling in speech recognition,” IEEE Sig. Proc. Mag. 29, 82–97 (2012).

(Juang’04) Juang, B. H. et al. “Automatic Speech Recognition – A Brief History of the Technology Development,” (2004).

(Kinoshita’13) Kinoshita, K. et al. “The REVERB challenge: a common evaluation framework for dereverberation and recognition of reverberant speech,” Proc. WASPAA (2013).

(Kuttruff’09) Kuttruff, H. “Room Acoustics,” 5th ed. Taylor & Francis (2009).

(Lincoln’05) Lincoln, M. et al., “The multichannel wall street journal audio visual corpus (MC-WSJ-AV): Specification and initial experiments,” Proc. ASRU (2005).

(Matassoni’14) Matassoni, M. et al. “The DIRHA-GRID corpus: baseline and tools for multi-room distant speech recognition using distributed microphones,” Proc. Interspeech (2014).

(Mohamed’12) Mohamed, A. et al. “Acoustic modeling using deep belief networks,” IEEE Trans. ASLP (2012).

(Pallett’03) Pallett, D. S. “A look at NIST'S benchmark ASR tests: past, present, and future,” ASRU (2003).

(Parihar’02) Parihar, N. et al. “DSR front-end large vocabulary continuous speech recognition evaluation”, (2002).

(Saon’15) Saon, G. et al. “The IBM 2015 English Conversational Telephone Speech Recognition System,” arXiv:1505.05899 (2015).

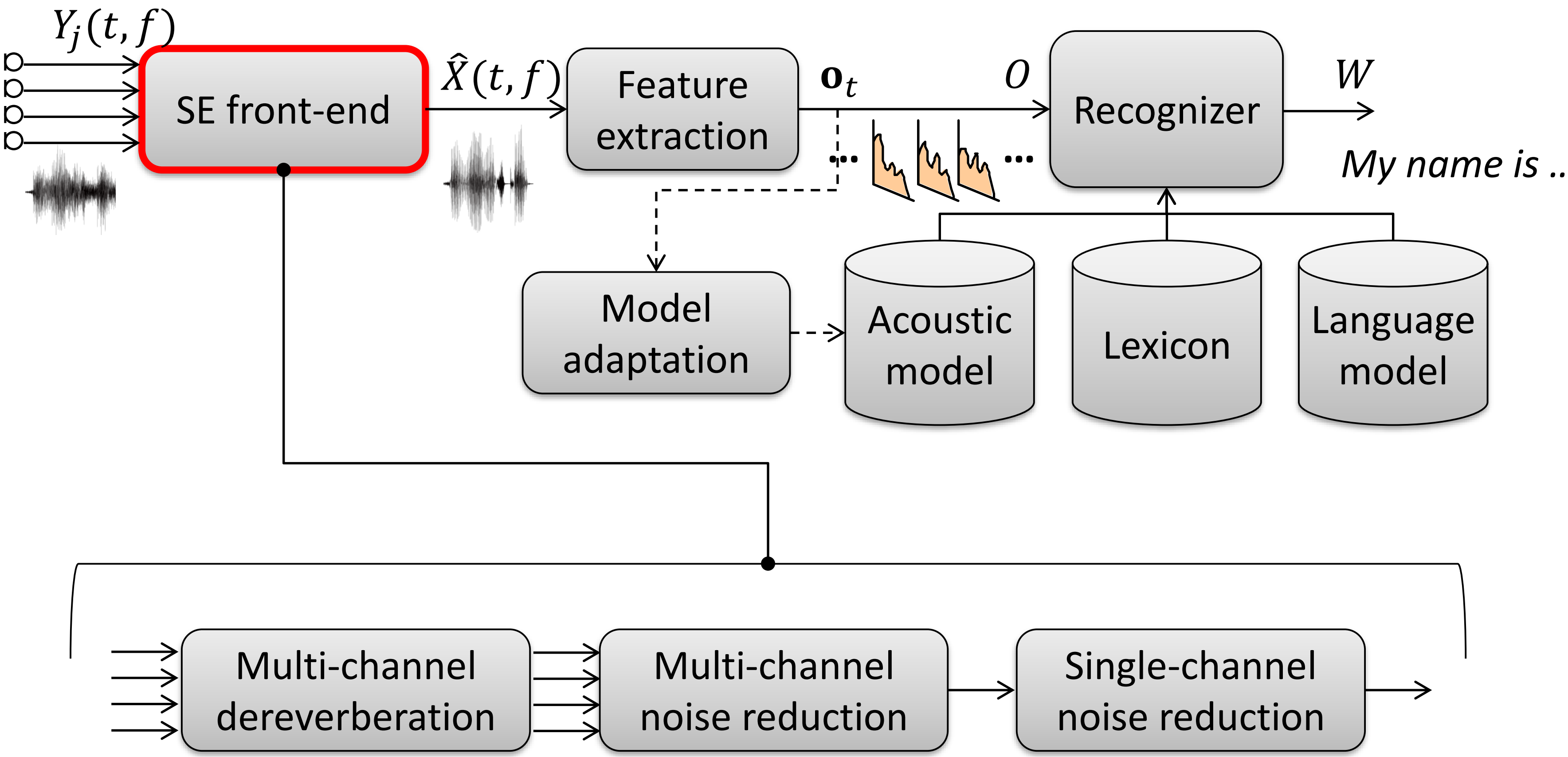
(Saon’16) Saon, G. et al. “The IBM 2016 English Conversational Telephone Speech Recognition System,” Proc. Interspeech (2016).

(Seltzer’14) Seltzer, M. “Robustness is dead! Long live Robustness!” Proc. REVERB (2014).

(Vincent’13) Vincent, E. et al. “The second ’CHiME’ speech separation and recognition challenge: Datasets, tasks and baselines,” Proc. ICASSP (2013).

2. Front-end techniques for distant ASR

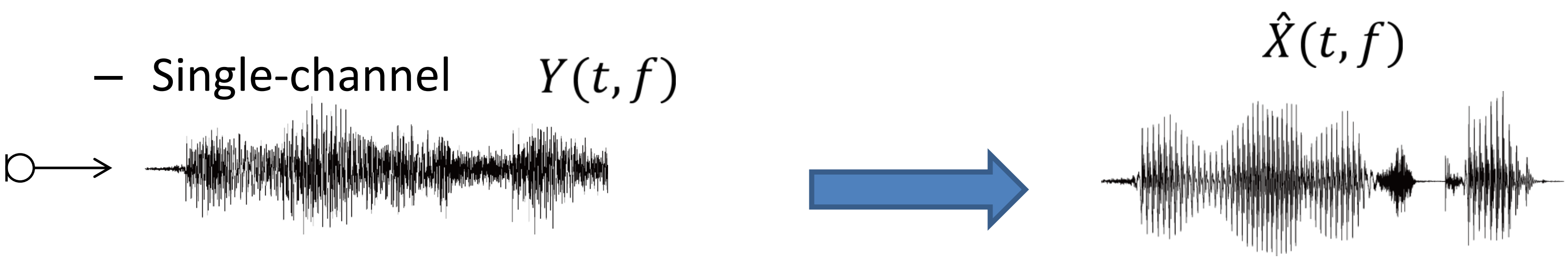
SE Front-end



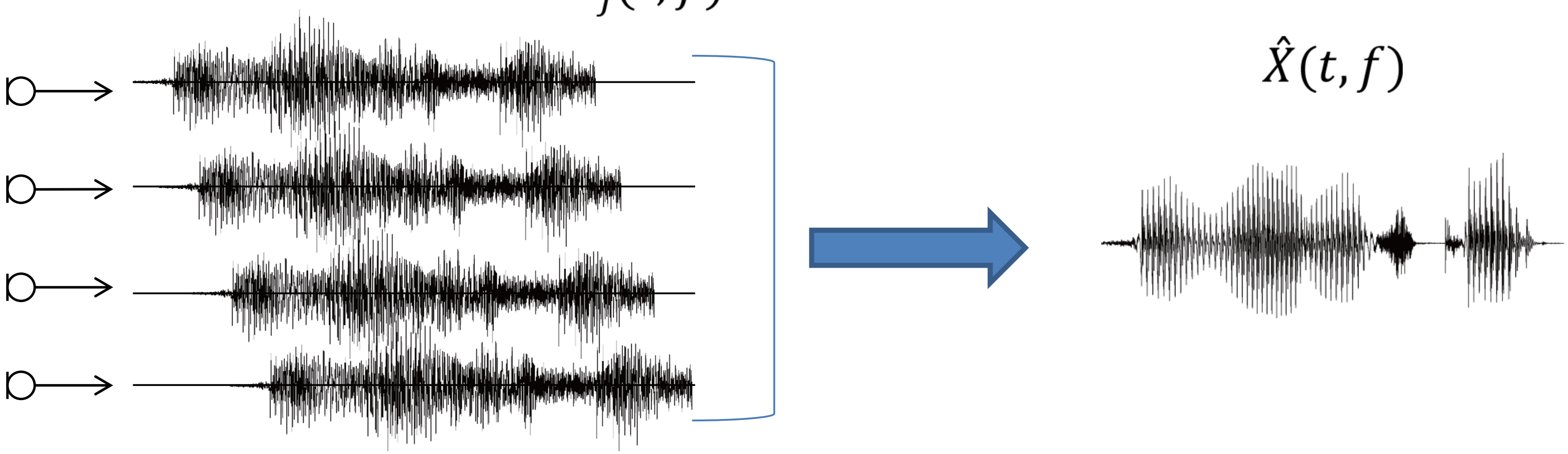
$Y_j(t,f)$
SE front-end
$\hat{X}(t,f)$
Feature extraction
$\mathbf{o}_t$
O
Recognizer
W
My name is …
Model adaptation
Acoustic model
Lexicon
Language model
Multi-channel dereverberation
Multi-channel noise reduction
Single-channel noise reduction

Speech enhancement (SE)

- Reduce mismatch between observed speech and ASR back-end due to noise/reverberation
  - Single-channel $Y(t,f)$



  - Multi-channel $Y_j(t,f)$



Type of processing

- Linear processing
  - Linear filter constant for long segments

$$Y(t,f) \longrightarrow \hat{X}(t,f) = W^*(f)Y(t,f)$$

- Non-linear processing
  - Linear filter changing for each time-frame

$$Y(t,f) \longrightarrow \hat{X}(t,f) = W^*(t,f)Y(t,f)$$

  - Non-linear transformation

$$Y(t,f) \longrightarrow \hat{X}(t,f) = F(Y(t,f))$$

With $F(\cdot)$ Non-linear function

Categorization of SE front-ends

	Single-channel	Multi-channel
Linear processing	• WPE dereverberation (Nakatani'10)	• Beamforming (Van Trees'02) • WPE dereverberation (Nakatani'10) • Neural network-based enhancement (Heymann'15)
Non-linear processing	• Spectral subtraction (Boll'79) • Wiener filter (Lim'79) • Time-frequency masking(Wang'06) • NMF (Virtanen'07) • Neural network-based enhancement (Xu'15, Narayanan'13, Weninger'15)	• Time-frequency masking (Sawada'04) • NMF (Ozerov'10) • Neural network-based enhancement (Xiao'16)

Categorization of SE front-ends

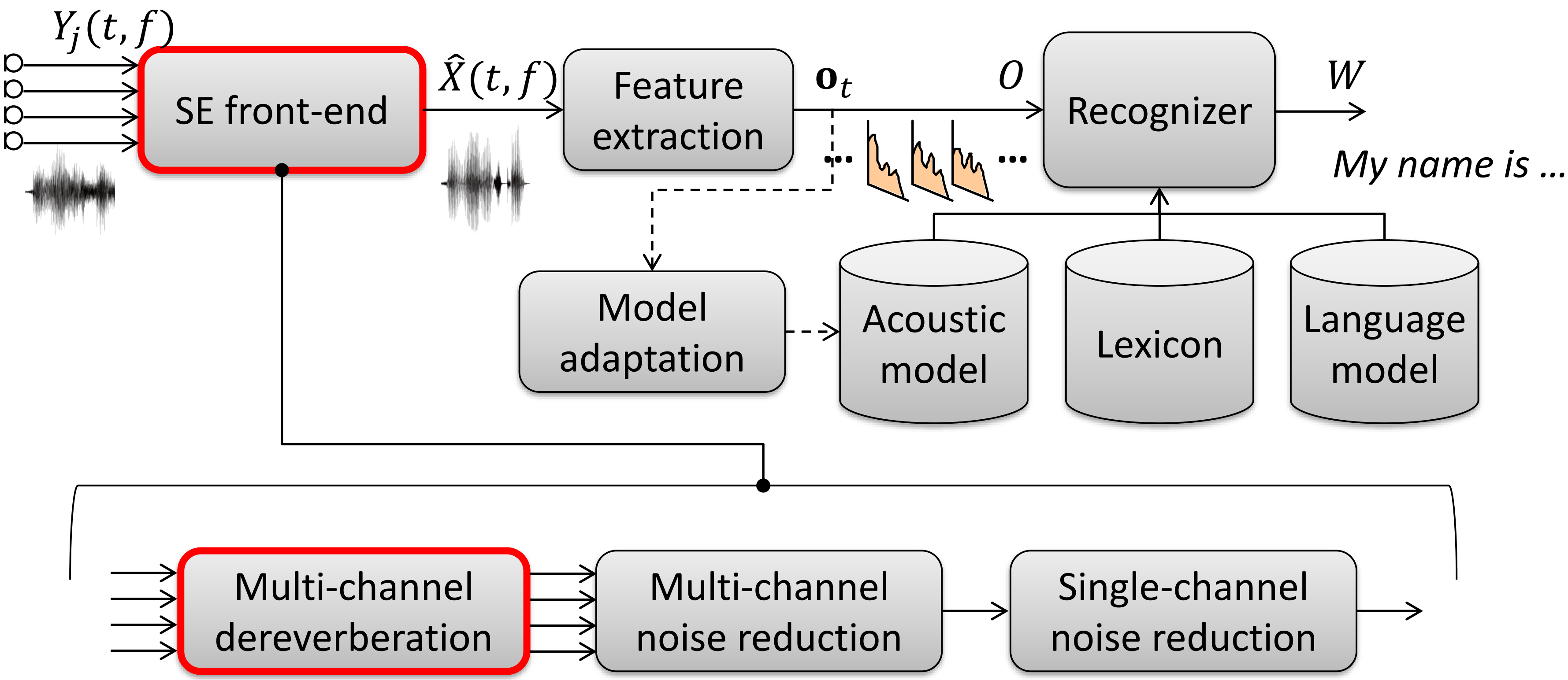
	Single-channel	Multi-channel
Linear processing	• **WPE dereverberation** (Nakatani'10)	• **Beamforming** (Van Trees'02) • **WPE dereverberation** (Nakatani'10) • **Neural network-based enhancement** (Heymann'15)
Non-linear processing	• Spectral subtraction (Boll'79) • Wiener filter (Lim'79) • Time-frequency masking(Wang'06) • NMF (Virtanen'07) • **Neural network-based enhancement** (Xu'15, Narayanan'13, Weninger'15)	• Time-frequency masking (Sawada'04) • NMF (Ozerov'10) • **Neural network-based enhancement** (Xiao'16)

Focus on

- Linear processing
- Neural network-based enhancement

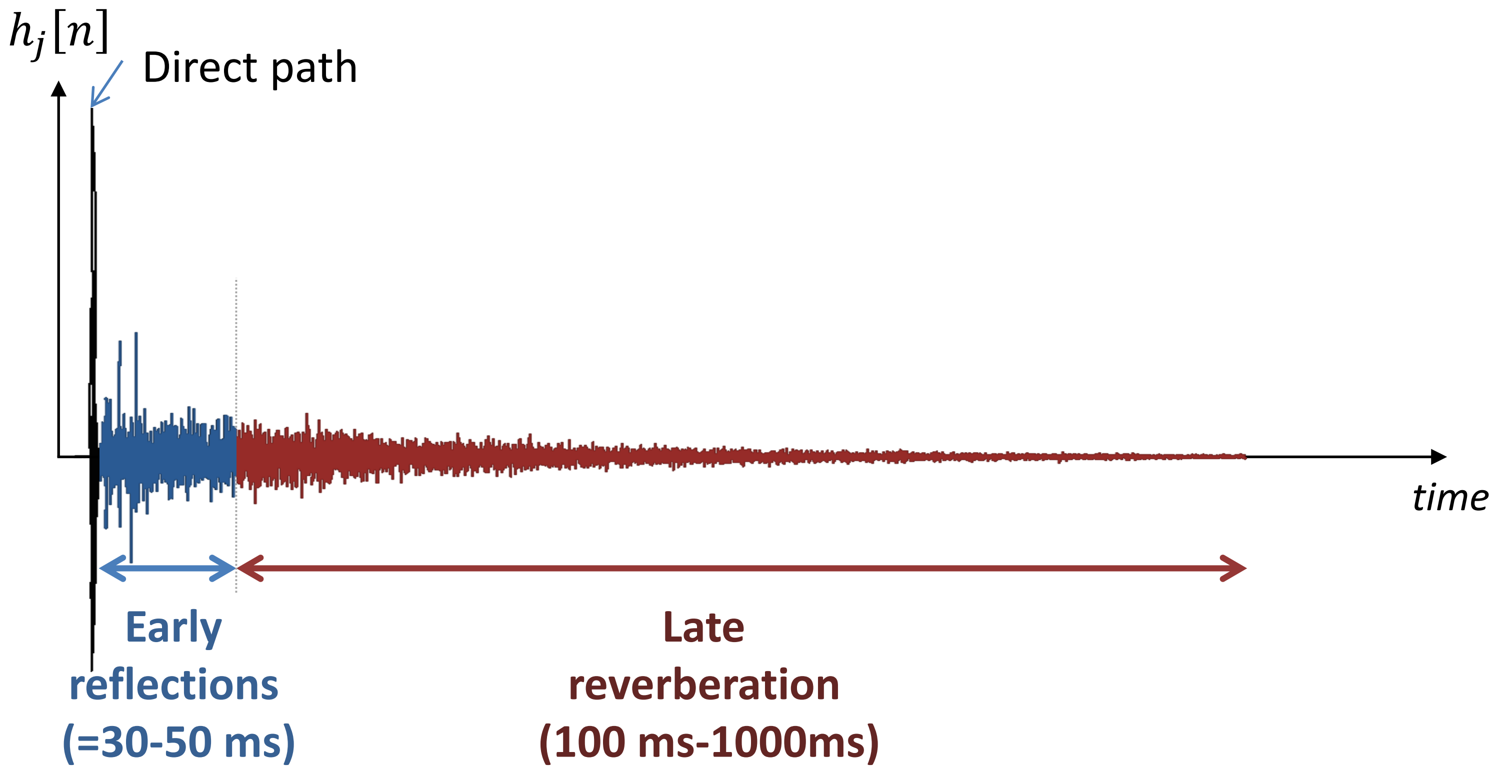
Have been shown to interconnect well with ASR back-end

2.1 Dereverberation



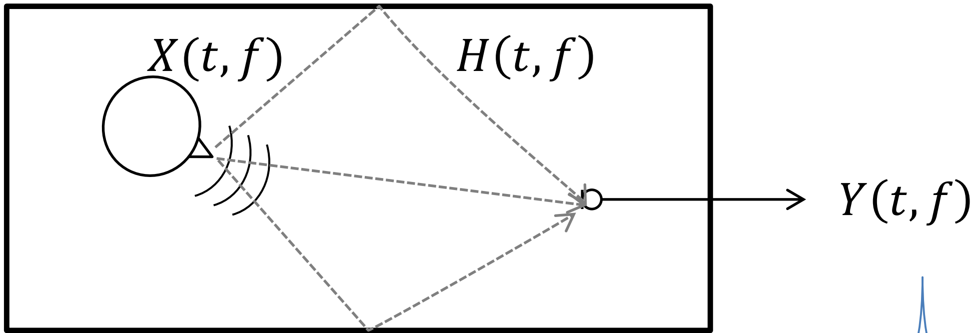
Room impulse response

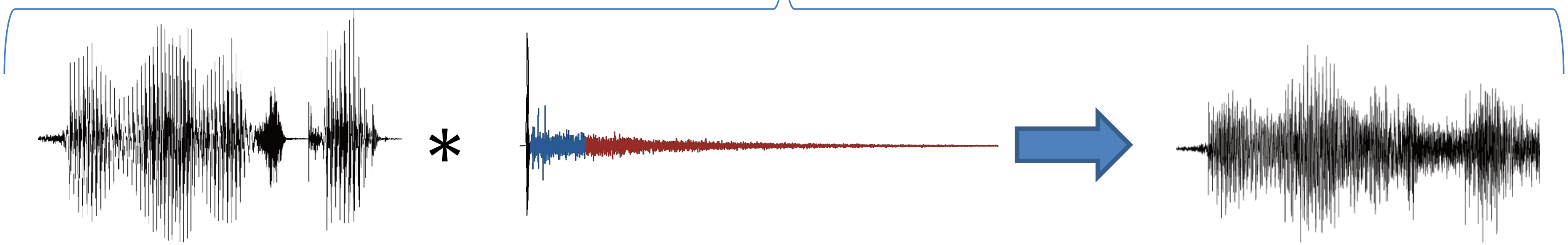
- Models the multi-path propagation of sound caused by reflections on walls and objects (Kuttruff'09)
  - Length 200-1000 ms in typical living rooms



Reverberant speech

(Yoshioka'12b)





$$Y(t,f) = H_j(t,f) * X(t,f) + U(t,f)$$
$$= \sum_{\tau=0}^{d} H(\tau,f)X(t-\tau,f) + \sum_{\tau=d+1}^{T} H(\tau,f)X(t-\tau,f) + U(t,f)$$

Direct sound + Early reflections $D(t,f)$

Late reverberation $L(t,f)$

Neglect noise for the derivations

Dereverberation aims at suppressing late reverberation

Dereverberation

- Linear filtering
  - Weighted prediction error

- Non-linear filtering
  - Spectral subtraction using a statistical model of late reverberation (Lebart'01, Tachioka'14)
  - Neural network-based dereverberation (Weninger'14)

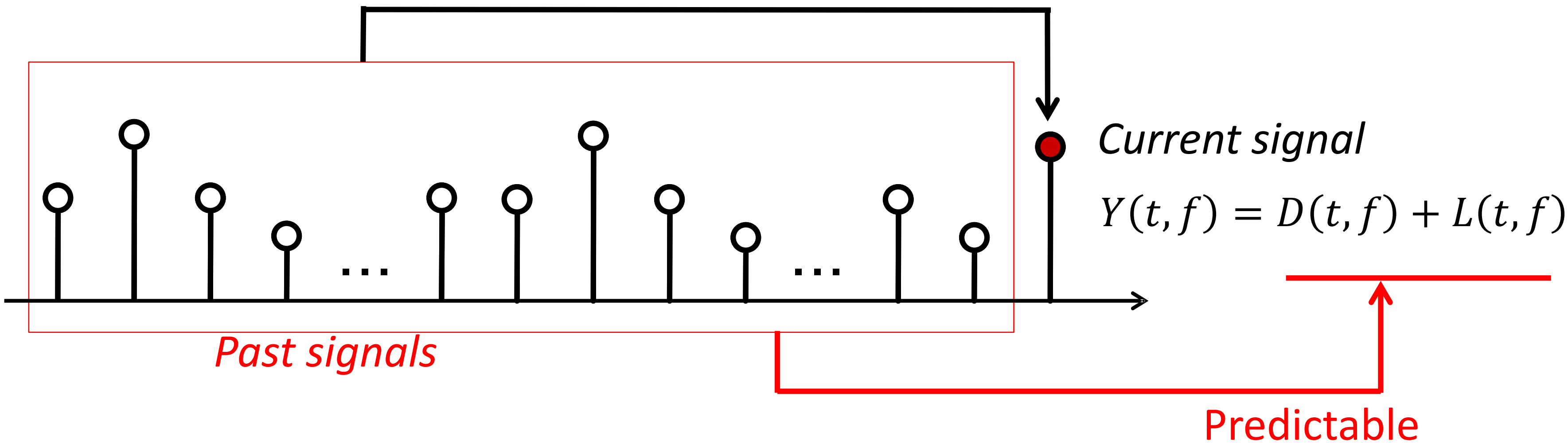
Linear prediction (LP)

(Haykin'96)

- Reverberation: linear filter

  → Can predict reverberation from past observations using linear prediction (*under some conditions*)

Prediction $\sum_{\tau=1} G^*(\tau, f) Y(t-\tau, f)$

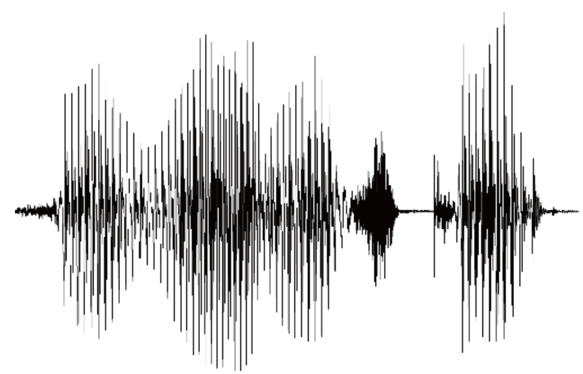


Dereverberation: $\hat{D}(t,f) = Y(t,f) - \sum_{\tau} G^*(\tau, f) Y(t-\tau, f)$

$D(t,f)$ and $L(t,f)$ are both reduced

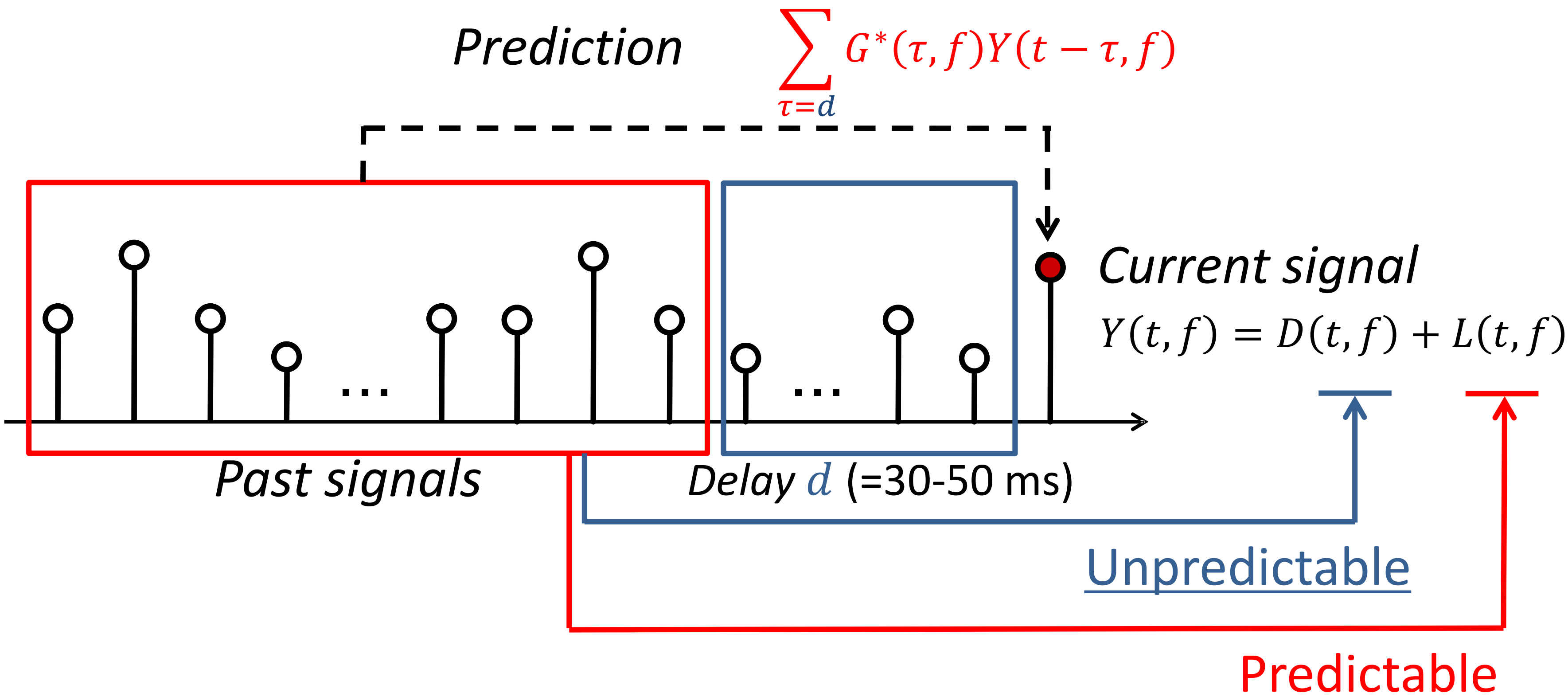
Problem of LP-based speech dereverberation

- LP predicts both early reflections and late reverberation
  - Speech signal exhibits short-term correlation (30-50 ms)
    → LP suppresses also the short-time correlation of speech
- LP assumes the target signal follows a stationary Gaussian distribution
  - Speech is not stationary Gaussian
    → LP destroys the time structure of speech
- Solutions:
  - Introduce a prediction delay (Kinoshita'07)
  - Introduce better modeling of speech signals (Nakatani'10, Yoshioka'12, Jukic'14)



Delayed linear prediction (LP)

(Kinoshita'07)



Delayed LP can only predict $L(t,f)$ from past signals

➡ Only reduce $L(t,f)$

Estimation of prediction coefficients

(Nakatani'10, Yoshioka'12)

Delayed LP: $$\hat{D}(t,f) = Y(t,f) - \sum_{\tau=d} G^*(\tau,f)Y(t-\tau,f)$$

- ML estimation for stationary signal

$$\{\hat{G}(\tau,f)\} = \underset{\{G(\tau,f)\}}{\operatorname{argmin}} \sum_t \left\| Y(t,f) - \sum_{\tau=d} G^*(\tau,f)Y(t-\tau,f) \right\|^2$$

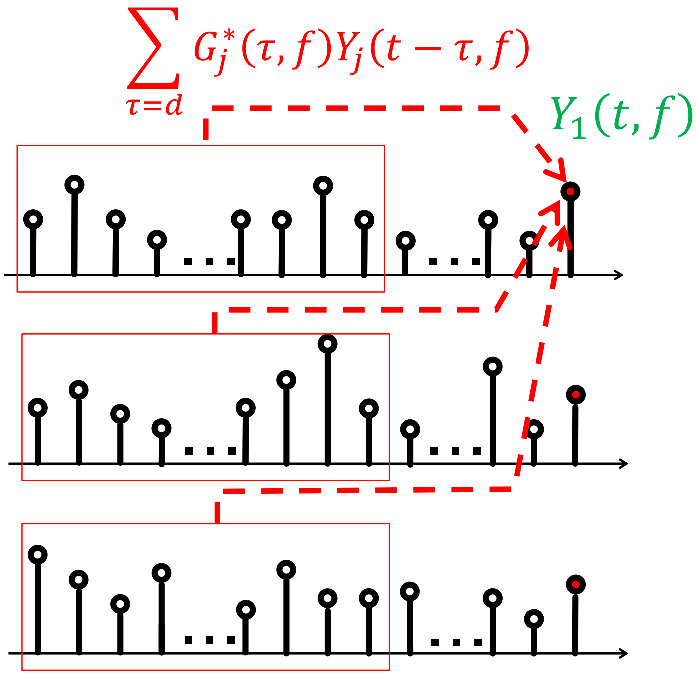
- For non-stationary signal with time-varying power $\phi_D(t,f)$

$$\{\hat{G}(\tau,f)\} = \underset{\{G(\tau,f)\}}{\operatorname{argmin}} \sum_t \underbrace{\frac{\| Y(t,f) - \sum_{\tau=d} G^*(\tau,f)Y(t-\tau,f) \|^2}{\phi_D(t,f)}}_{\text{Weighted prediction error (WPE)}}$$

Weighted prediction error (**WPE**)

Multi-channel extension

- Exploit past signals from all microphones to predict current signal at a microphone

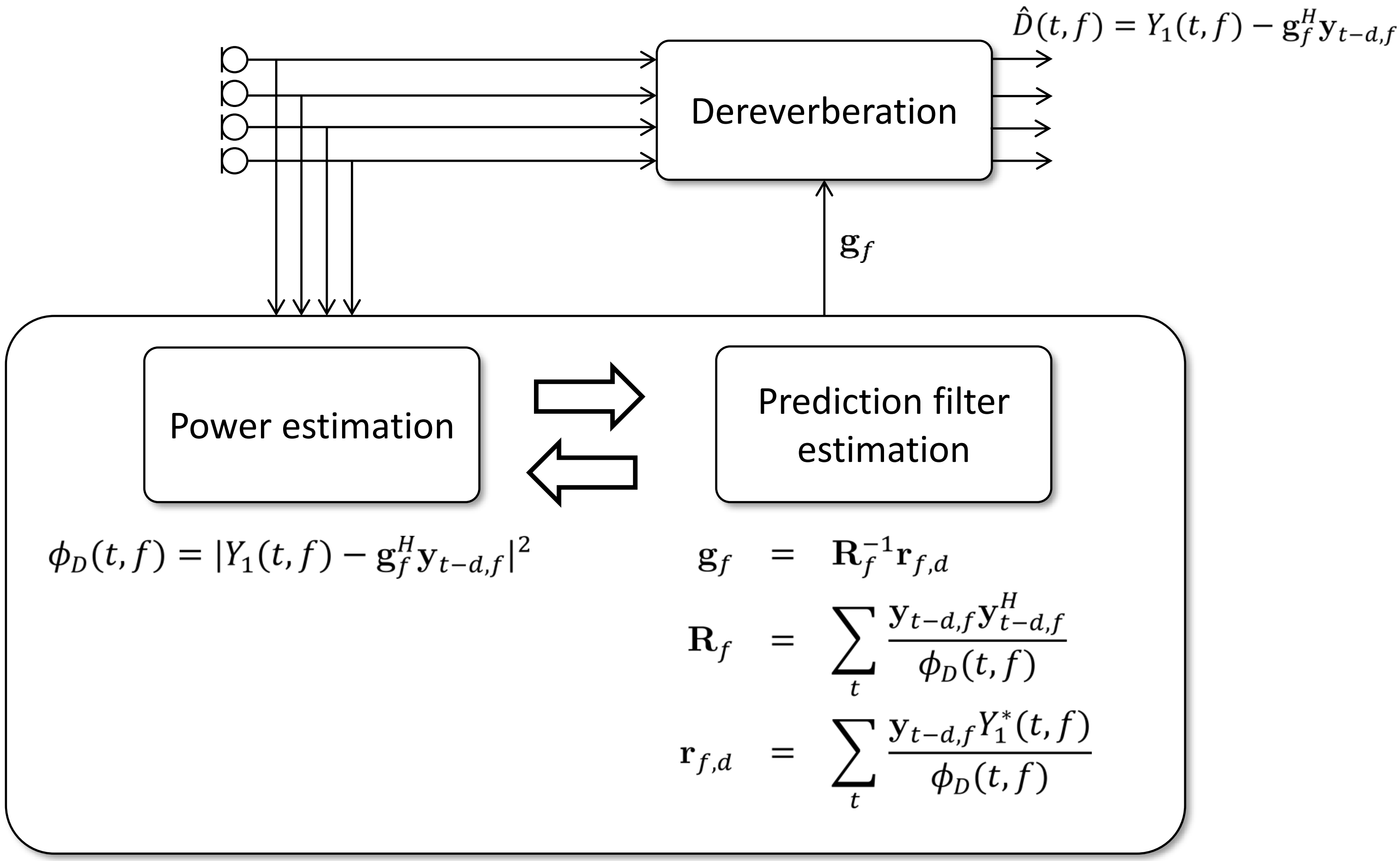


$$\hat{D}(t,f) = Y_1(t,f) - \sum_{j=1}^{J}\sum_{\tau=d} G_j^*(\tau,f)Y_j(t-\tau,f)$$
$$= Y_1(t,f) - \mathbf{g}_f^H \mathbf{y}_{t-d,f}$$

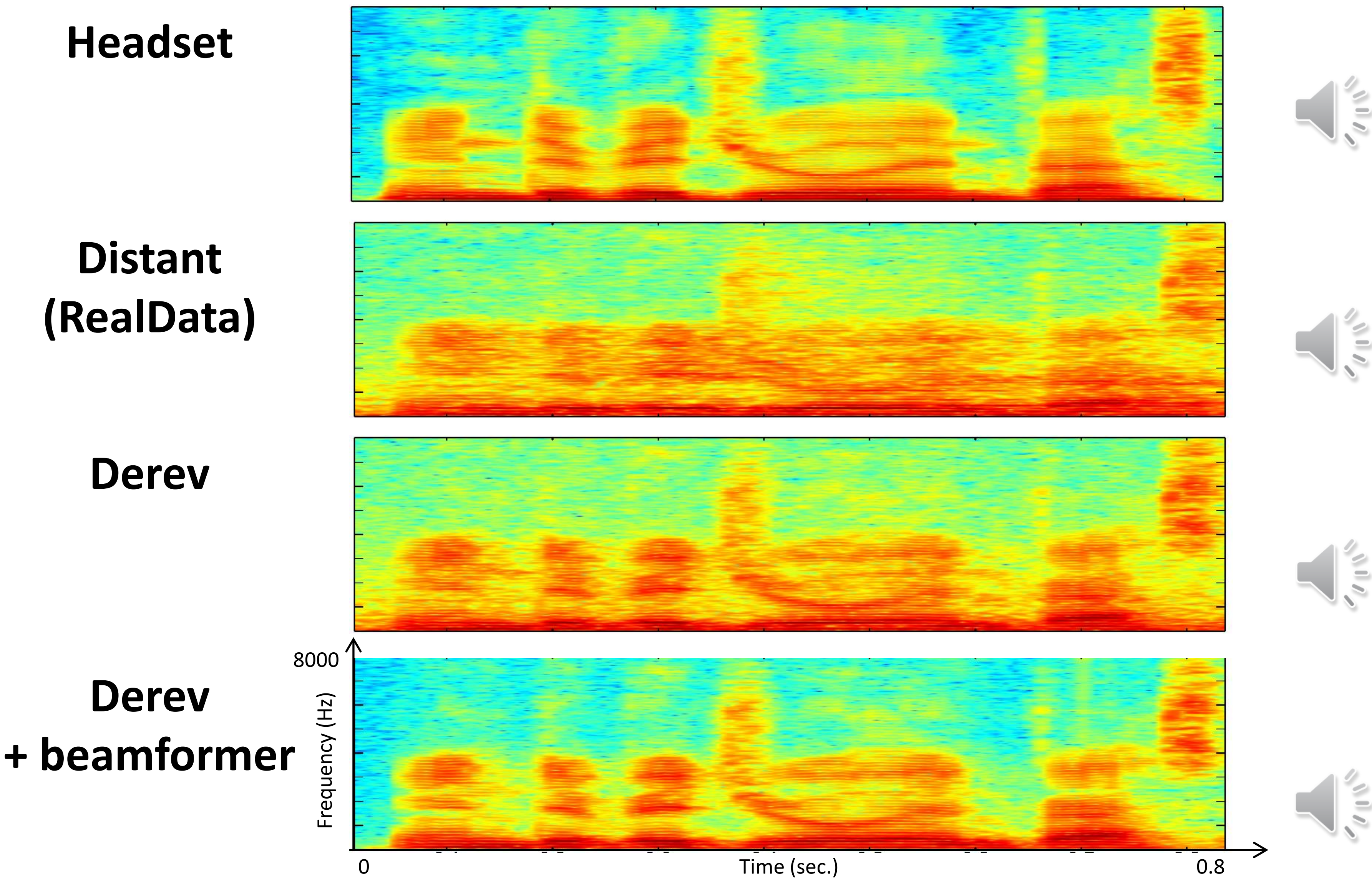
$$\mathbf{y}_{j,t,f} = [Y_j(t,f) \ldots Y_j(t-L,f)]^T$$
$$\mathbf{y}_{t,f} = [\mathbf{y}_{1,t,f}^T, \ldots, \mathbf{y}_{J,t,f}^T]^T$$
$$\mathbf{g}_{j,f} = [G_j(1,f) \ldots G_j(L,f)]^T$$
$$\mathbf{g}_f = [\mathbf{g}_{1,f}^T, \ldots, \mathbf{g}_{J,f}^T]^T$$

- Prediction filter obtained as $\hat{\mathbf{g}}_f = \underset{\mathbf{g}_f}{\operatorname{argmin}} \sum_t \frac{\left\| Y_1(t,f) - \mathbf{g}_f^H \mathbf{y}_{t-d,f} \right\|^2}{\phi_D(t,f)}$
- Can output multi-channel signals

Processing flow of WPE



Sound demo from REVERB challenge (Delcroix'14)



Results for REVERB and CHiME3

Front-end	REVERB (8 ch)	CHiME3 (6 ch)
-	19.2 %	15.6 %
WPE	12.9 %	14.7 %
WPE + MVDR Beamformer	9.3 %	7.6 %

Results for the REVERB task (Real Data, eval set) (Delcroix'15)
- DNN-based acoustic model trained with augmented training data
- Environment adaptation
- Decoding with RNN-LM

Results for the CHiME 3 task (Real Data, eval set) (Yoshioka'15)
- Deep CNN-based acoustic model trained with 6 channel training data
- No speaker adaptation
- Decoding with RNN-LM

Remarks

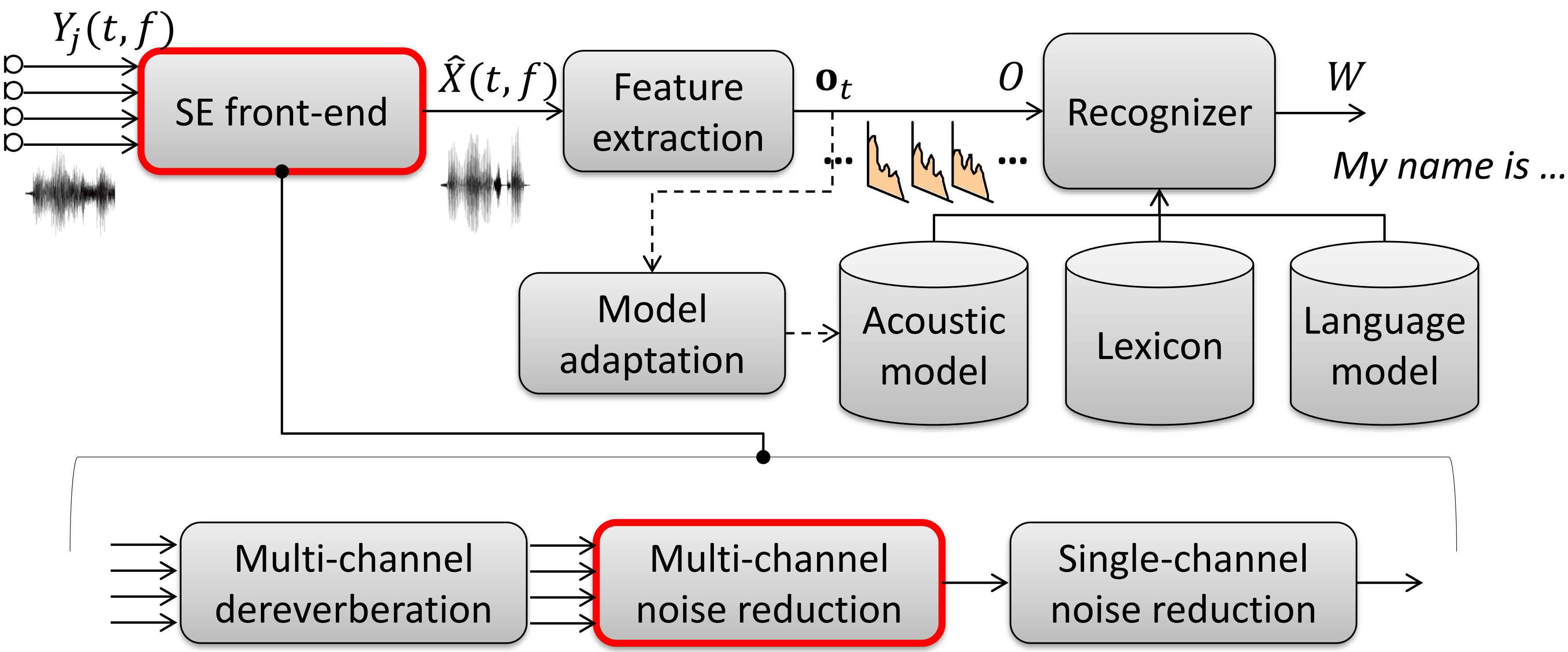
- Precise speech dereverberation with linear processing
  - Can be shown to cause no distortion to the target speech

    → Particularly efficient as an ASR front-end
- Can output multi-channel signals

    → Suited for beamformer pre-processing
- Relatively robust to noise
- Efficient implementation in STFT domain
- A few seconds of observation are sufficient to estimate the prediction filters

Matlab p-code available at: www.kecl.ntt.co.jp/icl/signal/wpe

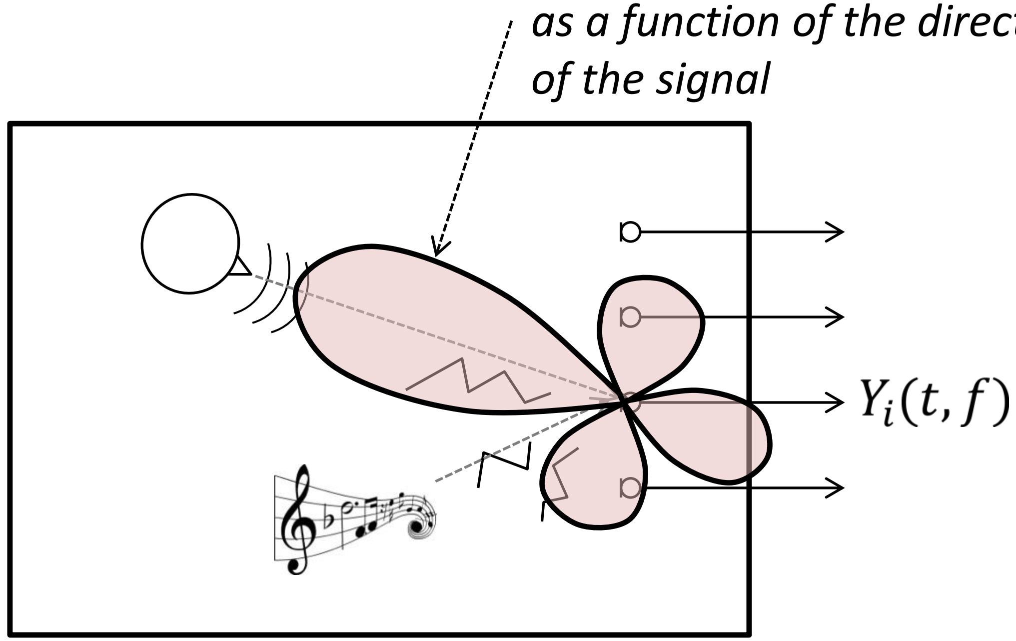
2.2 Beamforming



Principle

- Pickup signals in the direction of the target speaker
- Attenuate signals in the direction of the noise sources

Beam pattern – *microphone array gain as a function of the direction of arrival of the signal*



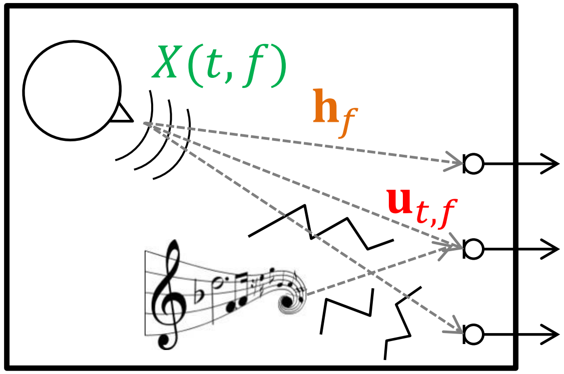
Microphone signal model

- Consider room impulse responses only within the STFT analysis window
  - Late reverberation as diffusive noise and included into the noise term

$$Y_j(t,f) \approx \sum_m H_j(m,f)X(t-m,f) + U_j(t,f)$$
$$= \underbrace{H_j(f)X(t,f)}_{O_j(t,f)} + U_j(t,f)$$

$O_j(t,f)$ source image at microphone j

- Using matrix notations



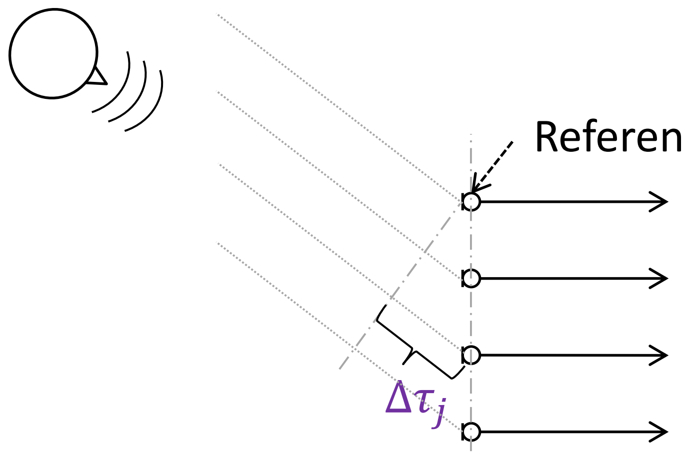
$$\mathbf{y}_{t,f} = \begin{bmatrix} Y_1(t,f) \\ \vdots \\ Y_J(t,f) \end{bmatrix} = \underbrace{\mathbf{h}_f \; X(t,f)}_{\triangleq \mathbf{o}_{t,f}} + \mathbf{u}_{t,f}$$

$\mathbf{o}_{t,f}$ Source image at microphones

$\mathbf{h}_f = [H_1(f), \ldots, H_J(f)]^{\mathrm{T}}$ Steering vector

Steering vector

- Represents the propagation from the source to the microphones, including
  - Propagation delays (information about the source direction)
  - Early reflections (reverberation within the analysis window)
- Example of plane wave assumption with free field condition
  (no reverberation and speaker far enough from the microphones)



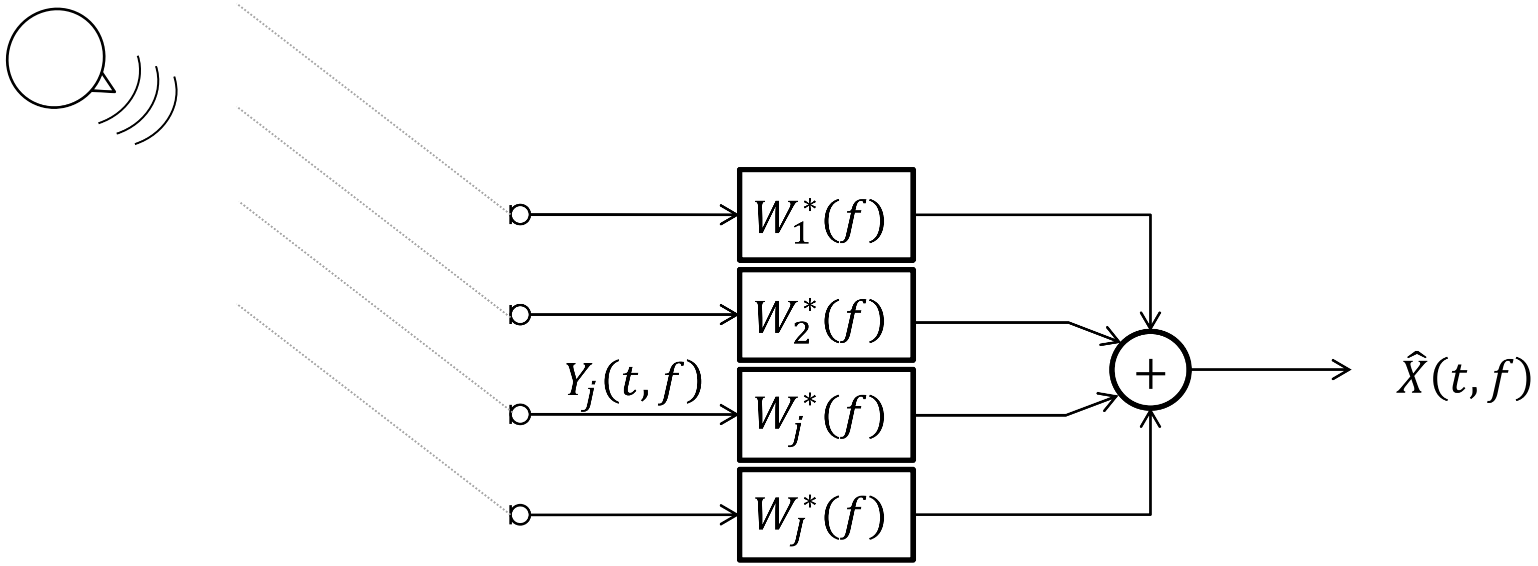
$$Y_j(t,f) = X(t,f)e^{-2\pi f \Delta\tau_j} + U_j(t,f)$$

$\Delta\tau_j$ time difference of arrival (TDOA)

Steering vector given as :

$$\mathbf{h}_f = \begin{bmatrix} e^{-2\pi f \Delta\tau_1} \\ \vdots \\ e^{-2\pi f \Delta\tau_J} \end{bmatrix}$$

Beamformer



- Output of beamformer

$$\hat{X}(t,f) = \sum_j W_j^*(f) Y_j(t,f)$$

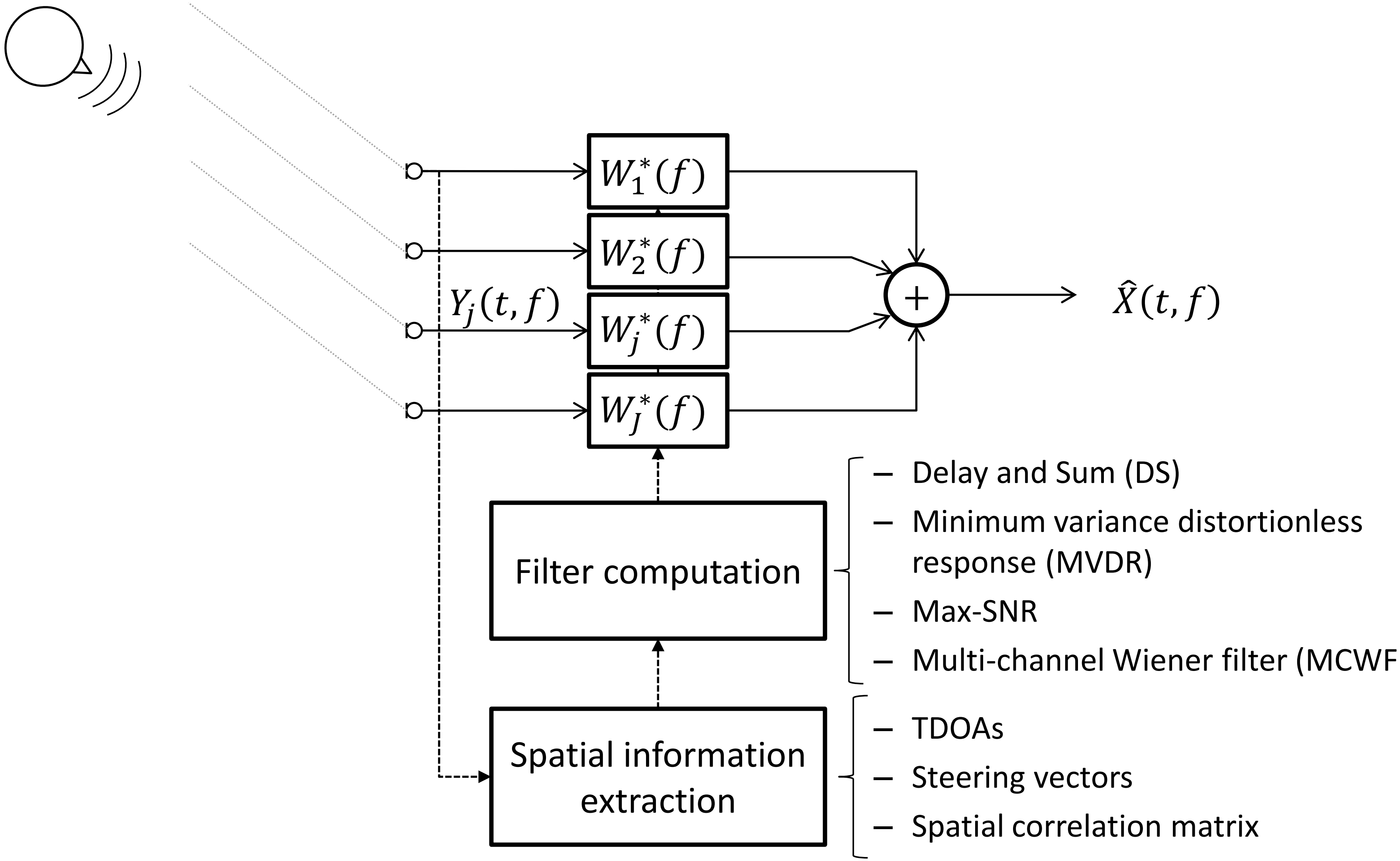
- Matrix notations

$$\hat{X}(t,f) = \mathbf{w}_f^H \mathbf{y}_{t,f}$$

$$\mathbf{w}_f = [W_1(f), \ldots, W_J(f)]^T \qquad \mathbf{y}_{t,f} = [Y_1(t,f), \ldots, Y_J(t,f)]^T$$

The filters $\mathbf{w}_f$ are designed to remove noise

Processing flow

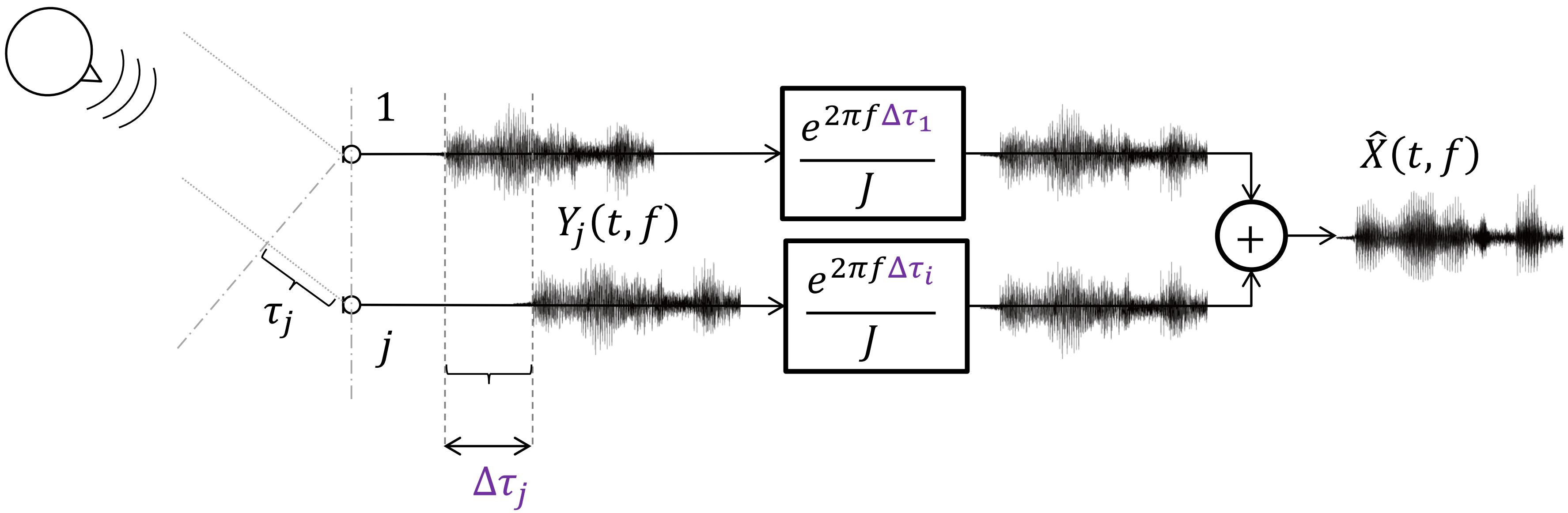


2.2.1 Delay and Sum beamformer

Delay and sum (DS) beamformer

(Van Veen'88)

- Align the microphone signals in time
  - Emphasize signals coming from the target direction
  - Destructive summation for signals coming from the other directions



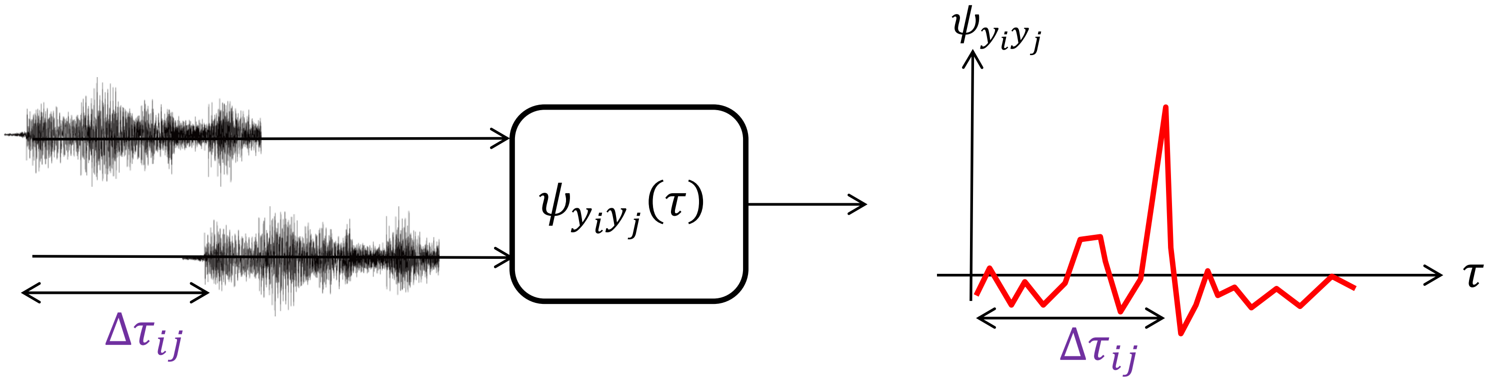
- Requires estimation of TDOAs $\Delta\tau_j$

TDOA estimation

- Signal cross correlation peaks when signals are aligned in time

$$\Delta\tau_{ij} = \arg\max_{\tau} \psi_{y_i y_j}(\tau)$$

$$\psi_{y_i y_j}(\tau) = E\{y_i(t) y_j(t+\tau)\}$$



- The cross correlation is sensitive to noise and reverberation
  - Usually use GCC-PHAT* coefficients that are more robust to reverberation

$$\psi_{y_i y_j}^{PHAT}(\tau) = IFFT\left(\frac{Y_i(f) Y_j^*(f)}{\left|Y_i(f) Y_j^*(f)\right|}\right)$$

(Knapp'76, Brutti'08)

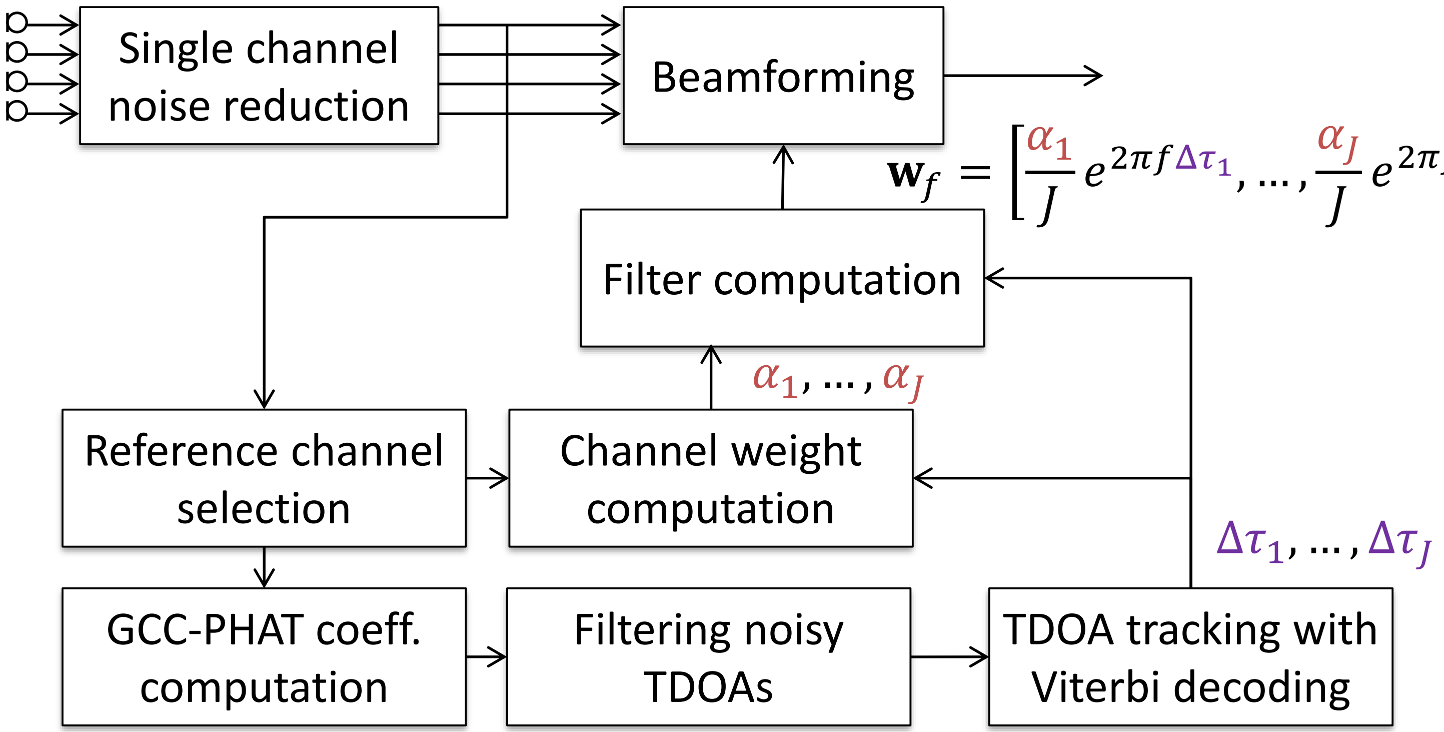
*Generalized Cross Correlation with Phase Transform (GCC-PHAT)

BeamformIt – a robust implementation of a weighted DS beamformer*

(Anguera'07)

- BeamformIt:
  - Used in baseline systems for several tasks, AMI, CHiME 3/4

    Toolkit available : www.xavieranguera.com/beamformit



$$\mathbf{w}_f = \left[\frac{\alpha_1}{J} e^{2\pi f \Delta\tau_1}, \dots, \frac{\alpha_J}{J} e^{2\pi f \Delta\tau_J}\right]^T$$

* Also sometimes called filter-and-sum beamformer

2.2.2 MVDR beamformer

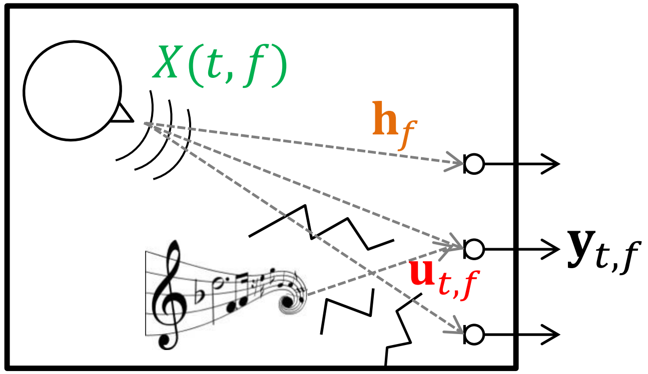
Minimum variance distortionless response (MVDR*) beamformer

- Beamformer output:

$$\hat{X}(t,f) = \mathbf{w}_f^H \mathbf{y}_{t,f} = \mathbf{w}_f^H \left(\mathbf{h}_f \, X(t,f)\right) + \mathbf{w}_f^H \mathbf{u}_{t,f}$$

Speech $X(t,f)$ is unchanged (distortionless): $\mathbf{w}_f^H \mathbf{h}_f = 1$

Minimize noise at the output of the beamformer



$$\Rightarrow \hat{X}(t,f) = X(t,f) + \mathbf{w}_f^H \mathbf{u}_{t,f}$$

- Filter is obtained by solving the following:

$$\mathbf{w}_f^{MVDR} = \underset{\mathbf{w}_f}{\operatorname{argmin}} \, E\{|\mathbf{w}_f^H \mathbf{u}_{t,f}|^2\}, \quad \text{subject to } \mathbf{w}_f^H \mathbf{h}_f = 1,$$

* MVDR beamformer is a special case of the more general linearly constrained minimum variance (LCMV) beamformer (Van Veen'88)

Expression of the MVDR filter

- MVDR filter given by

$$\mathbf{w}_f^{MVDR} = \frac{\left(\mathbf{R}_f^{noise}\right)^{-1}\mathbf{h}_f}{\mathbf{h}_f^H\left(\mathbf{R}_f^{noise}\right)^{-1}\mathbf{h}_f}$$

  - Where $\mathbf{R}_f^{noise}$ is the spatial correlation matrix* of the noise, which measures the correlation among noise signals at the different microphones

$$\mathbf{R}_f^{noise} = \sum_t \mathbf{u}_{t,f}\mathbf{u}_{t,f}^H = \begin{bmatrix} \frac{1}{T}\sum_t^T U_1(t,f)U_1^*(t,f) & \cdots & \frac{1}{T}\sum_t^T U_1(t,f)U_J^*(t,f) \\ \vdots & \ddots & \vdots \\ \frac{1}{T}\sum_t^T U_J(t,f)U_1^*(t,f) & \cdots & \frac{1}{T}\sum_t^T U_J(t,f)U_J^*(t,f) \end{bmatrix}$$

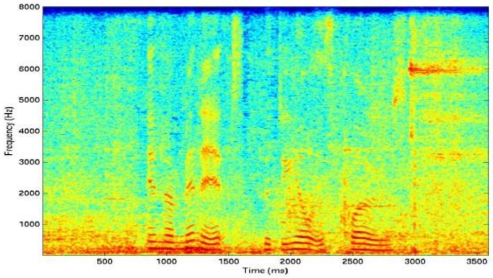
* The spatial correlation matrix is also called cross spectral density

Steering vector estimation

The steering vector $\mathbf{h}_f$ can be obtained as the principal eigenvector of the spatial correlation matrix of the source image signals $\mathbf{R}_f^{speech}$

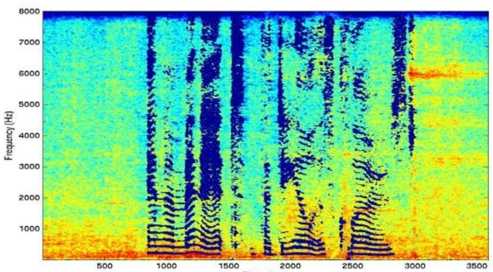
$$\mathbf{h}_f = \mathcal{P}\left(\mathbf{R}_f^{speech}\right)$$

Microphone signal (speech + noise)



$$\mathbf{R}_f^{obs} = \sum_t \mathbf{y}_{t,f}\mathbf{y}_{t,f}^H$$

Noise estimate



$M(t,f)Y_i(t,f)$

$$\mathbf{R}_f^{noise} = \frac{\sum_t M(t,f)\mathbf{y}_{t,f}\mathbf{y}_{t,f}^H}{\sum_t M(t,f)}$$

Source image spatial correlation matrix

$$\mathbf{R}_f^{speech} = \mathbf{R}_f^{obs} - \mathbf{R}_f^{noise}$$

Spectral masks

$$M(t,f) = \begin{cases} 1 & \text{if noise} > \text{speech} \\ 0 & \text{otherwise} \end{cases}$$

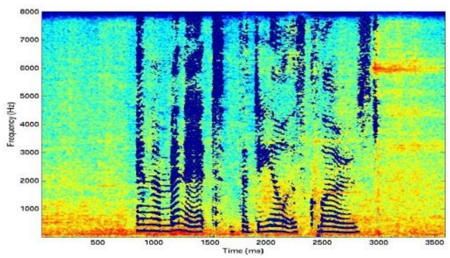
(Souden'13, Higuchi'16, Yoshioka'15, Heymann'15)

Spectral mask estimation

- Clustering of spatial features for mask estimation
  - Source models
    - Watson mixture model (Souden’13)
    - Complex Gaussian mixture model (Higuchi’16)

E-step: update masks

$$M_{t,f} = p\left(noise \middle| \boldsymbol{y}_{t,f}, \mathbf{R}_f^{noise}, \mathbf{R}_f^{speech}\right)$$



$M_{t,f}$ $\rightarrow$

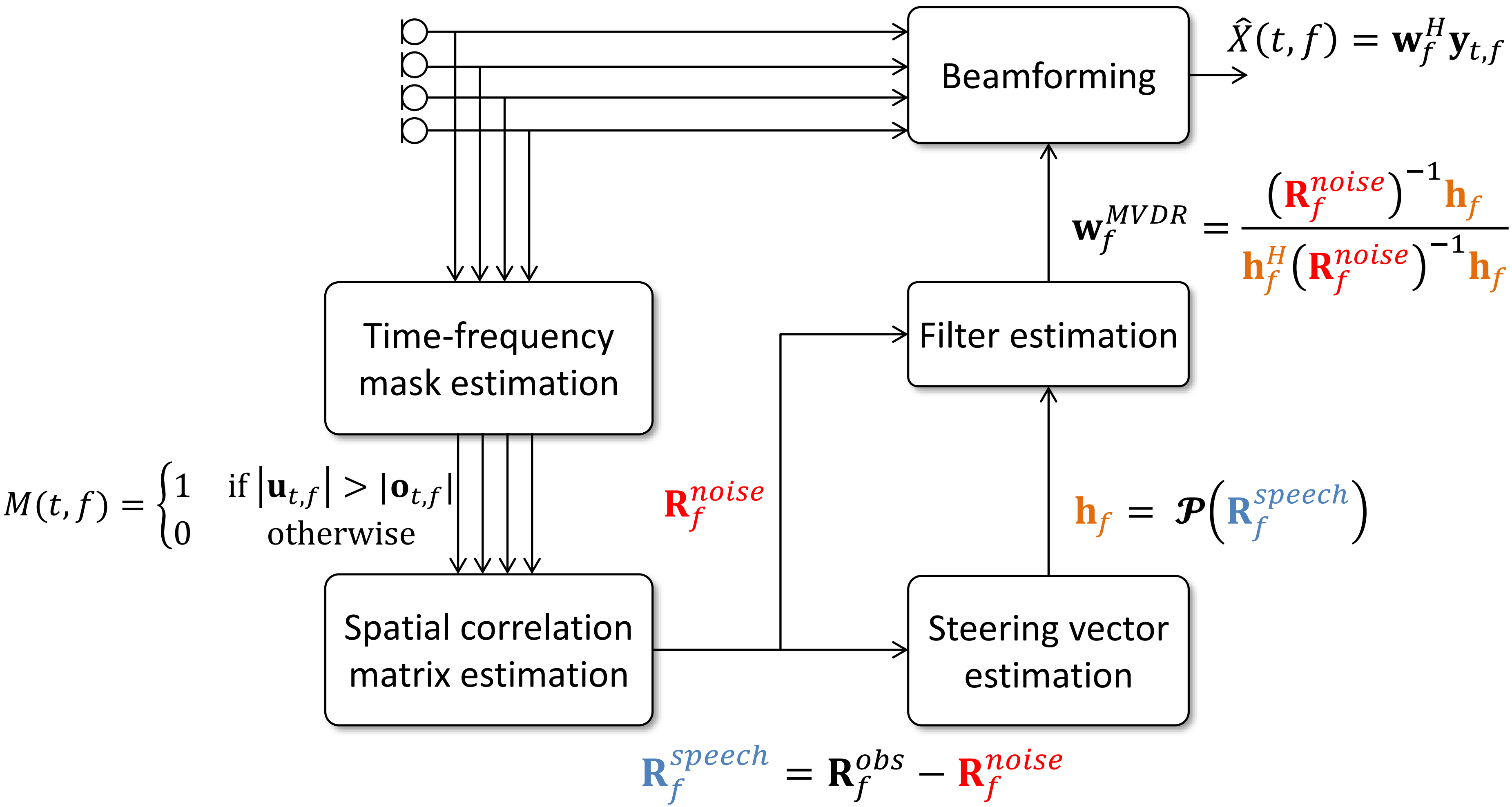
$\leftarrow$ $\mathbf{R}_f^{noise}$

M-step: update spatial corr. matrix

$$\mathbf{R}_f^{noise} = \frac{\sum_t M(t,f)\mathbf{y}_{t,f}\mathbf{y}_{t,f}^H}{\sum_t M(t,f)}$$

- Neural network-based approach (Hori'15, Heymann’15)
  - See slides 94-96

Processing flow of MVDR beamformer



Other beamformers

- Max-SNR beamformer* (VanVeen'88, Araki'07, Waritz'07)
  - Optimize the output SNR without the distortionless constraint

$$\mathbf{w}_f^{maxSNR} = \mathcal{P}\left(\left(\mathbf{R}_f^{noise}\right)^{-1}\mathbf{R}_f^{obs}\right)$$

- Multi-channel Wiener filter (MCWF) (Doclo'02)
  - Preserves spatial information at the output (multi-channel output)

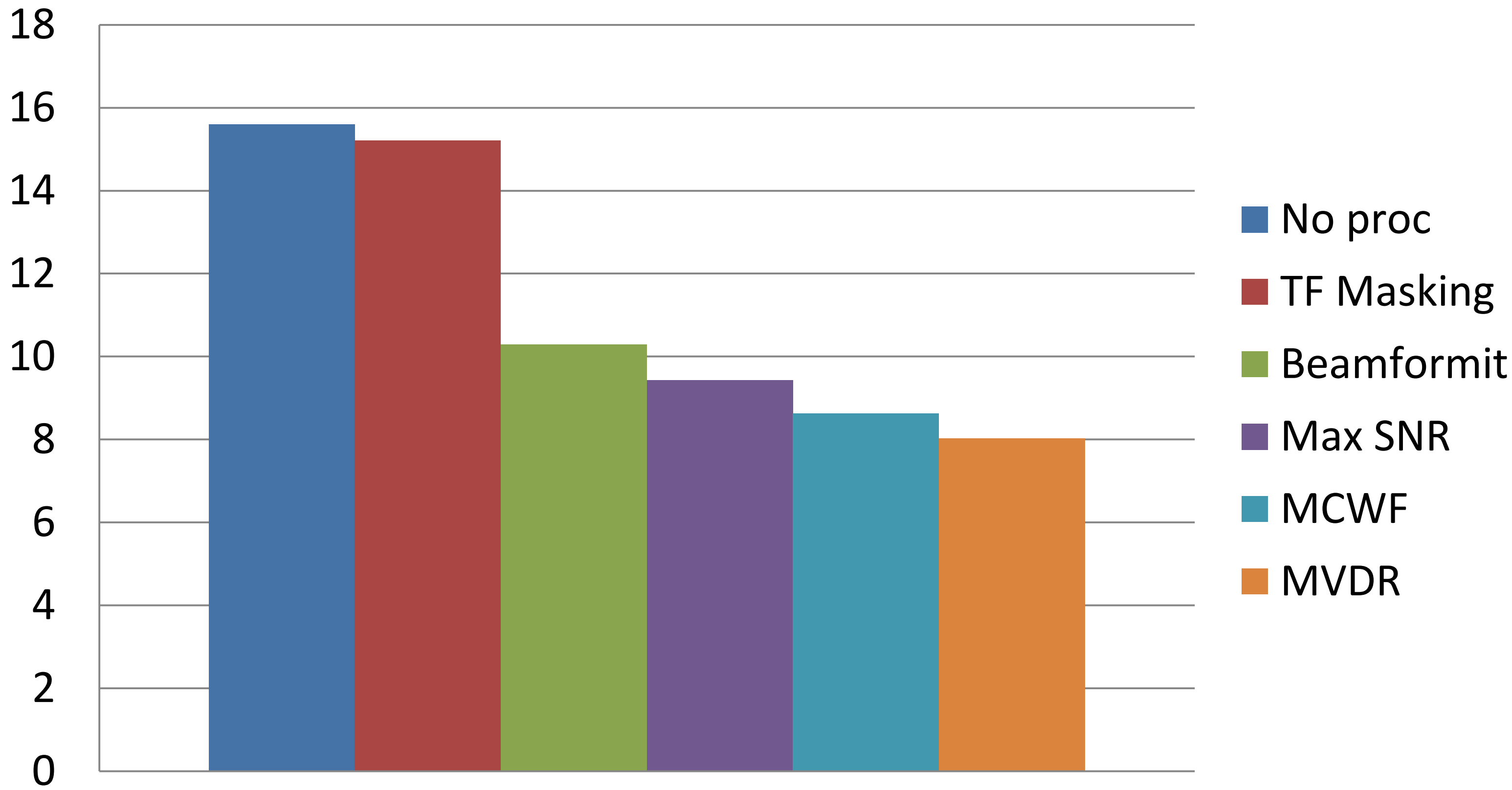
$$\mathbf{w}_f^{MCWF} = \left(\mathbf{R}_f^{obs}\right)^{-1}\mathbf{R}_f^{speech}$$

→ Max-SNR beamformer and MCWF can also be derived from the spatial correlation matrices

* Max-SNR beamformer is also called generalized eigenvalue beamformer

2.2.3 Experiments

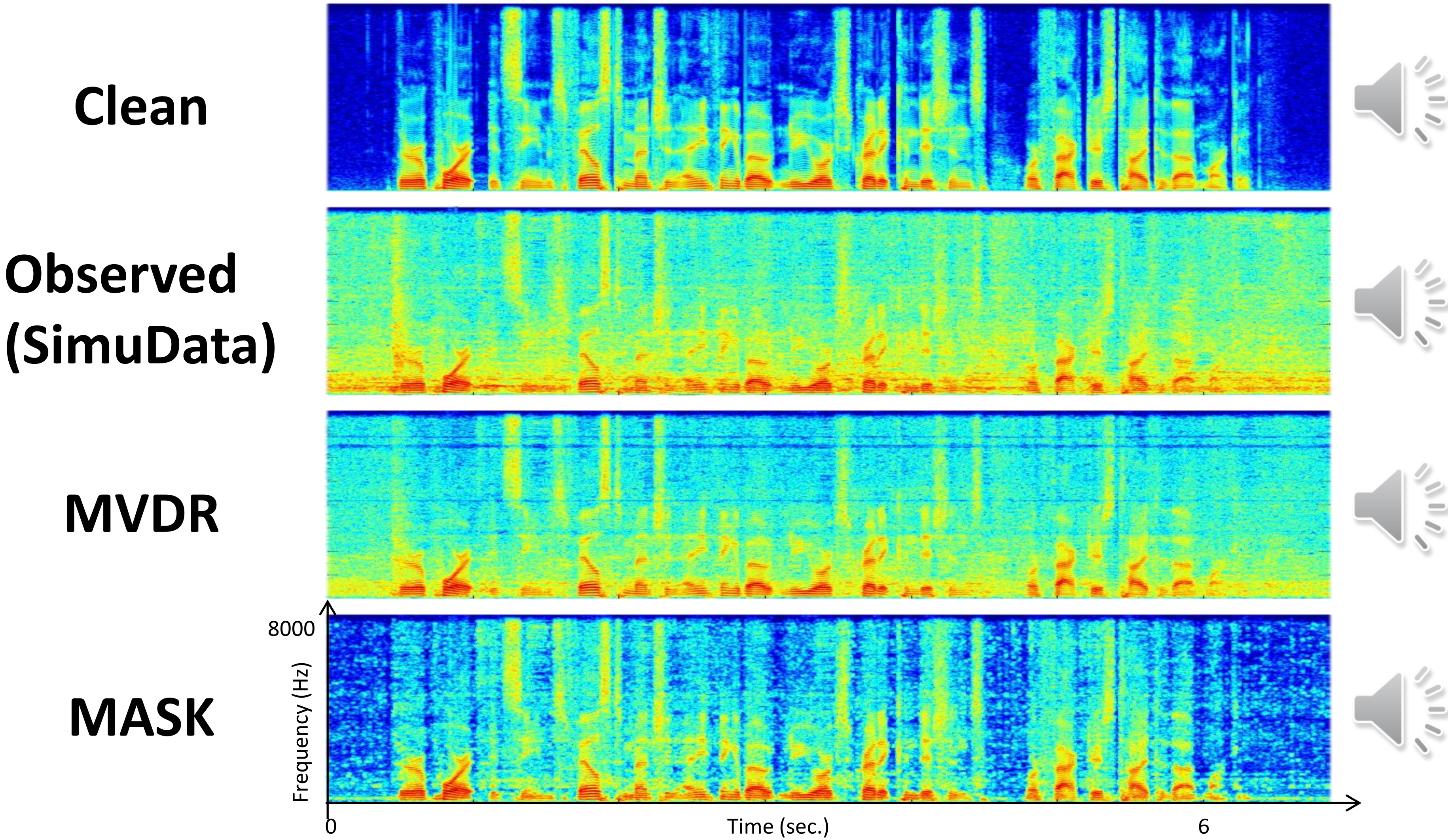
CHiME 3 results



Results for the CHiME 3 task (Real Data, eval set)

- Deep CNN-based acoustic model trained with 6 channel training data
- No speaker adaptation
- Decoding with RNN-LM

Sound demo



Clean
Observed
(SimuData)
MVDR
MASK
8000
Frequency (Hz)
0
Time (sec.)
6

remarks

- Delay-and-sum beamformer
  ☺ Simple approach
  ☹ Relies on correct TDOA estimation
    - Errors in TDOA estimation may result in amplifying noise
  ☹ Not optimal for noise reduction in general

- Weighted DS beamformer (BeamformIt)
  ☺ Includes weights to compensate for amplitude differences among the microphone signals
  ☺ Uses a more robust TDOA estimation than simply GCC-PHAT
    - Still potentially affected by noise and reverberation
  ☹ Not optimal for noise reduction

- MVDR beamformer
  ☺ Optimized for noise reduction while preserving speech (distortionless)
  – Extracting spatial information is a key for success
    - From TDOA → Poor performance with noise and reverberation
    - From signal statistics → More robust to noise and reverberation
  ☹ More involving in terms of computations compared to DS beamformer

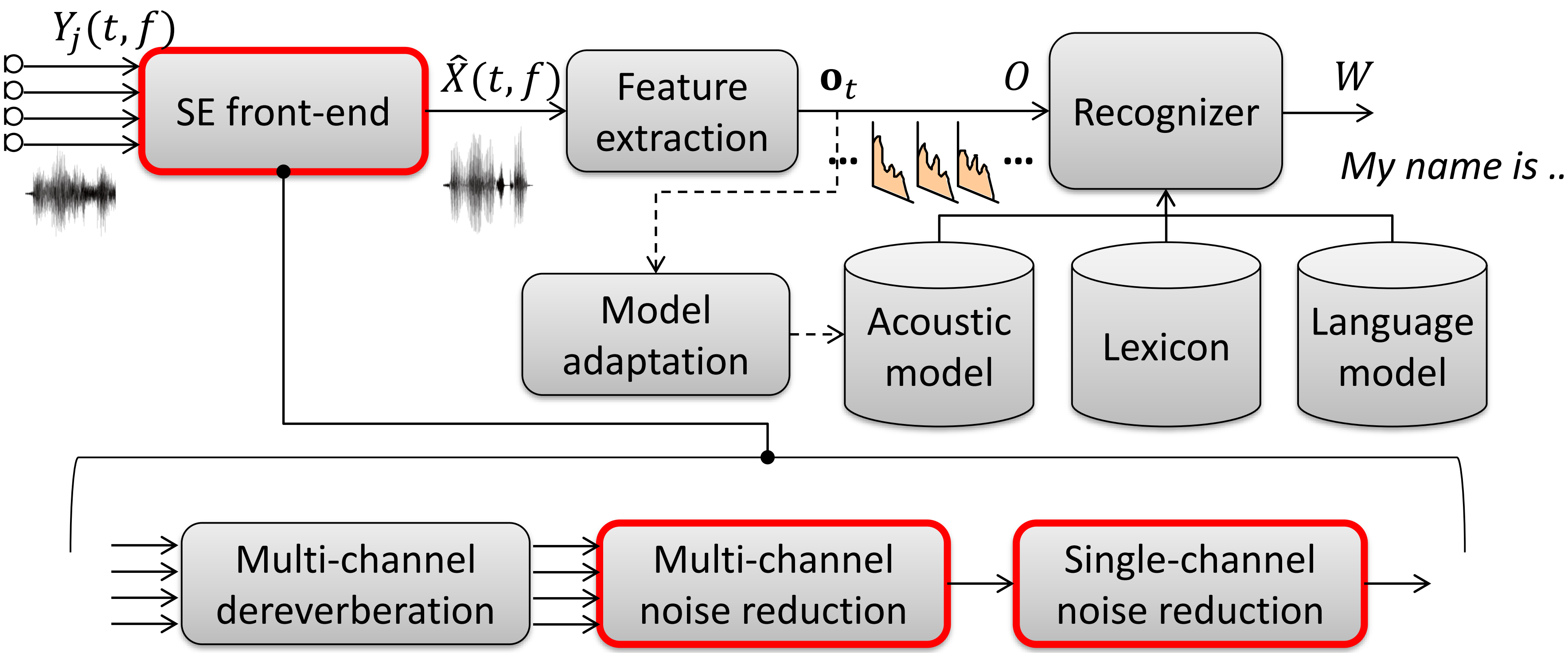
Remarks

- Beamforming can greatly reduce WER even when using a strong ASR back-end
  - Beamforming outperforms TF masking for ASR
    - TF masking removes more noise
    - Linear filtering causes less distortion (especially with the distortionless constraint)

    → This leads to better ASR performance

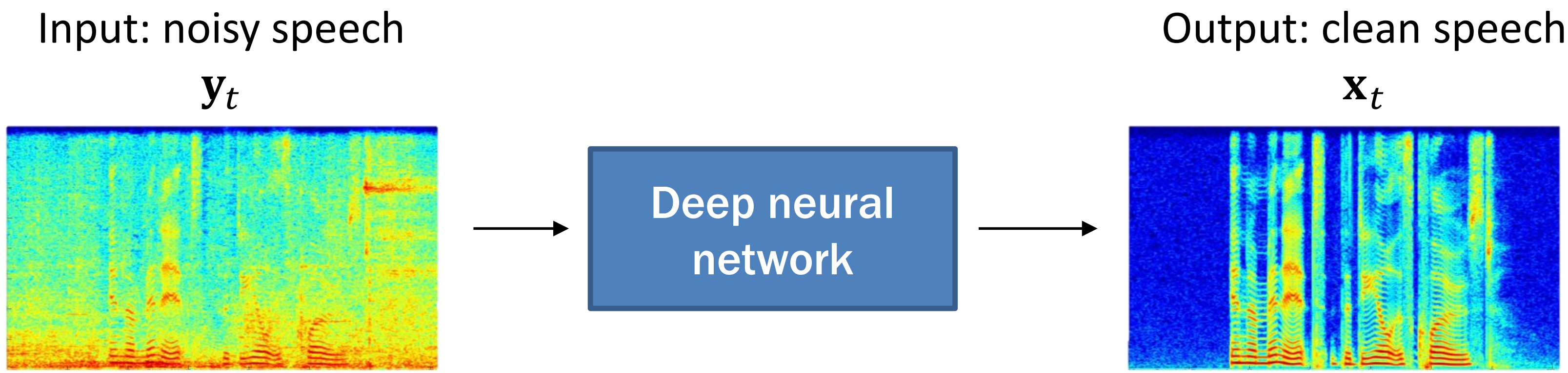
- Future directions
  - Online extension (source tracking)
  - Multiple speakers

2.3 Deep neural network based enhancement



Deep network based enhancement: Parallel data processing

- Basic architecture: regression problem
  → Train a neural network to map noisy speech to clean speech



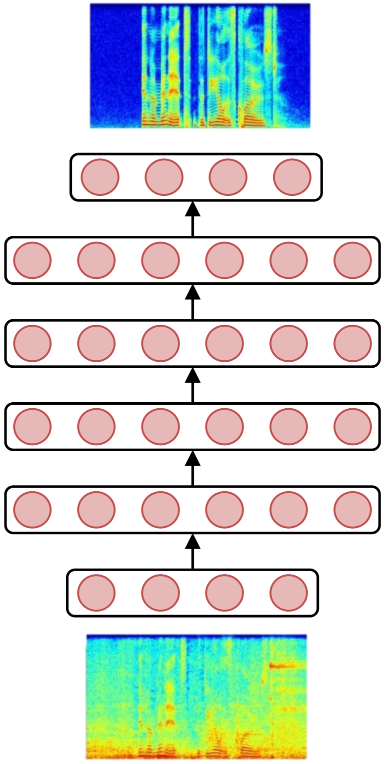
- Many variations investigated in terms of
  - Objective functions
  - Architectures
  - Input/output

2.3.1 Objective functions

Regression based DNN

(Xu’15)

Output: clean speech feature $\mathbf{x}_t$



Input: noisy speech features $\mathbf{y}_t$

- Train a DNN to directly predict the clean spectrum from the noisy speech spectrum
- Objective function: minimum mean square error (MMSE) between clean and enhanced signal,

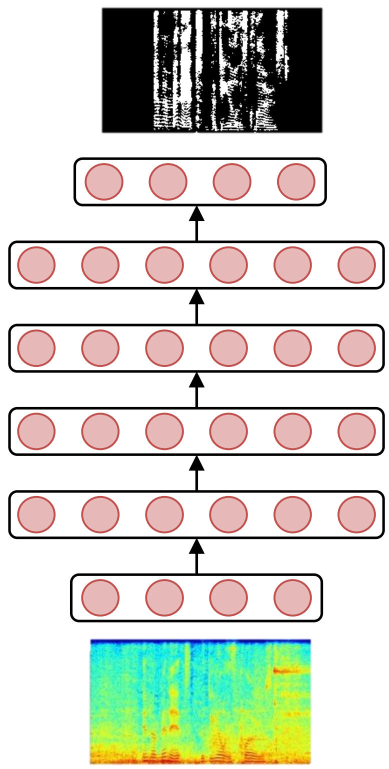
$$J(\theta) = \sum_t |\mathbf{x}_t - \mathbf{h}_t^L(\theta)|^2$$

  – $\mathbf{x}_t$ clean speech feature (output)
    - Log power spectrum
  – $\mathbf{y}_t$ noisy speech feature (input)
    - Log power spectrum + Context
  – $\mathbf{h}_t^L$ network output
    - $\mathbf{h}_t^L$ can be unbounded (i.e., $\mathbf{h}_t^L \in [-\infty, \infty]$, which is considered to be difficult
    - Normalize the output by $[-1, 1]$
    - Use $\tanh()$ as an activation function
  – θ network parameters
- When trained with sufficient data, it can be used to enhance speech in unseen noisy conditions

Mask-estimation based DNN (Cross entropy)

(Narayanan'13, Wang'16)

Output: time-frequency mask $\mathbf{m}_t$



Input: noisy speech features $\mathbf{y}_t$

- Train a DNN to predict the coefficient of an ideal ratio mask (IRM)

$$m_{t,f} = \frac{x_{t,f}}{x_{t,f} + u_{t,f}} = \frac{clean}{clean + noise}$$

- Objective function: cross entropy (CE) between estimated mask and IRM

$$J(\theta) = -\sum_{t,f} m_{t,f} \log\left(h^L_{t,k}(\theta)\right) - \left(1 - m_{t,f}\right) \log\left(1 - h^L_{t,k}(\theta)\right)$$

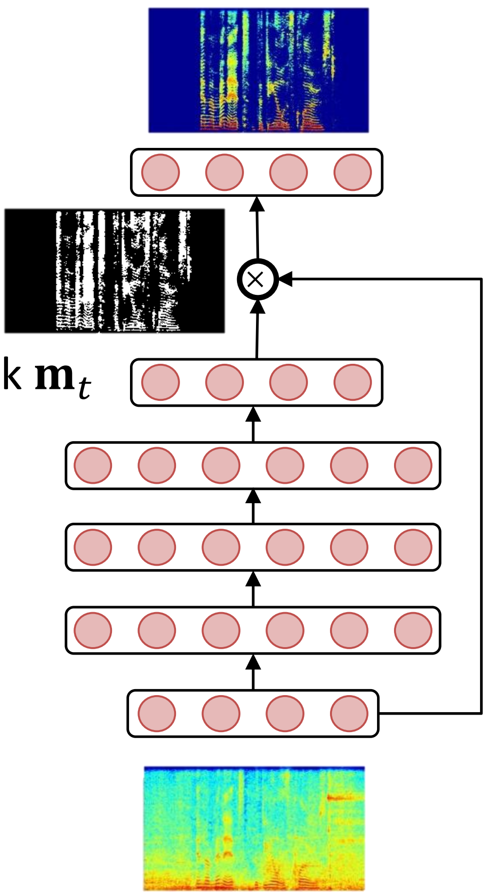
  – $\mathbf{h}^L_t$ network output (continuous mask)
    - Bounded with $m^L_t \in [0, 1]$, using a sigmoid function
    - Simplifies learning and tends to perform better than directly estimating clean speech

- Enhanced signal obtained as $\hat{\mathbf{x}}_t = \mathbf{m}_t \circ \boldsymbol{y}_t$

Mask estimation based DNN (MMSE)

Output: clean speech feature $\mathbf{x}_t$

(Weninger '15)



Mask $\mathbf{m}_t$

Input: noisy speech features $\mathbf{y}_t$

- Train a DNN to predict the coefficient of a time-frequency mask $\mathbf{m}_t = \mathbf{h}_t^L$
  - Do not restrict the output to the IRM

- Objective function: minimum mean square error (MMSE) between clean and enhanced signal,

$$J(\theta) = \sum_t \left| \mathbf{x}_t - \mathbf{m}_t(\theta) \circ \mathbf{y}_t \right|^2$$

  - $\mathbf{x}_t$ clean speech feature (output)
    - Magnitude spectrum
  - $\mathbf{y}_t$ noisy speech feature (input)
    - Log mel filterbank spectrum (as input to the network)
    - Magnitude spectrum to compute the enhanced signal
  - $\mathbf{m}_t$ network output (continuous mask)
    - Bounded with $m_t^L \in [0, 1]$ using a sigmoid function

Experiments on CHiME 2

Results from (Wang'16)

Front-end	WER
-	16.2 %
Mask-estimation with cross entropy	14.8 %

Can be jointly trained with the ASR back-end

→ More details in *3.4 Integration of front-end and back-end with deep networks*

Enhancement DNN

- Predict mask (CE Objective function)
- Features: Log power spectrum

Acoustic model DNN

- Log Mel Filterbanks
- Trained on noisy speech

2.3.2 Recurrent architectures

Exploiting recurrent networks

- Neural network based enhancement
  - Exploits only the context seen within its input features
  - Noise reduction could benefit from exploiting longer context

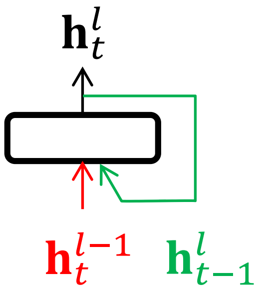
→ Some investigations for RNN-based approaches (Weninger'14, Weninger'15, Erdogan'15, Heymann'15)

LSTM: Long Short-Term Memory RNN

- Elman RNN

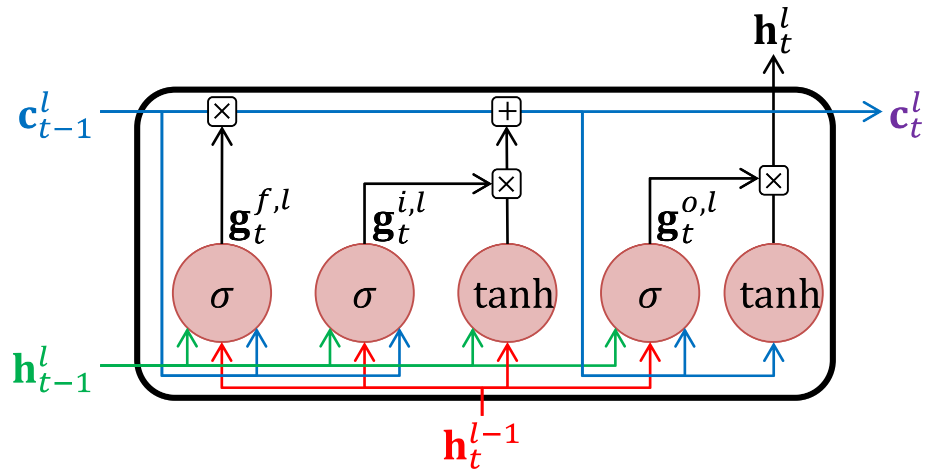
$$\mathbf{h}_t^l = \sigma\left(\mathbf{W}^l \begin{bmatrix} \mathbf{h}_t^{l-1} \\ \mathbf{h}_{t-1}^l \end{bmatrix} + \mathbf{b}^l\right)$$

  - Vanishing gradient due to recurrent weights $\mathbf{W}^l$



- LSTM
  - Avoids recurrent weights in the Elman form by introducing gates $(\mathbf{g}_t^{f,l}, \mathbf{g}_t^{i,l}, \mathbf{g}_t^{o,l})$ and cell states $\mathbf{c}_t^l$

$$\mathbf{h}_t^l = \mathbf{g}_t^{o,l} \circ \tanh(\mathbf{c}_t^l)$$



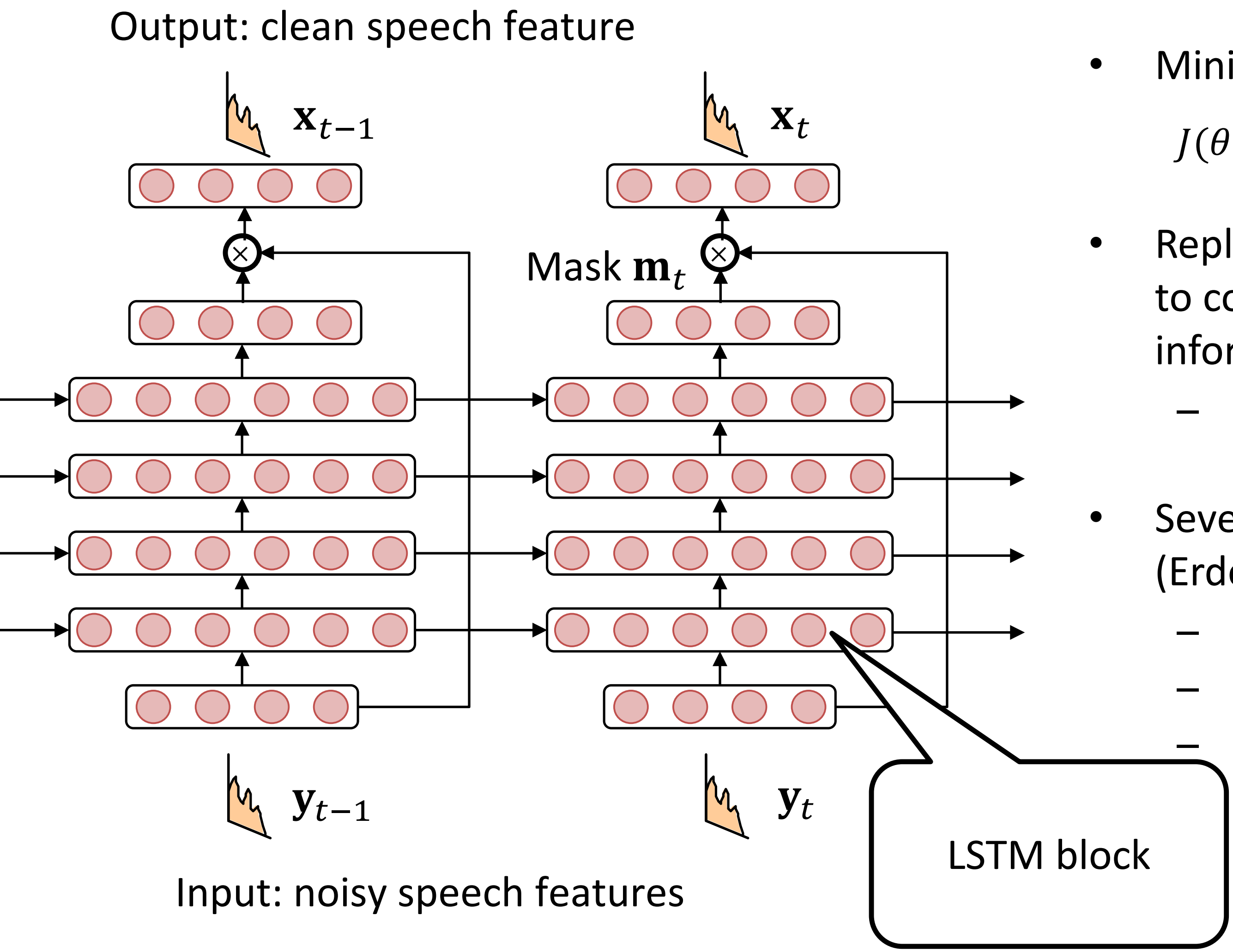
Cell state:

$$\mathbf{c}_t^l = \mathbf{g}_t^{f,l} \circ \mathbf{c}_{t-1}^l + \mathbf{g}_t^{i,l} \circ \tanh\left(\mathbf{W}^{c,l} \begin{bmatrix} \mathbf{h}_t^{l-1} \\ \mathbf{h}_{t-1}^l \end{bmatrix} + \mathbf{b}^{fc,l}\right)$$

Forget, input and output gates:

$$\mathbf{g}_t^{f,l} = \sigma\left(\mathbf{W}^{f,l} \begin{bmatrix} \mathbf{h}_t^{l-1} \\ \mathbf{h}_{t-1}^l \\ \mathbf{c}_{t-1}^l \end{bmatrix} + \mathbf{b}^{f,l}\right), \mathbf{g}_t^{i,l} = \sigma\left(\mathbf{W}^{i,l} \begin{bmatrix} \mathbf{h}_t^{l-1} \\ \mathbf{h}_{t-1}^l \\ \mathbf{c}_{t-1}^l \end{bmatrix} + \mathbf{b}^{i,l}\right), \mathbf{g}_t^{o,l} = \sigma\left(\mathbf{W}^{o,l} \begin{bmatrix} \mathbf{h}_t^{l-1} \\ \mathbf{h}_{t-1}^l \\ \mathbf{c}_t^l \end{bmatrix} + \mathbf{b}^{o,l}\right)$$

Mask estimation based LSTM



- Minimize Mean Square Error

$$J(\theta) = \sum_t \left| \mathbf{x}_t - \mathbf{m}_t \circ \mathbf{y}_t \right|^2$$

- Replace DNN with LSTM-RNN to consider long-context information
  - known to be effective for speech modeling
- Several extensions (Erdogan'15)
  - Bidirectional LSTM
  - Phase sensitive objectives
  - Recognition boosted features

Effect of introducing LSTM

Front-end	WER
-	31.2 %
DNN based enhancement	29.7 %
LSTM based enhancement	26.1 %

Experiments on CHiME 2 Dev set with DNN back-end

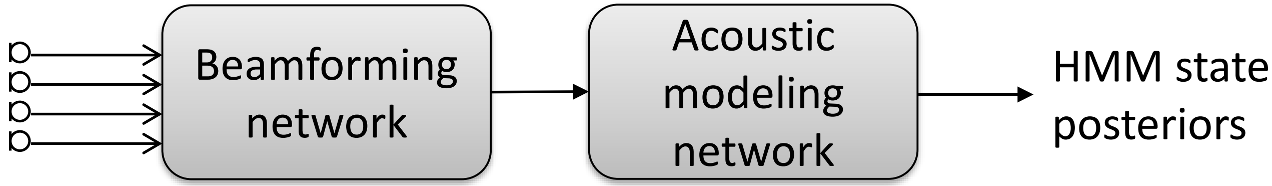
2.3.3 Multi-channel extensions

Multi-channel extensions

- Estimate mask for noise $M(t,f)$ using neural network
  - Use the mask to compute the noise spatial correlation matrix that is used to derive the beamformer filters (see slide 74)

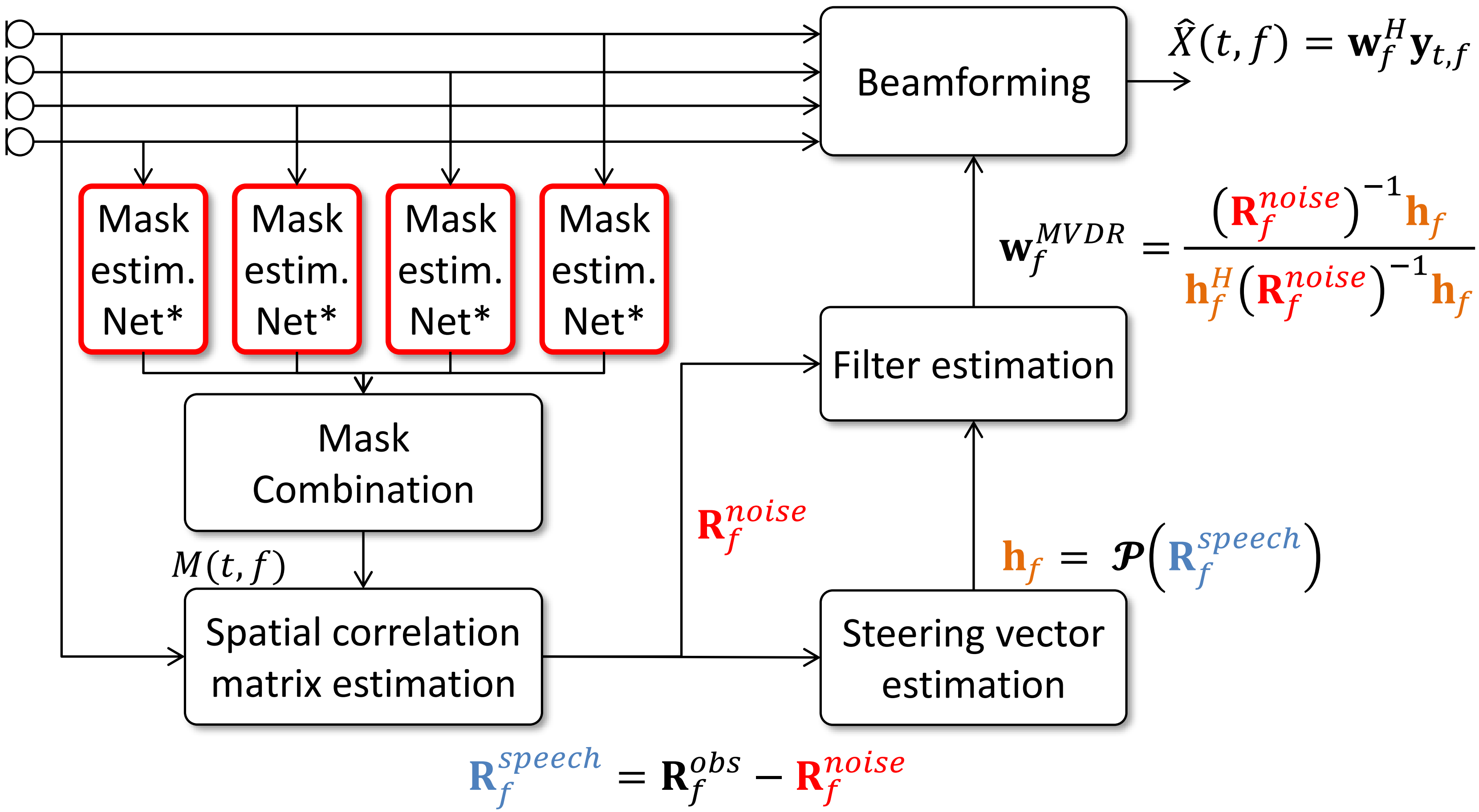
$$\mathbf{R}_f^{NOISE} = \frac{\sum_t M(t,f)\mathbf{y}_{t,f}\mathbf{y}_{t,f}^H}{\sum_t M(t,f)}$$

- Beamforming networks or multi-channel deep networks
  - Design a network to perform beamforming
  - Can be jointly trained with the acoustic model
  - More details in *3.4 Integration of front-end and back-end with deep networks*



DN-based mask estimation for beamforming

(Heymann'15, Hori'15, Heymann'16)
http://github.com/fgnt/nn-gev



** Masks derived from 1ch signals → does not exploit spatial information for mask estimation*

CHiME 3 investigations

(Heymann'16)

Front-end	WER
-	40.2 %
BeamformIt	22.7 %
DNN mask estimation + MaxSNR BF	17.7 %
BLSTM mask estimation + MaxSNR BF	15.4 %

Avg. results for Real eval sets

ASR back-end

- DNN-based AM
- Retrained on enhanced speech

Remarks

- Exploit deep-learning for speech enhancement
  - ☺ Possible to train complex non-linear function for regression
  - ☺ Exploits long context, extra input features…
  - ☺ Online mask estimation/enhancement
  - ☺ Offers the possibility for jointly train the front-end and back-end

- Requirements
  - Relatively large amount of training data
  - Noisy/Clean parallel corpus
    - This requirement can be potentially released if SE front-end and acoustic models are jointly trained or when predicting masks (Heymann'16)

References (SE-Front-end 1/3)

(Anguera'07) Anguera, X., et al. "Acoustic beamforming for speaker diarization of meetings," IEEE Trans. ASLP (2007).

(Araki'07) Araki, S., et al. "Blind speech separation in a meeting situation with maximum snr beamformers," Proc. ICASSP (2007).

(Boll'79) Boll, S. "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. ASSP (1979).

(Brutti'08) Brutti, A., et al. "Comparison between different sound source localization techniques based on a real data collection," Proc. HSCMA (2008).

(Delcroix'14) Delcroix, M., et al. "Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB challenge," Proc. REVERB (2014).

(Delcroix'15) Delcroix, M., et al. "Strategies for distant speech recognition in reverberant environments," EURASIP Journal ASP (2015).

(Doclo'02) Doclo, S., et al. "GSVD-based optimal filtering for single and multi-microphone speech enhancement,". IEEE Trans. SP (2002).

(Erdogan'15) Erdogan, H., et al. "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," Proc. ICASSP (2015).

(Haykin'96) Haykin, S. "Adaptive Filter Theory (3rd Ed.)," Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1996).

(Heymann'15) Heymann, J., et al. "BLSTM supported GEV Beamformer front-end for the 3rd CHiME challenge," Proc. ASRU (2015).

(Heymann'16) Heymann, J., et al. "Neural network based spectral mask estimation for acoustic beamforming," Proc. ICASSP (2016).

(Higuchi'16) Higuchi, T., et al. "Robust MVDR beamforming using time frequency masks for online/offline ASR in noise," Proc. ICASSP (2016).

(Hori'15) Hori, T., et al. "The MERL/SRI system for the 3rd CHiME challenge using beamforming, robust feature extraction, and advanced speech recognition," Proc. of ASRU (2015).

(Jukic'14) Jukic, A., et al. "Speech dereverberation using weighted prediction error with Laplacian model of the desired signal," Proc. ICASSP (2014).

References (SE-Front-end 2/3)

(Kinoshita'07) Kinoshita, K., et al., "Multi-step linear prediction based speech dereverberation in noisy reverberant environment," Proc. Interspeech (2007).

(Knapp'76) Knapp, C.H., et al. "The generalized correlation method for estimation of time delay," IEEE Trans. ASSP (1976).

(Lebart'01) Lebart, K., et al. "A new method based on spectral subtraction for speech Dereverberation," Acta Acoust (2001).

(Lim'79) Lim, J.S., et al. "Enhancement and bandwidth compression of noisy speech," Proc. IEEE (1979).

(Mandel'10) Mandel, M.I., et al. "Model-based expectation maximization source separation and localization," IEEE Trans. ASLP (2010).

(Nakatani'10) Nakatani, T., et al. "Speech Dereverberation based on variance-normalized delayed linear prediction," IEEE Trans. ASLP (2010).

(Narayanan'13) Narayanan, A., et al. "Ideal ratio mask estimation using deep neural networks for robust speech recognition," Proc. ICASSP (2013).

(Ozerov'10) Ozerov, A., et al. "Multichannel Nonnegative Matrix Factorization in Convolutive Mixtures for Audio Source Separation," in IEEE Trans. ASLP (2010).

(Sawada'04) Sawada, H., et al. "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," IEEE Trans. SAP (2004).

(Souden'13) Souden, M., et al. "A multichannel MMSE-based framework for speech source separation and noise reduction," IEEE Trans. ASLP (2013).

(Tachioka'14) Tachioka, Y., et al. "Dual system combination approach for various reverberant environments with dereverberation techniques," Proc. REVERB (2014).

(Van Trees'02) Van Trees, H.L. "Detection, estimation, and modulation theory. Part IV. , Optimum array processing," Wiley-Interscience, New York (2002).

(Van Veen'88) Van Veen, B.D., et al. "Beamforming: A versatile approach to spatial filtering," IEEE ASSP (1988).

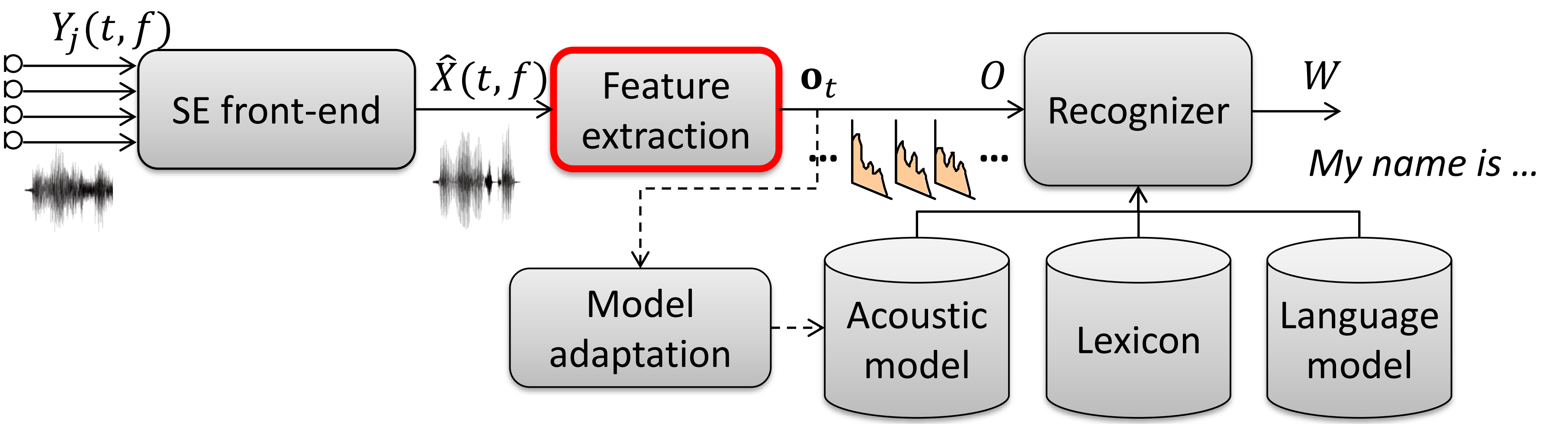
(Virtanen'07) Virtanen, T. "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," IEEE Trans. ASLP (2007).

References (SE-Front-end 3/3)

(Wang’06)	Wang, D. L., & Brown, G. J. (Eds.). “Computational auditory scene analysis: Principles, algorithms, and applications,” Hoboken, NJ: Wiley/IEEE Press (2006).
(Wang’16)	Wang, Z.-Q., et al., “A Joint Training Framework for Robust automatic speech recognition,” IEEE/ACM Trans. ASLP (2016).
(Waritz’07)	Warsitz, E., et al. “Blind acoustic beamforming based on generalized eigenvalue decomposition,” IEEE Trans. ASLP (2007).
(Weninger’14)	Weninger, F., et al. “The MERL/MELCO/TUM system for the REVERB challenge using deep recurrent neural network feature enhancement,” Proc. REVERB (2014).
(Weninger ’15)	Weninger, F., et al. “Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR,” Proc. LVA/ICA (2015).
(Xiao’16)	Xiao, X., et al. “Deep beamforming networks for multi-channel speech recognition,” Proc. ICASSP (2016).
(Xu’15)	Xu, Y., et al. “A regression approach to speech enhancement based on deep neural networks,” IEEE/ACM Trans. ASLP (2015).
(Yoshioka’12)	Yoshioka, T., et al. “Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening,” IEEE Trans. ASLP (2012).
(Yoshioka’12b)	Yoshioka, T., , et al. “Making machines understand us in reverberant rooms: robustness against reverberation for automatic speech recognition,” IEEE Signal Process. Mag. (2012).
(Yoshioka’15)	Yoshioka, T., et al. “The NTT CHiME-3 system: advances in speech enhancement and recognition for mobile multi-microphone devices,” Proc. ASRU (2015).

3. Back-end techniques for distant ASR

3.1 Feature extraction

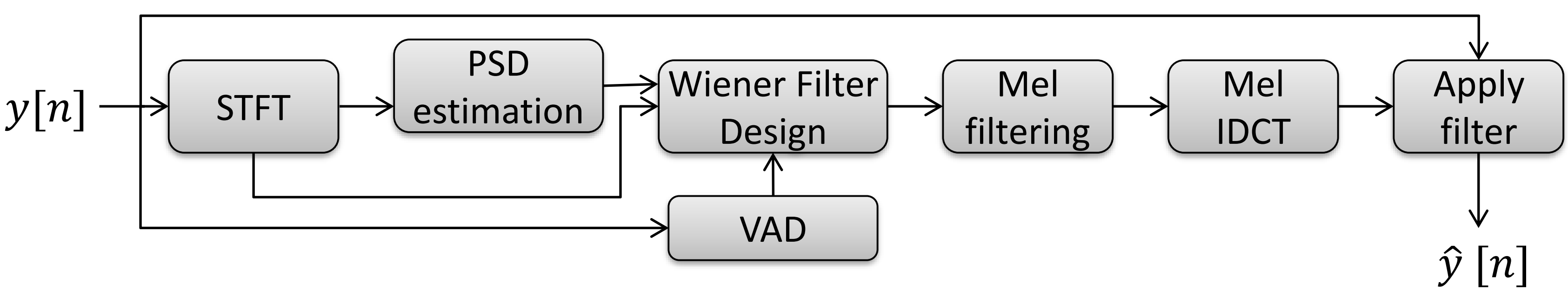


Feature extraction

- Log mel filterbank

  - Spectrum analysis
  - Power extraction (disregard phase)
  - Emphasize low-frequency power with perceptual knowledge (Mel scale)
  - Dynamic range control
  - Cepstrum Mean and Variance Normalization (CMVN)

- ETSI Advanced front-end (ETSI707)



  - Developed at the Aurora project
  - Time domain Wiener-filtering (WF) based noise reduction

Gammatone Filtering based features

- Human auditory system motivated filter
- Power-Normalized Cepstral Coefficients (PNCC) (Kim'12)

  - Replace $\log(\cdot)$ to power $(\cdot)^{0.1}$, frequency-domain Gammatone filtering, Medium-duration Power bias subtraction

- Time-domain Gammatone filtering (e.g., Schulter'09, Mitra'14)
  - Can combine amplitude modulation based features
  - Gammatone filtering and amplitude modulation based features (Damped Oscillator Coefficients (DOC), Modulation of Medium Duration Speech Amplitudes (MMeDuSA)) showed significant improvement for CHiME3 task

	MFCC	**DOC**	**MMeDuSA**
CHiME 3 Real Eval (MVDR enhanced signal)	8.83	5.91	6.62

(Hori'15)

(Linear) Feature transformation

- **Linear Discriminant Analysis (LDA)**
  - Concatenate contiguous features, i.e., $\mathbf{x}_\mathrm{t} = [\mathbf{o}_{t-L}^T, \dots, \mathbf{o}_t,^T \dots, \mathbf{o}_{t+L}^T]^T$
  - $\hat{\mathbf{o}}_t^{\mathrm{LDA}} = \mathbf{A}^{\mathrm{LDA}}\mathbf{x}_t$
  - Estimate a transformation to reduce the dimension with discriminant analysis

    → Capture long-term dependency
- **Semi-Tied Covariance (STC)/Maximum Likelihood Linear Transformation (MLLT)**
  - $N(\mathbf{o}_t|\boldsymbol{\mu}_{kl}, \boldsymbol{\Sigma}_{kl}^{\mathrm{diag}}) \to N\left(\mathbf{o}_t\middle|\boldsymbol{\mu}_{kl}, \boldsymbol{\Sigma}_{kl}^{\mathrm{full}}\right)$ with the following relationship

$$\boldsymbol{\Sigma}_{kl}^{\mathrm{full}} = \mathbf{A}^{\mathrm{STC}}\boldsymbol{\Sigma}_{kl}^{\mathrm{diag}}\left(\mathbf{A}^{\mathrm{STC}}\right)^T$$

  - Estimate $\mathbf{A}^{\mathrm{STC}}$ by using maximum likelihood
  - During the recognition, we can evaluate the following likelihood function with diagonal covariance and feature transformation

$$N\left(\hat{\mathbf{o}}_t^{\mathrm{STC}}\middle|\mathbf{A}^{\mathrm{STC}}\boldsymbol{\mu}_{kl}, \boldsymbol{\Sigma}_{kl}^{\mathrm{diag}}\right), \text{where } \hat{\mathbf{o}}_t^{\mathrm{STC}} = \mathbf{A}^{\mathrm{STC}}\mathbf{o}_t$$

(Linear) Feature transformation, Cont'd

- **Feature-space Maximum Likelihood Linear Regression (fMLLR)**
  - Affine transformation: $\hat{\mathbf{o}}_t = \mathbf{A}^{\text{fM}}\mathbf{o}_t + \mathbf{b}^{\text{fM}}$
  - Estimate transformation parameter $\mathbf{A}^{\text{fM}}$ and $\mathbf{b}^{\text{fM}}$ with maximum likelihood estimation

$$Q(\mathbf{A}^{\text{fM}}, \mathbf{b}) = \sum_{k,t,l} \gamma_{t,k,l} \left(\log\left|\mathbf{A}^{\text{fM}}\right| + \log N(\mathbf{A}^{\text{fM}}\mathbf{o}_t + \mathbf{b}^{\text{fM}} | \boldsymbol{\mu}_{kl}, \boldsymbol{\Sigma}_{kl})\right)$$

- LDA, STC, fMLLR are cascadely combined, i.e.,

$$\hat{\mathbf{o}}_t = \mathbf{A}^{\text{fM}}(\mathbf{A}^{\text{STC}}(\mathbf{A}^{\text{LDA}}[\mathbf{o}_{t-L}^T, \dots, \mathbf{o}_t,^T \dots, \mathbf{o}_{t+L}^T]^T)) + \mathbf{b}^{\text{fM}}$$

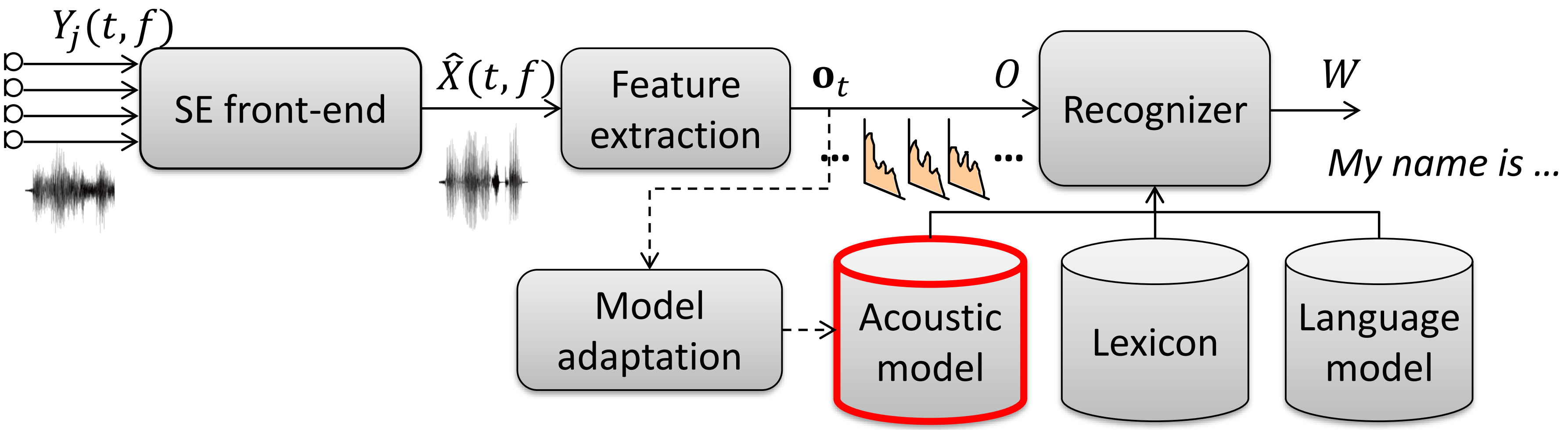
- Effect of feature transformation with distant ASR scenarios GMM

	MFCC, Δ, ΔΔ	LDA, STC, fMLLR
CHiME-2	44.04	33.71
REVERB	39.56	30.88

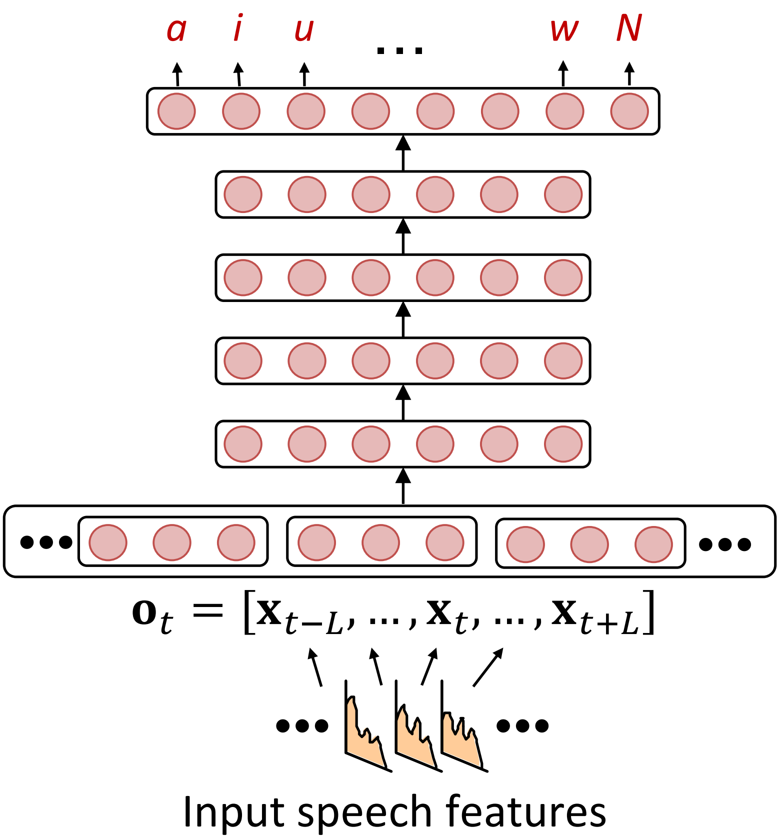
(Tachioka'13,'14)

  - LDA, STC, and fMLLR are cascadely used, and yield significant improvement
  - All are based on **GMM-HMM**, but still applicable to DNN as feature extraction
  - MFCC is more appropriate than Filterbank feature, as MFCC matches GMM

3.2 Robust acoustic models



DNN acoustic model



- Non-linear transformation of (**long**) context features by concatenating contiguous frames

→ Very powerful for noise robust ASR

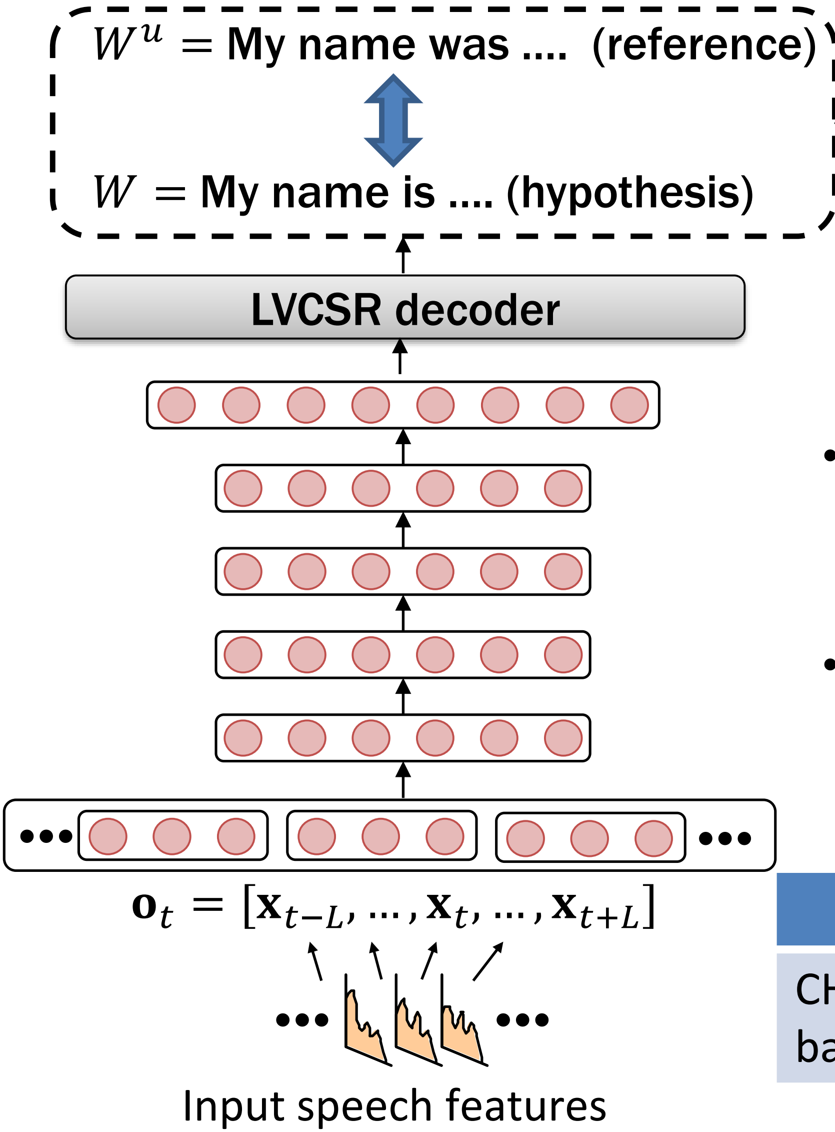
Long context!
(usually 11 frames)

- Cross entropy criterion $J^{\mathrm{ce}}(\theta)$

$$J^{\mathrm{ce}}(\theta) = -\sum_t \sum_k \tau_{t,k} \log h^L_{t,k}(\theta)$$

- There are several other criteria

Sequence discriminative criterion



Compute sequence level errors

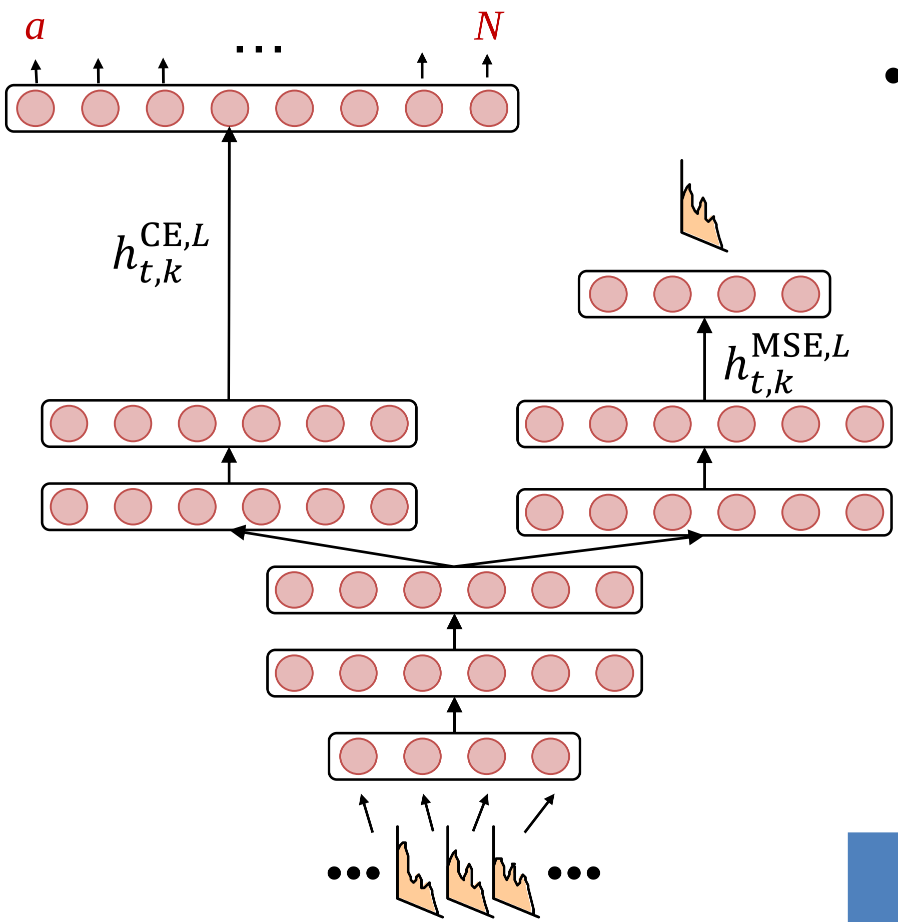
- Sequence discriminative criterion $J^{\text{seq}}(\theta)$

$$J^{\text{seq}}(\theta) = \sum_u \sum_W E(W, W^u) p(W|O^u)$$

- $E(W, W^u)$ is a sequence error between reference W^u and hypothesis W
  – State-level Minimum Bayes Risk (sMBR)

	GMM	DNN CE	DNN sMBR
CHiME3 baseline v2	23.06	17.89	15.88

Multi-task objectives



- Use both MMSE and CE criteria
  - X as clean speech target
  - T as transcription

$$J(\theta) = \rho J^{\mathrm{CE}}(T;\theta) + (1-\rho) J^{\mathrm{MSE}}(X;\theta)$$
$$= -\rho \sum_{t,k} \tau_{t,k} \log h_{t,k}^{\mathrm{CE},L} + (1-\rho) \sum_{t,d} \left| x_{t,d} - h_{t,d}^{\mathrm{MSE},L} \right|^2$$

  - Network tries to solve both enhancement and recognition
  - ρ controls the balance between the two criteria

(Giri'15)

	CE	Multi-task $\rho = 0.91$
REVERB RealData	32.12	31.97

Toward further long context

Time Delayed Neural Network (TDNN)

Convolutional Neural Network (CNN)

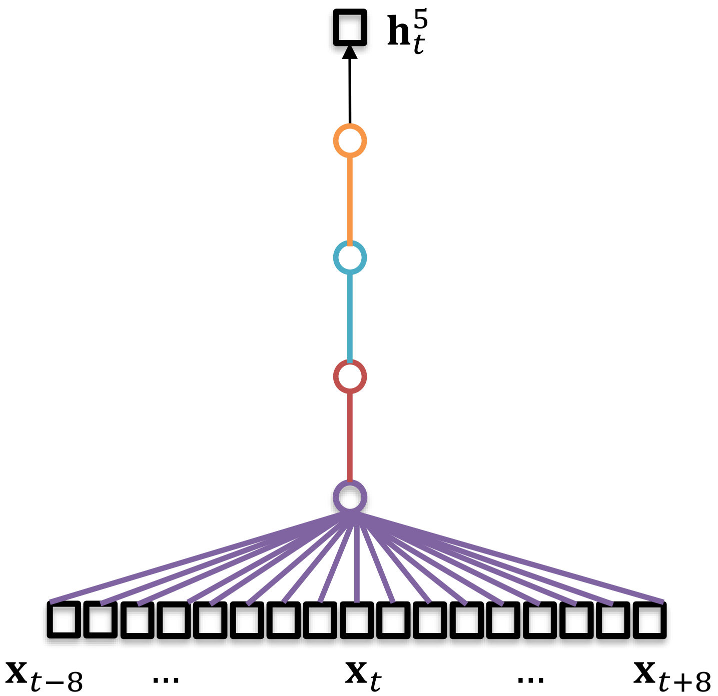
Recurrent Neural Network (RNN)

- Long Short-Term Memory (LSTM)

Time delayed neural network (TDNN)

(Waibel'89, Peddinti'15)

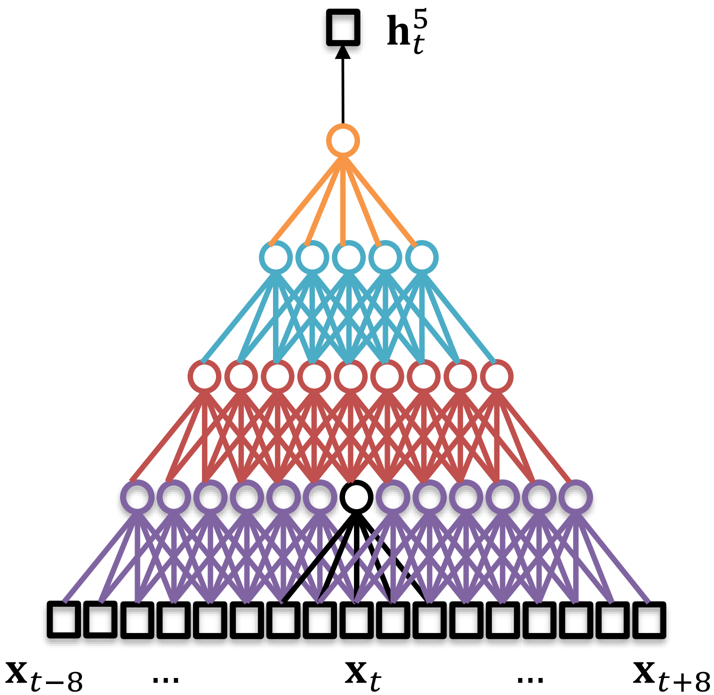
- Deal with "very" long context (e.g., 17 frames)



- Difficult to train the first layer matrix due to vanishing gradient

Time delayed neural network (TDNN)

(Waibel'89, Peddinti'15)



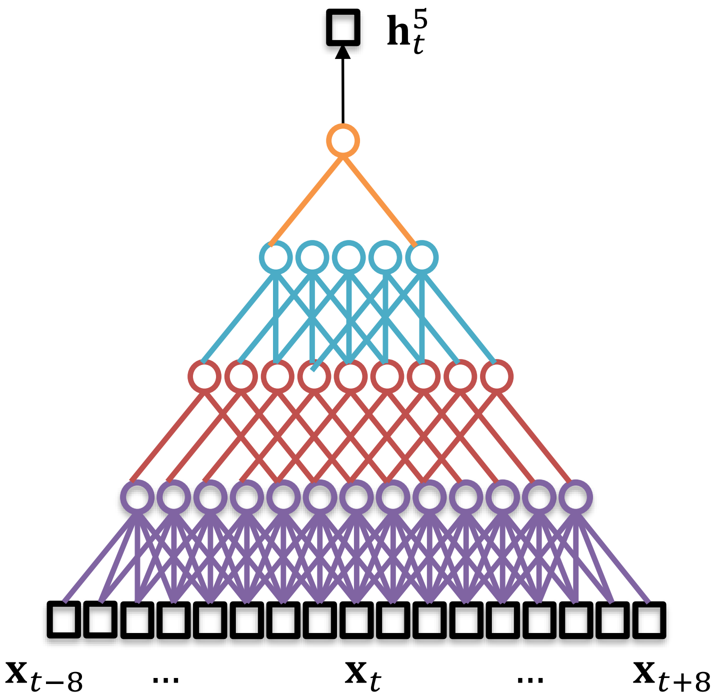
- Original TDNN
  - Consider short context (e.g., [-2, 2]), but expand context at each layer

$$\mathbf{h}_t^1 = \sigma(\mathbf{A}^1[\mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_{t,} \mathbf{x}_{t+1}, \mathbf{x}_{t+2}] + \mathbf{b}^1)$$
$$\mathbf{h}_t^2 = \sigma(\mathbf{A}^2[\mathbf{h}_{t-2}^1, \mathbf{h}_{t-1}^1, \mathbf{h}_{t,}^1 \mathbf{h}_{t+1}^1, \mathbf{h}_{t+2}^1] + \mathbf{b}^2)$$
$$\mathbf{h}_t^3 = \cdots$$

Very large computational cost

Time delayed neural network (TDNN)

(Waibel'89, Peddinti'15)



- Original TDNN
  - Consider short context (e.g., [-2, 2]), but expand context at each layer

$$\mathbf{h}_t^1 = \sigma(\mathbf{A}^1[\mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_{t,} \mathbf{x}_{t+1}, \mathbf{x}_{t+2}] + \mathbf{b}^1)$$
$$\mathbf{h}_t^2 = \sigma(\mathbf{A}^2[\mathbf{h}_{t-2}^1, \mathbf{h}_{t-1}^1, \mathbf{h}_{t,}^1 \mathbf{h}_{t+1}^1, \mathbf{h}_{t+2}^1] + \mathbf{b}^2)$$
$$\mathbf{h}_t^3 = \cdots$$

Very large computational cost

- Subsampled TDNN (Peddinti'15)
  - Subsample frames in the context expansion

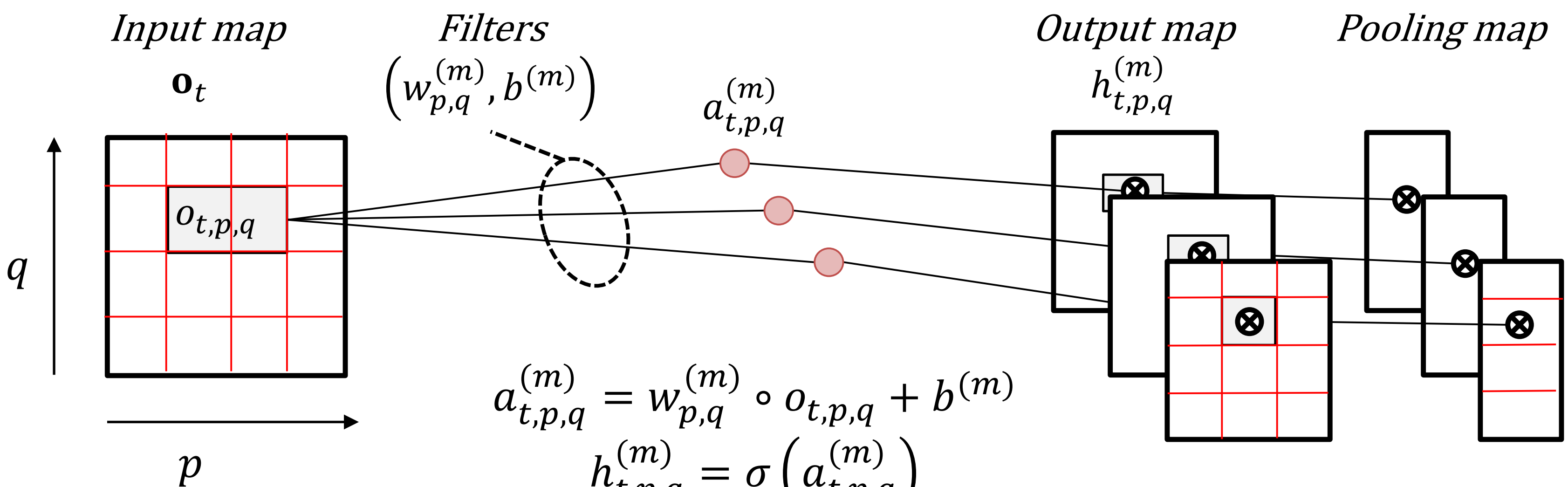
$$\mathbf{h}_t^2 = \sigma(\mathbf{A}^2[\mathbf{h}_{t-2}^1, \mathbf{h}_{t+2}^1] + \mathbf{b}^2)$$

  - Efficiently compute long context network

	DNN	TDNN
ASpIRE	33.1	30.8
AMI	53.4	50.7

Convolutional Neural Network (CNN)

- Represents the input as time-frequency feature map $o_{t,p,q}$ (we can also use multiple maps one for static, delta and delta-delta features), where p and q are indexes along the time and frequency axes of the feature maps



$$a^{(m)}_{t,p,q} = w^{(m)}_{p,q} \circ o_{t,p,q} + b^{(m)}$$

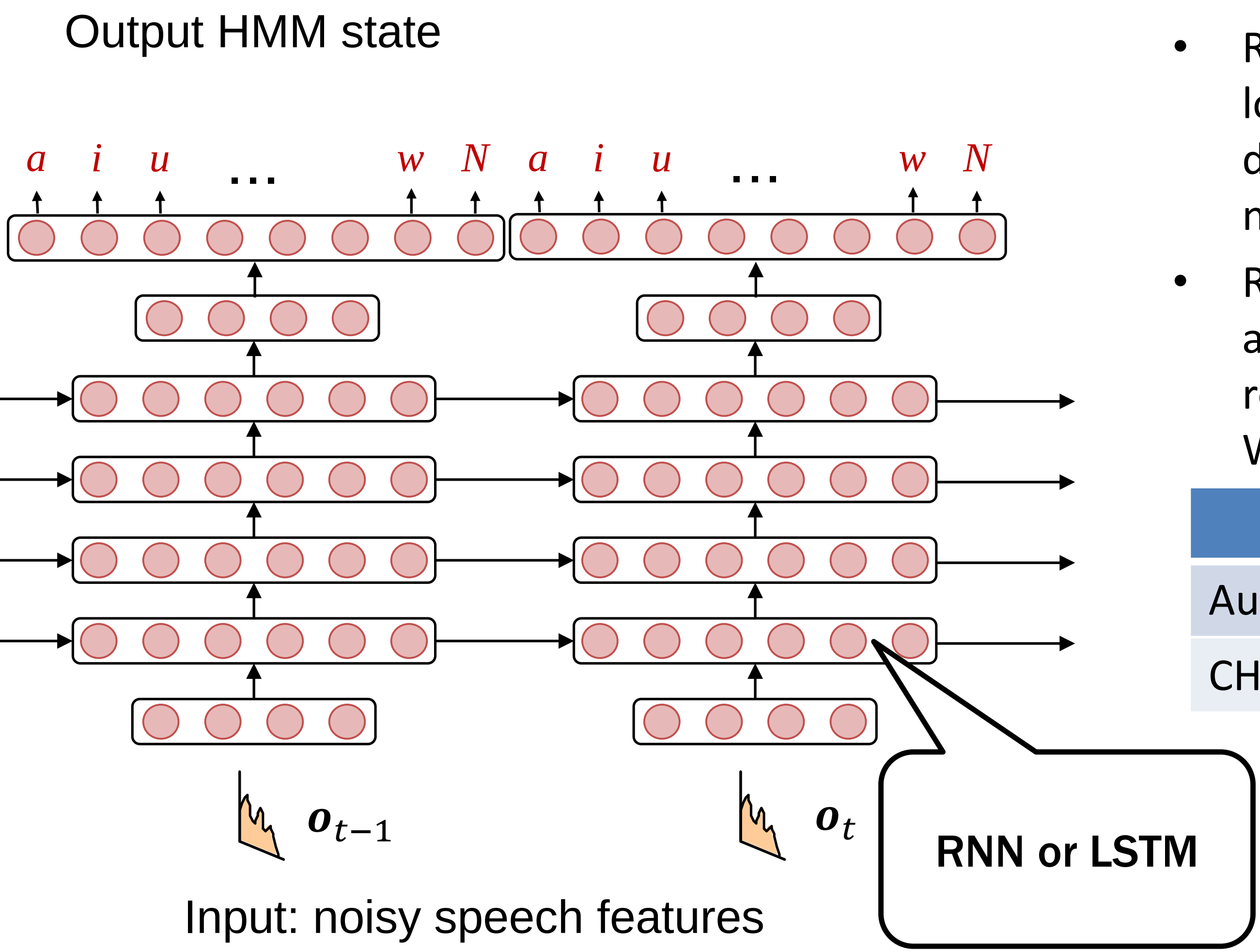
$$h^{(m)}_{t,p,q} = \sigma\left(a^{(m)}_{t,p,q}\right)$$

$$\hat{h}^{(m)}_{t,q} = \max_{p} h^{(m)}_{t,p,q}$$ **(max pooling)**

- Time-dimensional feature maps can capture long context information

REVERB: **23.5** (DNN) → **22.4** (CNN-DNN) (Yoshioka'15a)

RNN/LSTM acoustic model



- RNN can alos capture the long-term distortion effect due to reverberation and noise
- RNN/LSTM can be applied as an acoustic model for noise robust ASR (Weng'14, Weninger'14)

	DNN	RNN
Aurora4	13.33	12.74
CHiME2	29.89	27.70

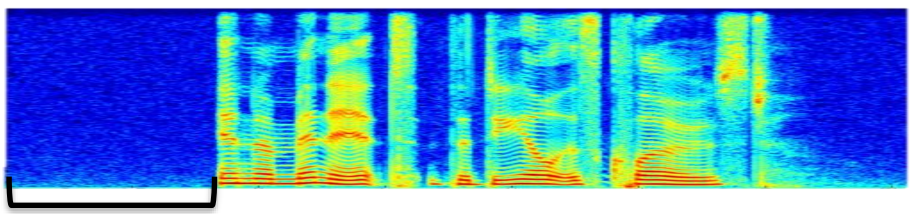
Practical issues

The importance of the alignments

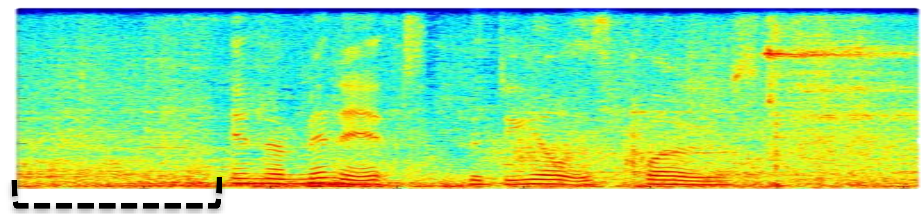
- DNN CE training needs frame-level label $\tau_{t,k}$ obtained by Viterbi algorithm

$$J^{\mathrm{CE}}(\theta) = -\sum_t \sum_k \tau_{t,k} \log h_{t,k}^L$$

- However, it is very difficult to obtain precise label $\tau_{t,k}$ for noisy speech



sil



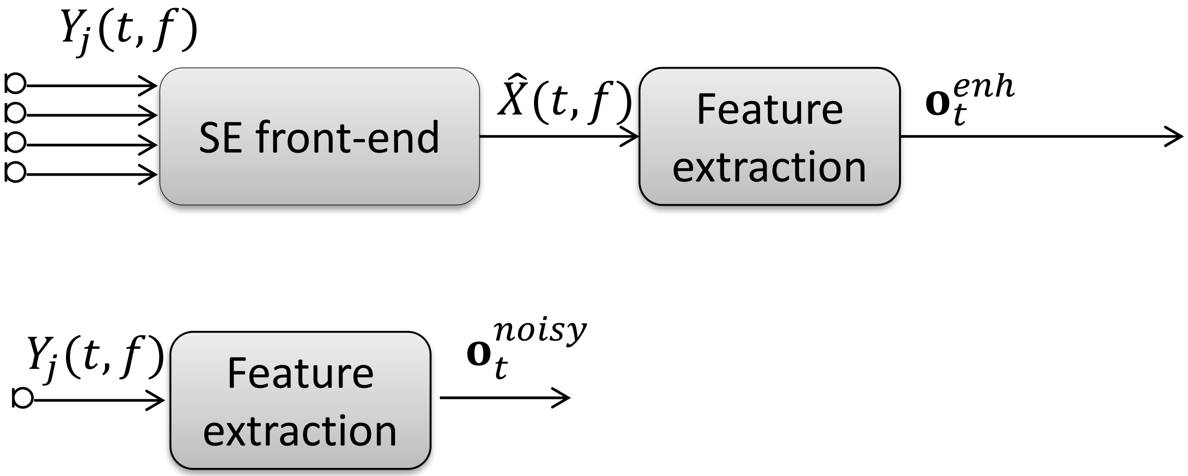
sil?

- How to deal with the issue?
  - Re-alignment after we obtain DNN several times
  - Sequence discriminative training can mitigate this issue (however, since we use CE as an initial model, it is difficult to recover this degradation)
  - Parallel clean data alignment if available

	Noisy alignment	Clean alignment
CHiME2	29.89	24.75

(Weng'14)

Degradation due to enhanced features

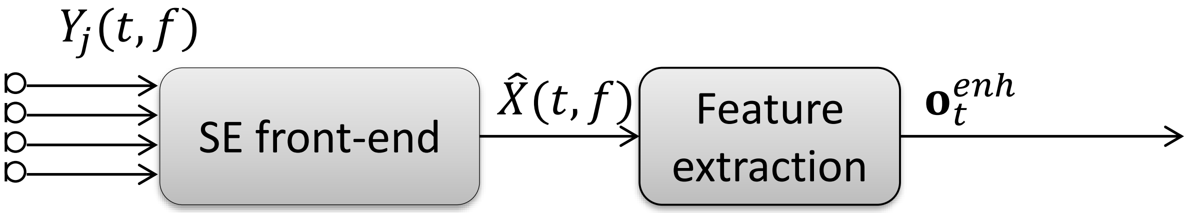


- Which features we should use for training acoustic models?
  - Noisy features: $\mathbf{o}_t^{noisy} = \mathrm{FE}(Y)$
  - Enhanced features: $\mathbf{o}_t^{enh} = \mathrm{FE}(\hat{X})$

CHiME 3 Real Eval

Training	Testing	WER (%)
Noisy $\mathbf{o}_t^{noisy}$	Noisy $\mathbf{o}_t^{noisy}$	23.66
Noisy $\mathbf{o}_t^{noisy}$	Enhanced $\mathbf{o}_t^{enh}$	14.86
Enhanced $\mathbf{o}_t^{enh}$	Enhanced $\mathbf{o}_t^{enh}$	????

Degradation due to enhanced features



- Which features we should use for training acoustic models?
  - Noisy features: $\mathbf{o}_t^{noisy} = \mathrm{FE}(Y)$
  - Enhanced features: $\mathbf{o}_t^{enh} = \mathrm{FE}(\hat{X})$

CHiME 3 Real Eval

Training	Testing	WER (%)
Noisy $\mathbf{o}_t^{noisy}$	Noisy $\mathbf{o}_t^{noisy}$	23.66
Noisy $\mathbf{o}_t^{noisy}$	Enhanced $\mathbf{o}_t^{enh}$	14.86
Enhanced $\mathbf{o}_t^{enh}$	Enhanced $\mathbf{o}_t^{enh}$	16.17

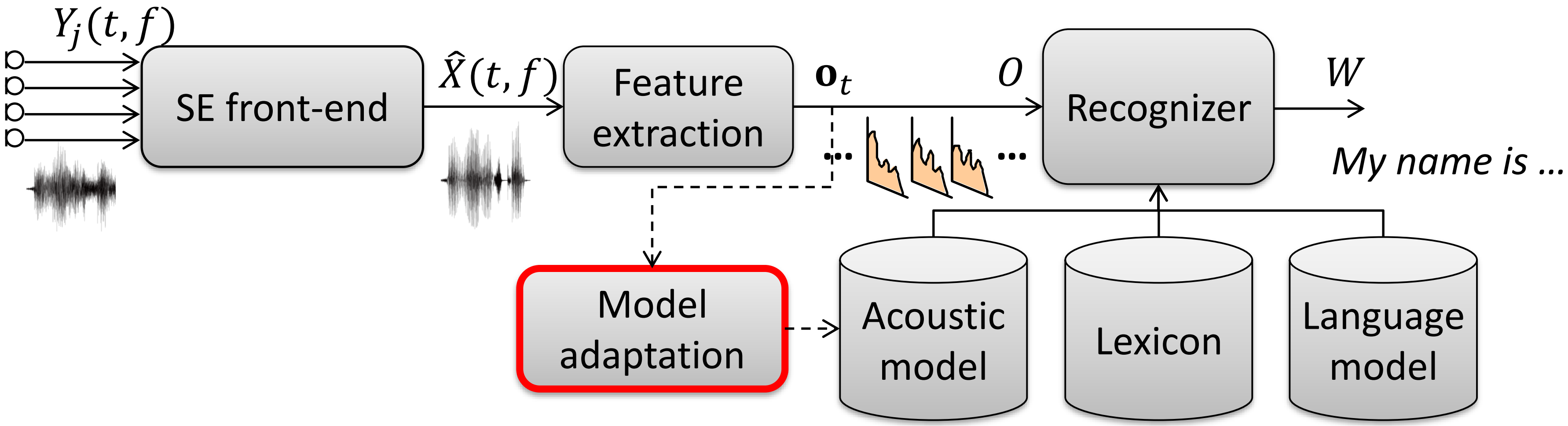
Re-training with enhanced features degrades the ASR performance!!

- Noisy data training are robust for distorted speech (?)

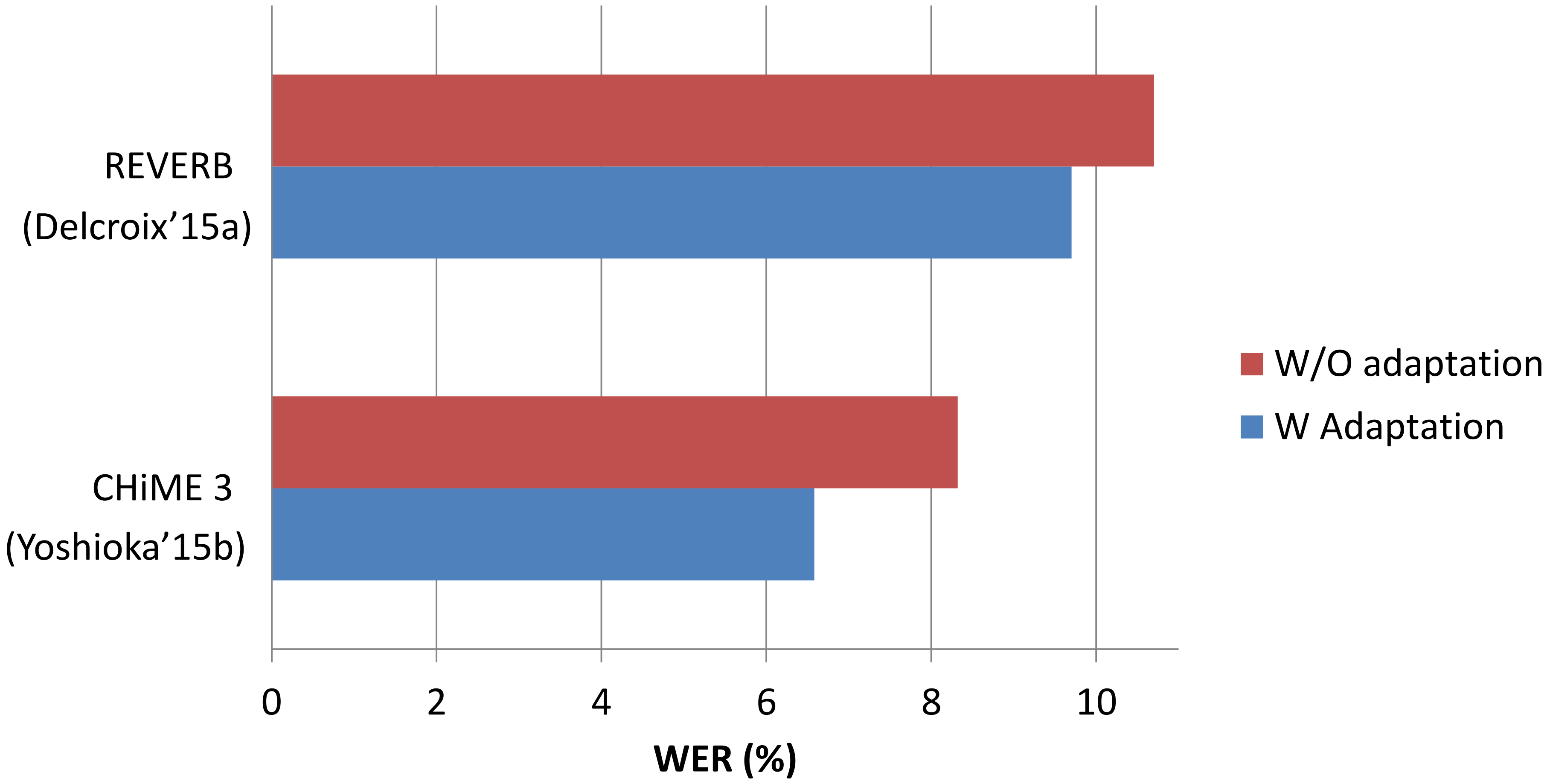
Remarks

- Noise robust feature and linear feature transformation are effective
  - Effective for both GMM and DNN acoustic modeling
- Deep learning is effective for noise robust ASR
  - DNN with sequence discriminative training is still powerful
  - RNN, TDNN, and CNN can capture the long-term dependency of speech, and are more effective when dealing with reverberation and complex noise
- We can basically use standard acoustic modeling techniques even for distant ASR scenarios
- However, need special cares for
  - Alignments
  - Re-training with enhanced features

3.3 Acoustic model adaptation

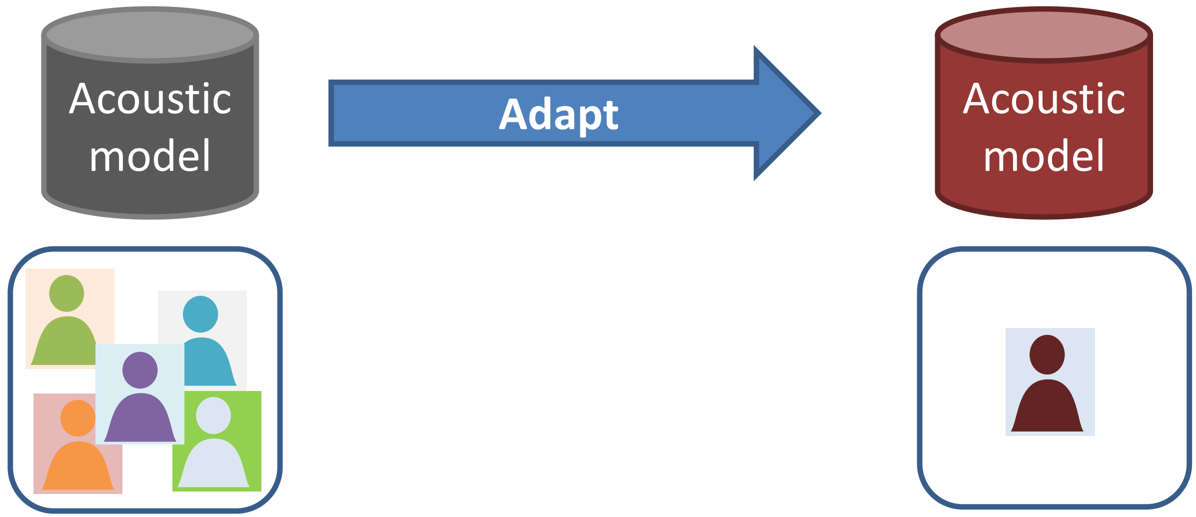


Importance of acoustic model adaptation



Acoustic model adaptation

- DNN is very powerful so why do we need adaptation?



Training data
Various speakers & environments

Testing data
Specific speaker & environment

  - Unseen test condition due to limited amount of training data
  - Model trained on large amount of data may be good on average but not optimal for a specific condition

Supervised/Unsupervised adaptation

- Supervised adaptation
  - *We know what was spoken*
  - There are transcriptions associated with adaptation data

- Unsupervised adaptation
  - *We do not know what was spoken*
  - There are no transcriptions

Supervised/Unsupervised adaptation

- Supervised adaptation
  - *We know what was spoken*
  - There are transcriptions associated with adaptation data

- **Unsupervised adaptation**
  - ***We do not know what was spoken***
  - **There are no transcriptions**

DNN adaptation techniques

- Model adaptation
  - Retraining
  - Linear transformation of input or hidden layers (fDLR, LIN, LHN, LHUC)
  - Adaptive training (Cluster/Speaker adaptive training)

- Auxiliary features
  - Auxiliary features
    - Noise aware training
    - Speaker aware training
    - Context adaptive DNN

DNN adaptation techniques

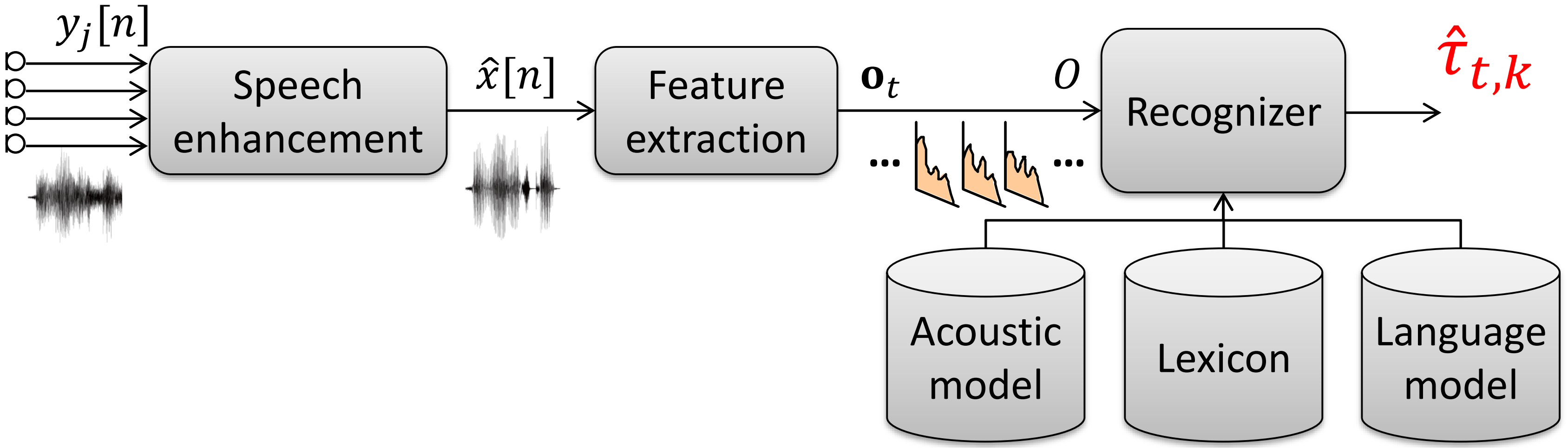
- Model adaptation
  - Retraining
  - Linear transformation of input or hidden layers (fDLR, LIN, LHN, LHUC)
  - Adaptive training (Cluster/Speaker adaptive training)

- Auxiliary features
  - Auxiliary features
    - Noise aware training
    - Speaker aware training
    - Context adaptive DNN

Unsupervised labels estimation

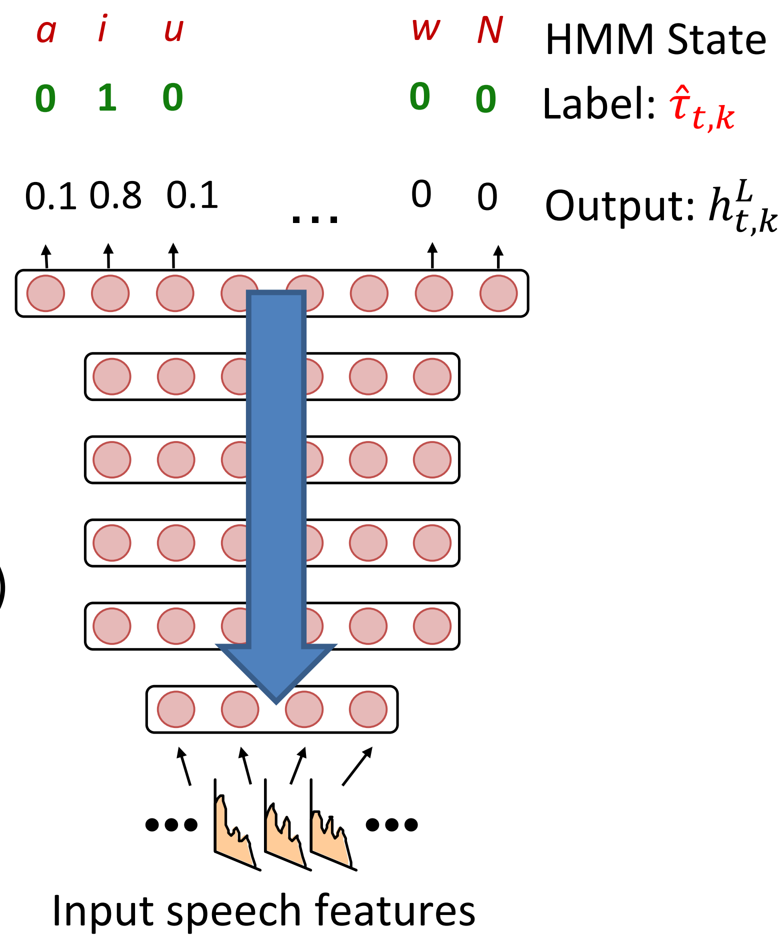
- 1st pass
  - Decode adaptation data with an existing ASR system
  - Obtain estimated labels, $\hat{\tau}_{t,k}$

Adaptation
speech data



Retraining

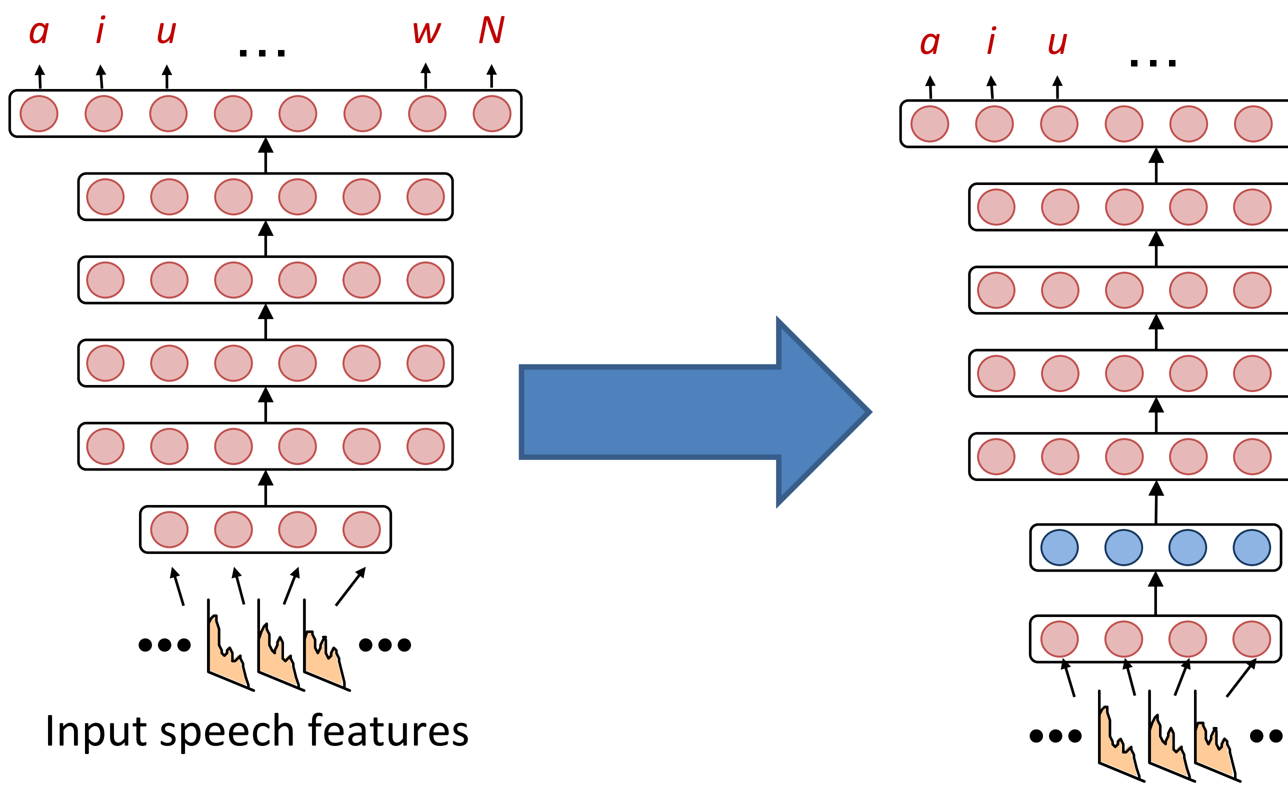
- Retrain/adapt acoustic model parameters given the estimated labels with error backpropagation (Liao'13)
- Prevent modifying too much the model
  - Small learning rate
  - Small number of epochs (early stopping)
  - Regularization (e.g. L2 prior norm (Liao'13), KL (Yu'13))
- For large amount of adaptation data, retraining all or part of the DNN (e.g. lower layers)



Linear input network (LIN)

(Neto'95)

- Add a linear layer that transforms the input features
- Learn the transform with error backpropagation



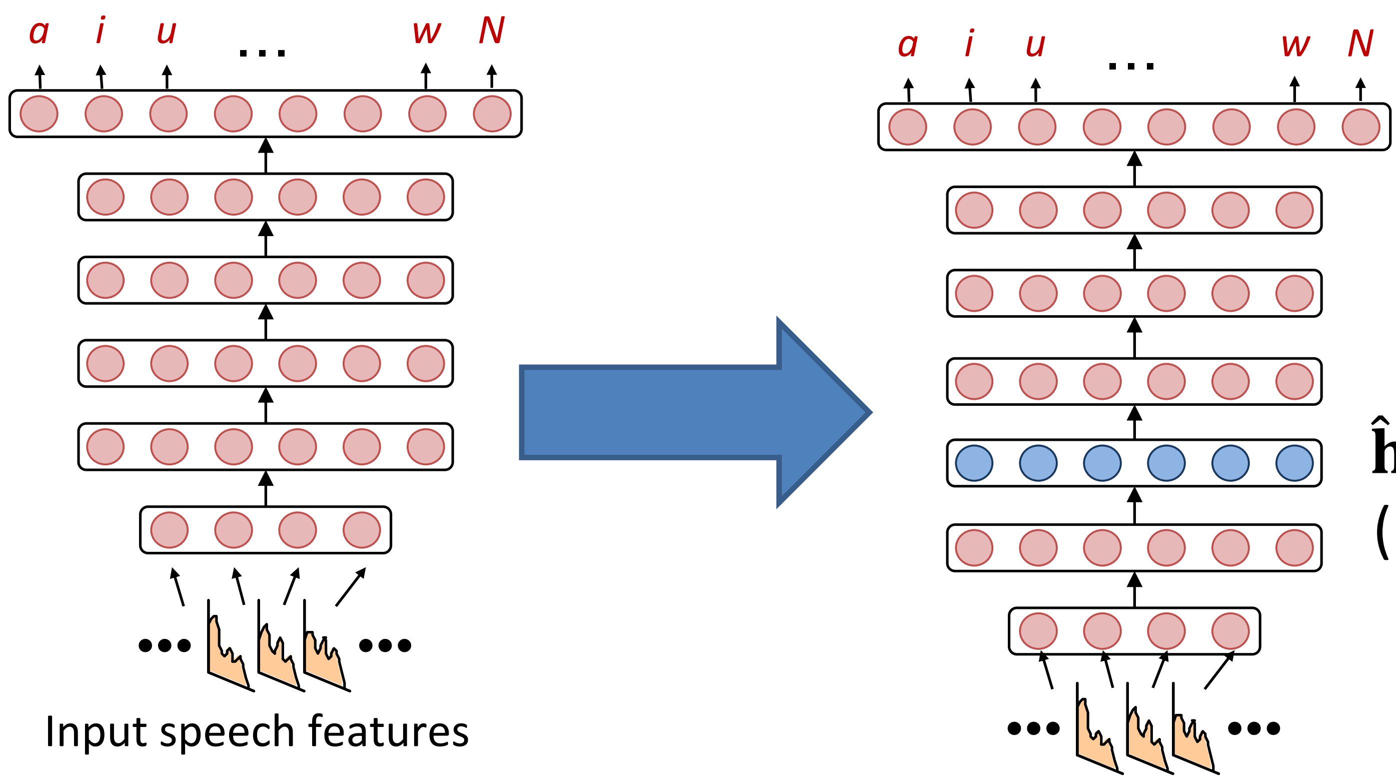
$$\hat{\mathbf{o}}_t = \mathbf{A}\,\mathbf{o}_t + \mathbf{b}$$

(no activation)

Linear hidden network (LHN)

(Gemello'06)

- Insert a linear transformation layer inside the network



$$\hat{\mathbf{h}}_t^l = \mathbf{A}\,\mathbf{h}_t^l + \mathbf{b}$$

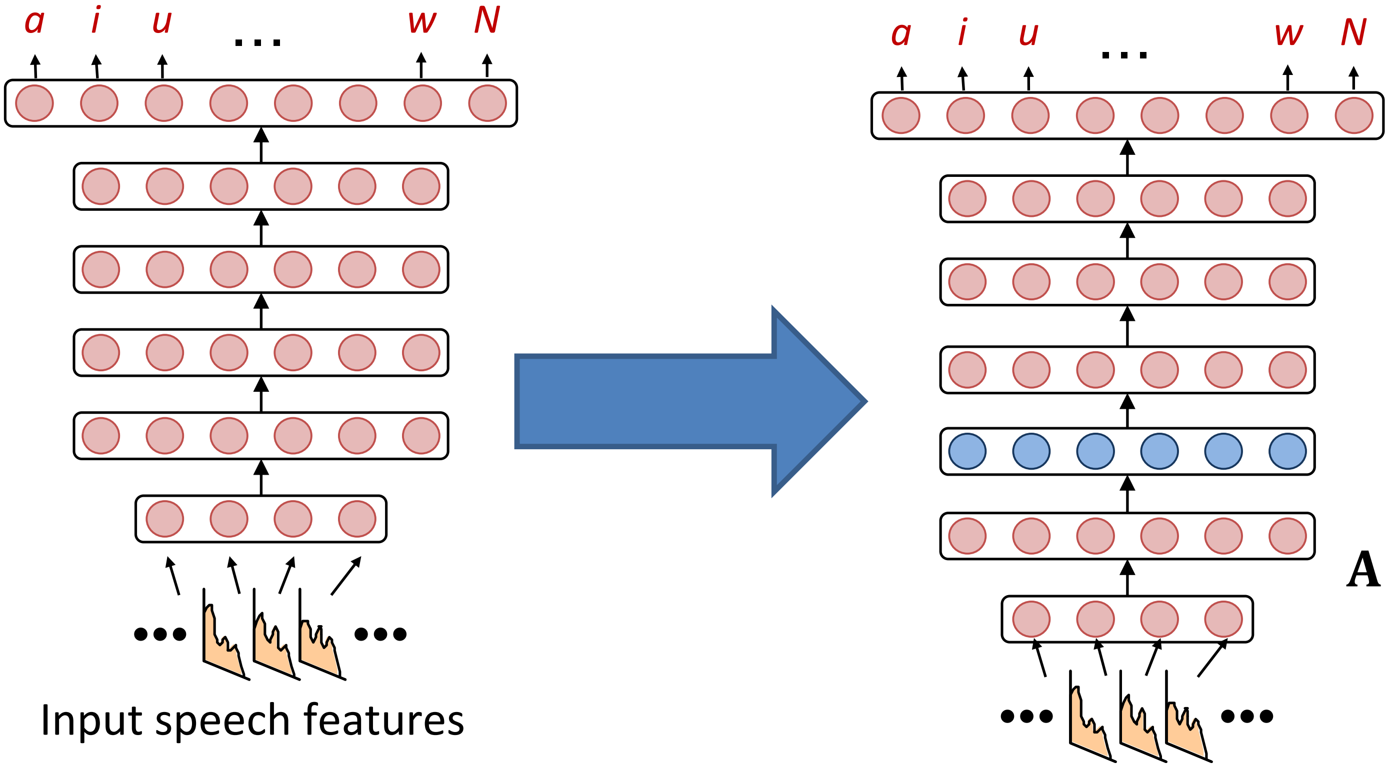
(no activation)

Learning hidden unit contribution (LHUC)

(Swietojanski '14b)

- Similar to LHN but with diagonal matrix

  → Fewer parameters

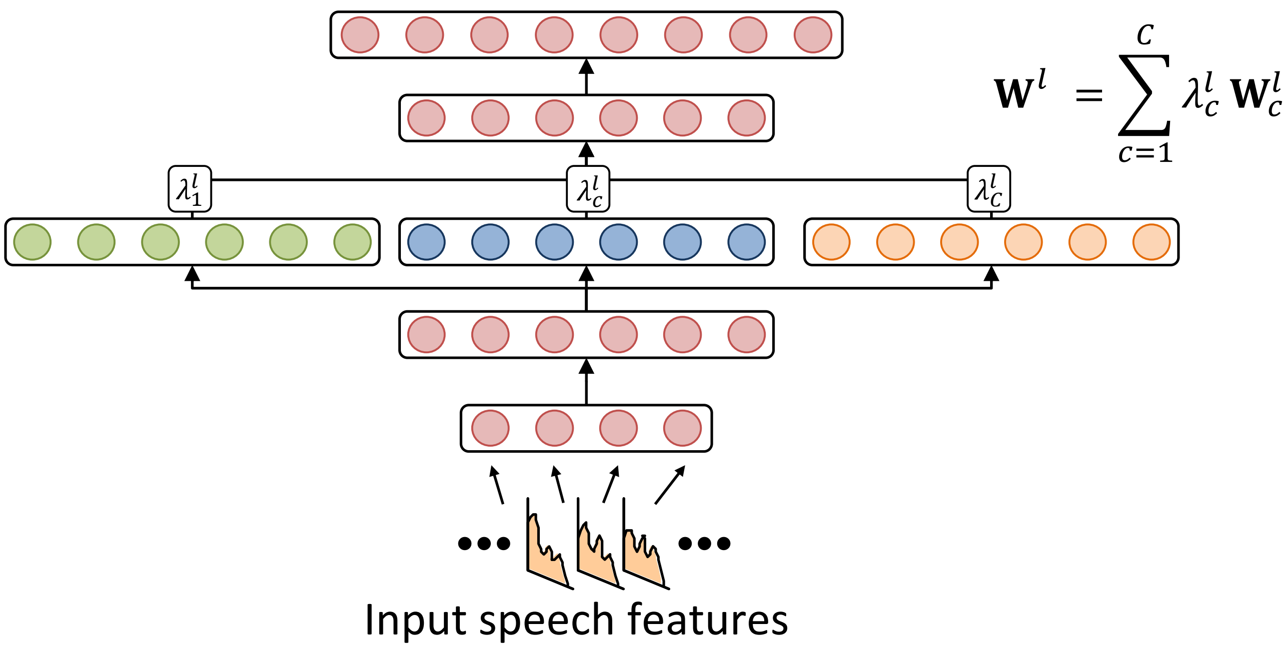


$$\hat{\mathbf{h}}_t^l = \mathbf{A}\,\mathbf{h}_t^l$$

$$\mathbf{A} = \begin{bmatrix} a_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_N \end{bmatrix}$$

Speaker/Cluster adaptive training

- Parameters of one or several layers are made dependent on conditions (speaker or noise)
  - During adaptation, adapt only the parameters of this layer (speaker adaptive training) (Ochiai'14)
  - Use the trained set of parameters as basis ($\mathbf{W}_c^l, c = 1, \dots, C$) and only adapt weights of these basis λ_c^l (Cluster adaptive training) (Tan'15, Chunyang'15)



Room adaptation for REVERB (RealData)

Results from (Delcroix'15a)

Adap	WER (%)
-	24.1
1st	21.7
All	22.1
LIN	22.1

Speech processed with WPE (1ch)
Amount of adaptation data ~9 min
Back-end:
- DNN with 7 hidden layers
- Trigram LM

Model adaptation

☺ Can adapt to conditions unseen during training

☹ Computationally expensive + processing delay

Requires 2 decoding step

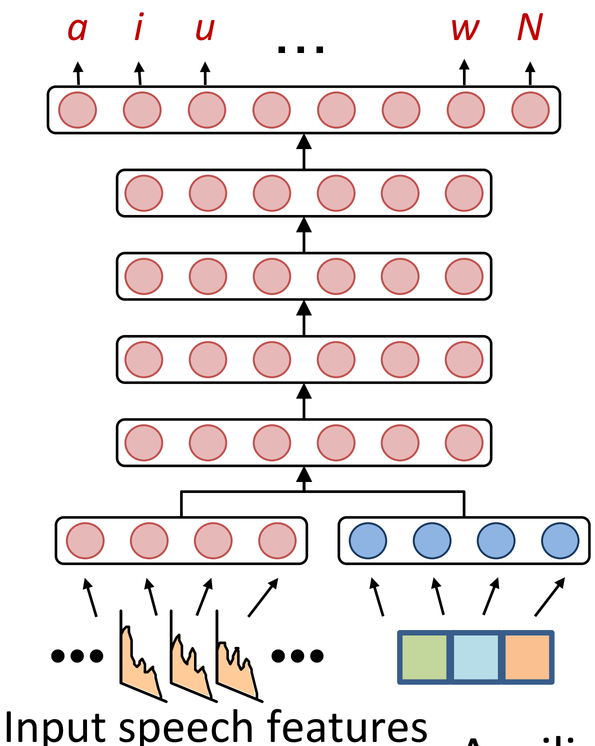
☹ Data demanding

Relatively large amount of adaptation data needed

DNN adaptation techniques

- Model adaptation
  - Retraining
  - Linear transformation of input or hidden layers (fDLR, LIN, LHN, LHUC)
  - Adaptive training (Cluster/Speaker adaptive training)
- Auxiliary features
  - Auxiliary features
    - Noise aware training
    - Speaker aware training
    - Context adaptive DNN

Auxiliary features based adaptation



- Exploit auxiliary information about speaker or noise
- Simple way:
  - Concatenate auxiliary features to input features
- Weights for auxiliary features learned during training

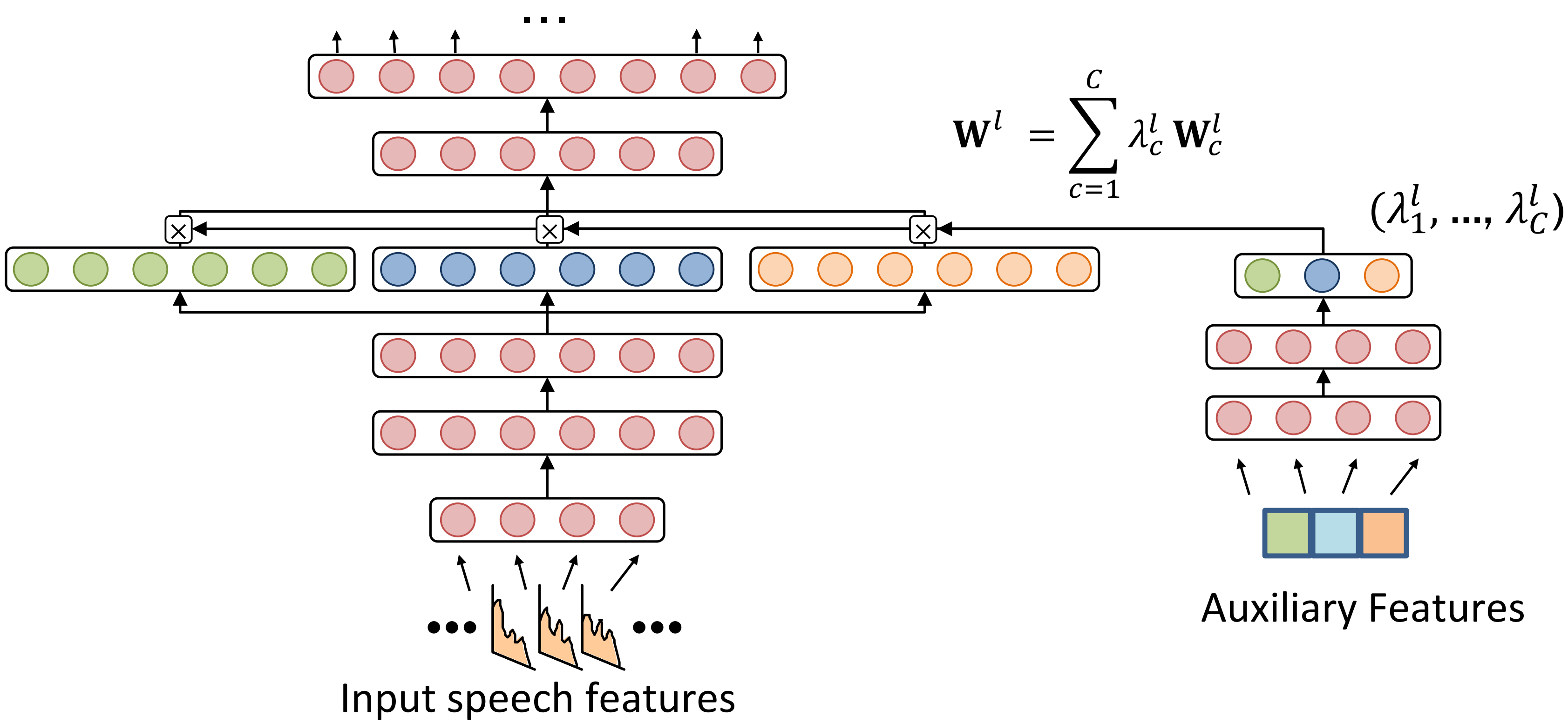
Auxiliary Features represents e.g.,

- Speaker aware (i-vector, Bottleneck feat.) (Saon'13)
- Noise aware (noise estimate) (Seltzer'13)
- Room aware (RT60, Distance, …) (Giri'15)

Context adaptive DNN

(Delcroix'15b, '16a, '16b)

- Similar to cluster adaptive training but the class weights λ_c^l are derived from an auxiliary network that input auxiliary features
- The joint optimization of context classes, class weights and DNN parameters enables class weights and class definitions optimized for ASR



Speaker adaptation

Results from (Kundu’15)

Auxiliary feature	AURORA 4	REVERB
-	9.6 %	20.1 %
i-vector	9.0 %	18.2 %
Speaker ID Bottleneck	9.3 %	17.4 %

- Speaker i-vectors or bottleneck features have shown to improve performance for many tasks
- Other features such as noise or room parameters have also been shown to improve performance

Auxiliary features-based adaptation

☺ Rapid adaptation

Auxiliary features can be computed per utterance (∼10 sec. or less)

☺ Computationally friendly

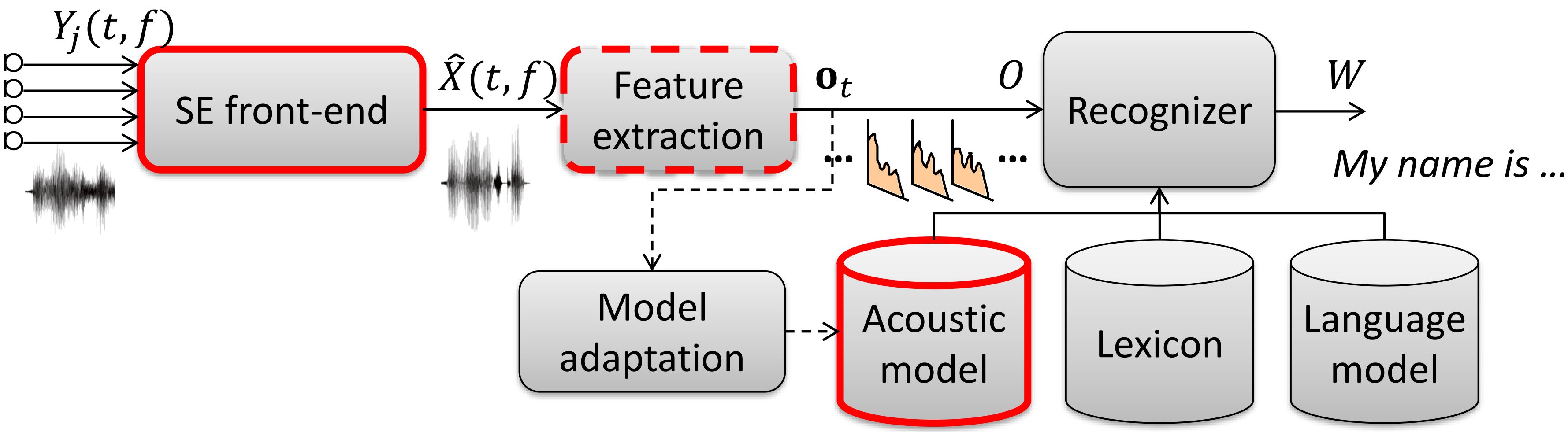
No need for the extra decoding step

(Single pass unsupervised adaptation)

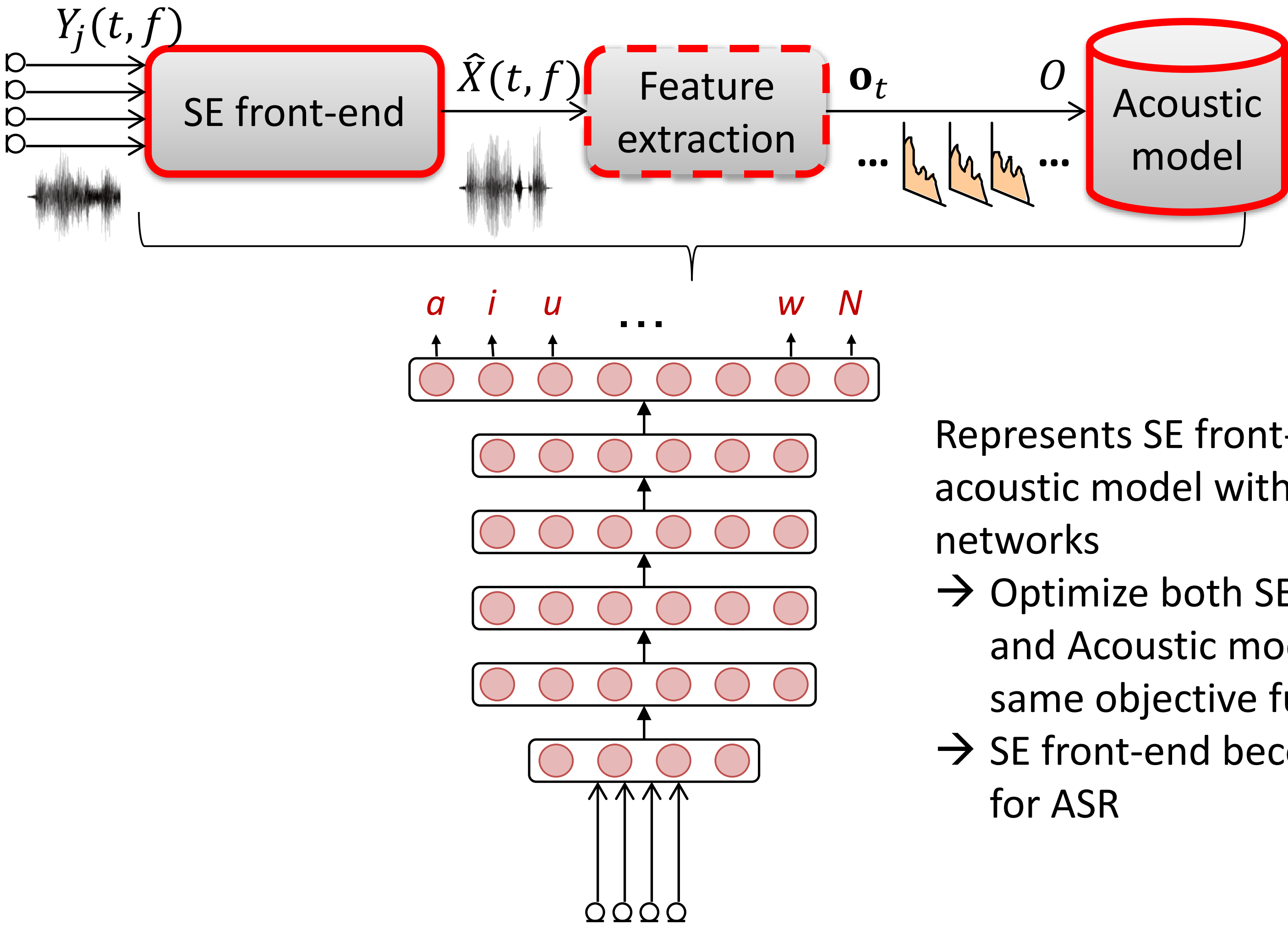
☹ Does not extend to unseen conditions

Requires training data covering all test cases

3.4 Integration of front-end and back-end with deep networks



Front-end and back-end integration

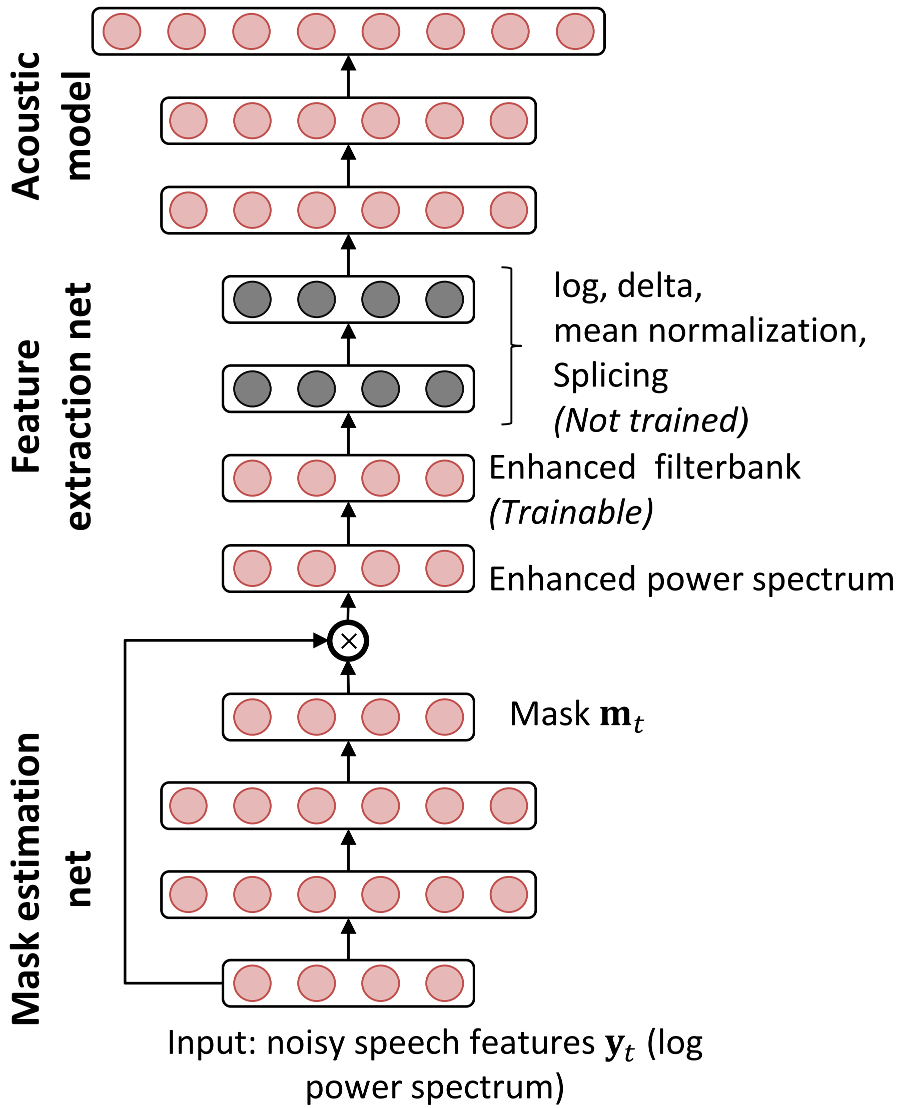


Represents SE front-end and acoustic model with neural networks

→ Optimize both SE front-end and Acoustic model using the same objective function

→ SE front-end becomes optimal for ASR

Single channel integrated system



(Wang'16)

- DNN-based SE front-end and ASR back-end can be connected to form a large network

→ Can be optimized for ASR objective function (Cross entropy or SMBR)

- Initialize each component independently

→ Requires parallel corpus for initialization

Experiments on CHiME 2

Results from (Wang'16)

System	CE	sMBR
Baseline (No SE front-end)	16.2 %	13.9 %
Mask estimation using CE	14.8 %	13.4 %
Mask estimation + retraining	15.5 %	13.9 %
Joint training of mask estimation and acoustic model	**14.0 %**	**12.1 %**
Large DNN-based acoustic model	15.2 %	-

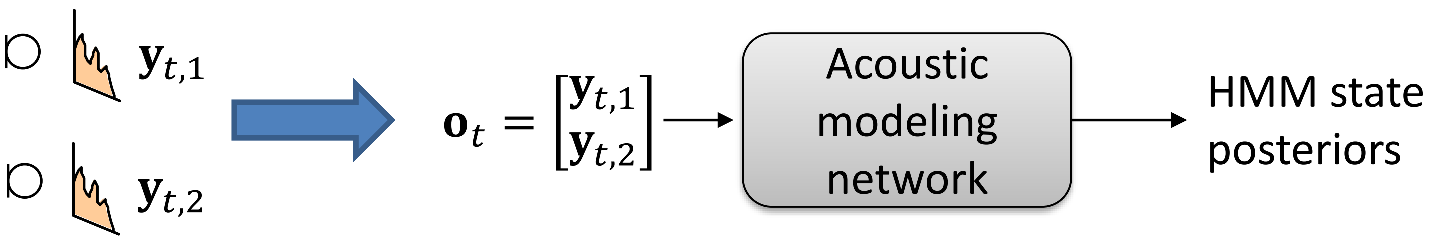
Enhancement DNN
- Predict mask (CE Objective function)
- Features: Log power spectrum

Acoustic model DNN
- Log Mel Filterbanks
- Trained on noisy speech with cross entropy (CE) or sMBR objective function

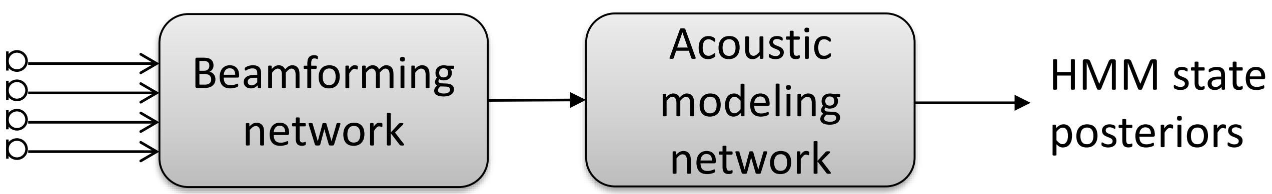
Multi-channel approaches

Multi-channel approaches

- Multi-channel input to the acoustic model



- Beamforming network

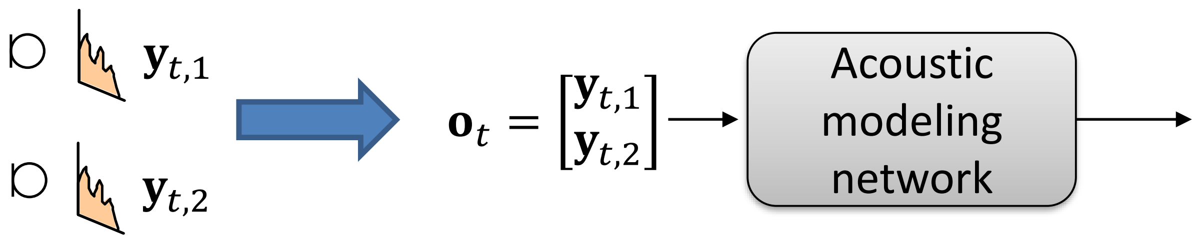


1. Directly enhance signal using CNN-based beamforming network (Filter learning)
2. DNN outputs beamforming filters (Filter prediction)

Multi-channel input acoustic model

(Marino'11, Swietojanski'13 , Liu'14, Swietojanski'14a)

- Concatenate speech features (e.g. log mel filterbank) for each channel at the input of the acoustic model



  – With fully connected networks (Swietojanski'13 , Liu'14)
  – With CNNs (Swietojanski'14a)

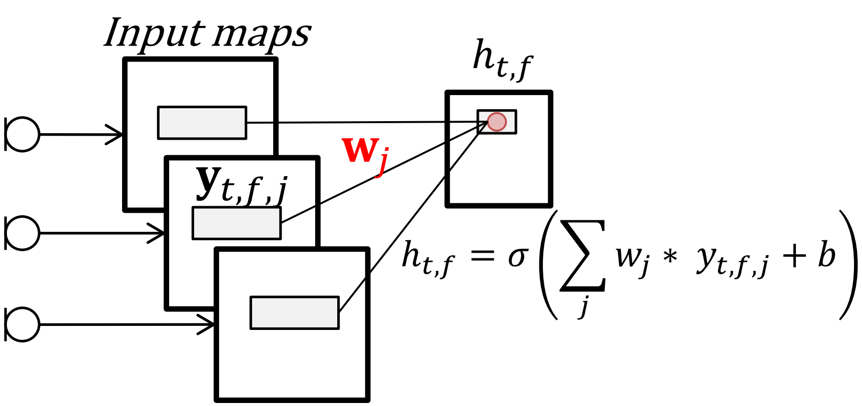
  – Without phase difference: lack of special information

CNN-based multi-channel input (feature domain)

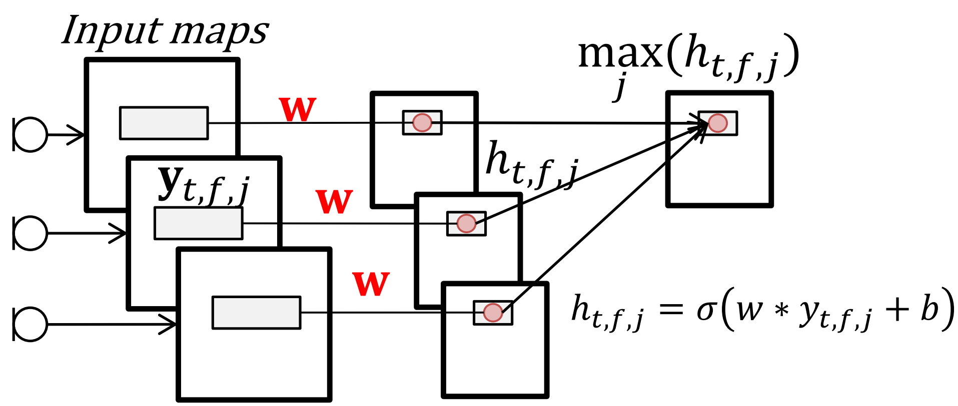
(Swietojanski'14a)

- Each channel considered as a different feature map input to a CNN acoustic model

Conventional CNN



Channel wise convolution



- Process each channel with different filters w_j
- Sum across channels

→ Similar to beamforming but

- Filter shared across time-frequency bins
- Input does not include phase information

Process each channel with same filter w

Max pooling across channels

→ Select the "most reliable" channel for each time-frequency bin

→ Applicable to different microphone configuration

Results for AMI corpus

Results from (Swietojanski’14a)

	DNN	CNN
Single distant mic	53.1 %	51.3 %
Multi-channel input (4ch)	51.2 %	50.4 %
Multi-channel input (4ch) channel-wise convolution	**-**	**49.4 %**
BeamformIt (8ch)	49.5 %	46.8 %

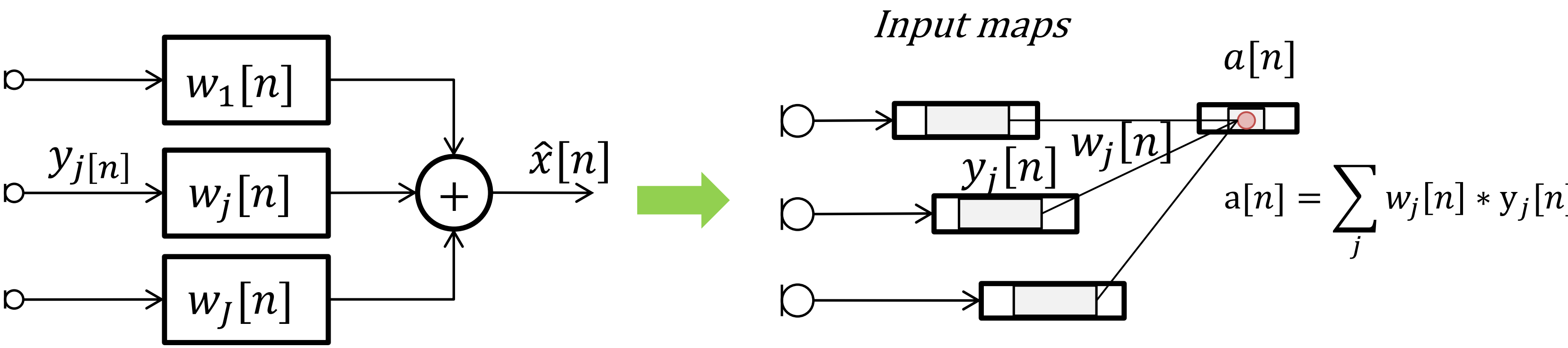
- Inputting multi-channel improves over single-channel input
- Beamforming seems to perform better possibly because it exploits phase difference across channels

Back-end configuration:

- 1 CNN layer followed by 5 fully connected layers
- Input feature 40 log mel filterbank + delta + delta-delta

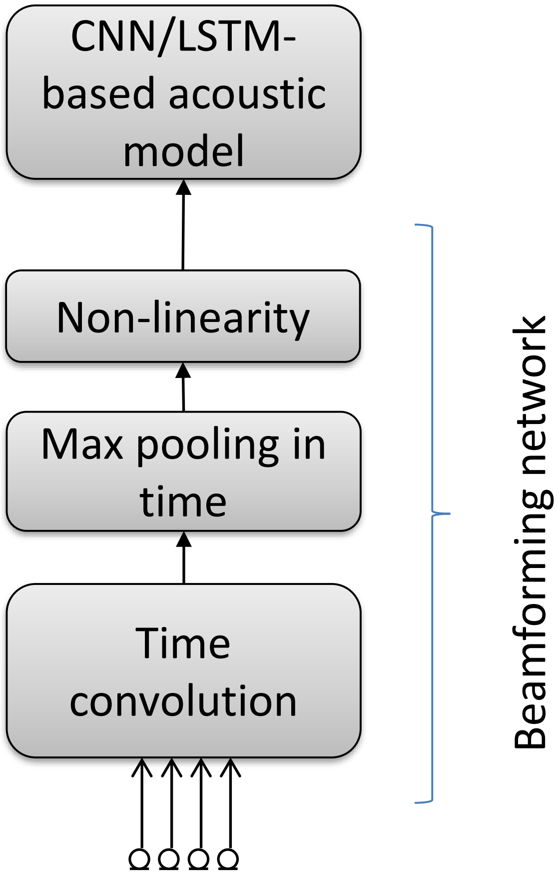
Filter learning-based Beamforming network (time domain) (Hoshen’15, Sainath’16)

- Beamforming can be expressed as a convolutional layer in the time domain (raw signals)



- Joint optimization is possible
  - Time domain → Can exploit phase information
  - **Fixed beamforming filter is learned** from corpus
  - By having multiple output maps, we can obtain a set of fixed beamformers steering at different directions $w_j[n] \rightarrow w_j^{(m)}[n]$

Filter learning-based Beamforming network architecture



- Beamforming and acoustic modeling can be expressed as a single neural network

→ Joint training becomes possible

- Beamforming network
- Performs beamforming + implicit filterbank extraction
- Max pooling in time and non-linearity removes phase information and mimic filterbank extraction

Results on a large corpus

Results from (Sainath'16)

	CE	sMBR
Raw signal (1ch)	23.5 %	19.3 %
Oracle delay and sum (8ch)	22.4 %	18.8 %
Beamforming network (8ch)	20.6 %	17.2 %
8ch log mel input	21.7 %	-

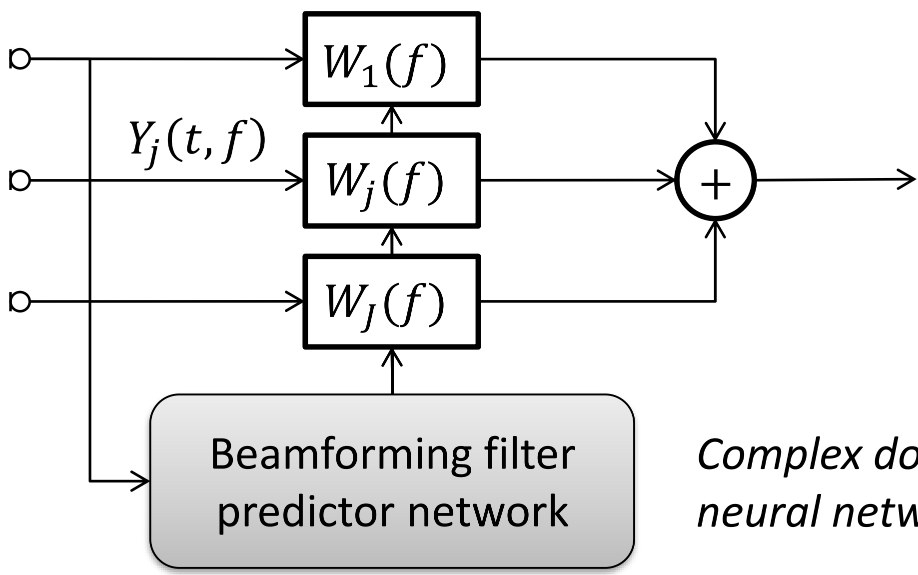
Google internal data
2000 h of training data with simulated distant speech

Filter prediction-based beamforming network

- Design a neural network to **predict** the beamforming filter coefficients given the input microphone signals

→ Adaptive to the input signal

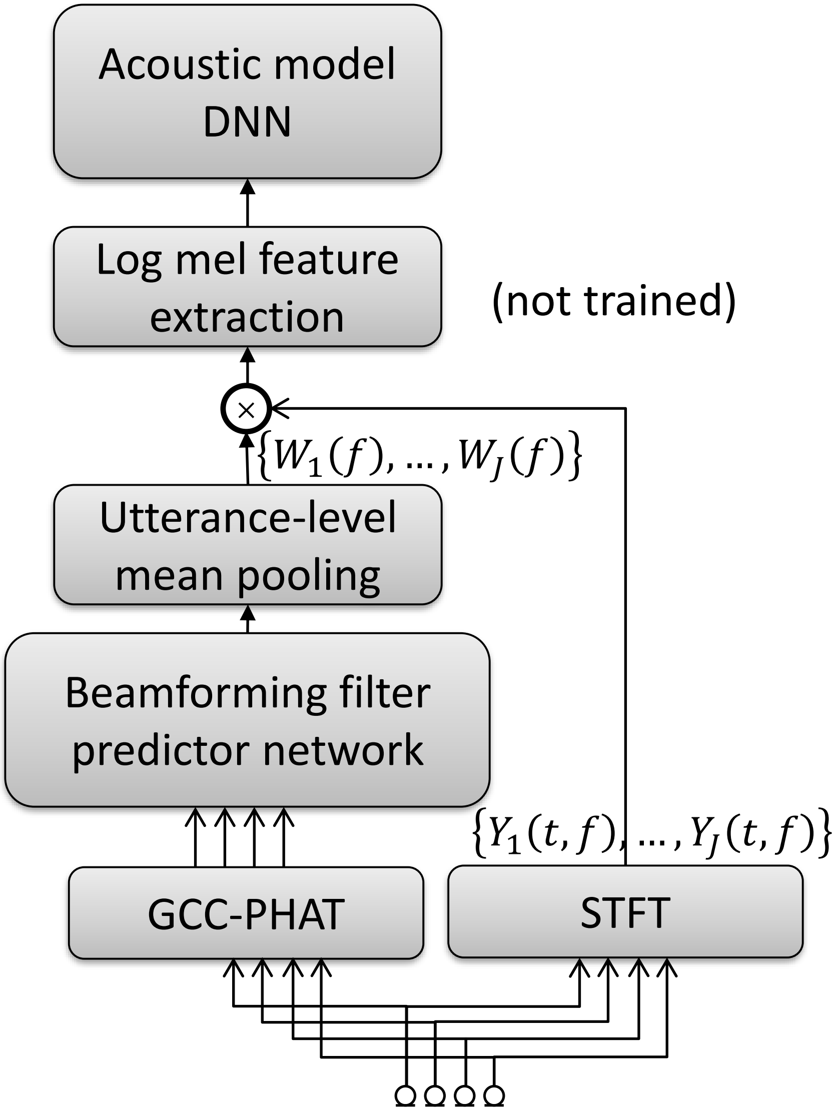
  - Time domain implementation (Li'16)
  - STFT domain implementation (Xiao'16)



$$\hat{X}(t,f) = \sum_j W_j(f) Y_j(t,f)$$

Complex domain neural network

Filter prediction-based beamforming network (Xiao'16)



- Beamforming and acoustic modeling can be expressed as a single neural network

→ Joint training becomes possible

- Mimic Log Mel Filterbank
- Utterance-level mean pooling
  - Time-independent linear filter $W_j(f)$

- Need careful training procedure
  - Train network, which predict Beamforming filter independently
    - Requires simulated data to have ground truth of the beamformer filter
  - Train acoustic model DNN independently on 1ch data
  - Refine with joint-optimization

Results on the AMI corpus

Results from (Xiao'16)

	WER
Single distant mic (1ch)	53.8 %
BeamformIt (8ch)	47.9 %
Beamforming filter predictor network (8ch)	47.2 %
+ Joint training (8ch)	**44.7 %**

Back-end configuration:

- Acoustic model (6 layer fully connected)
- Training criterion: Cross entropy

Remarks

- Integration of SE front-end and ASR back-end becomes possible when all components are using neural networks
- Joint optimization improves performance
- For multi-channel, including phase information using raw signals or STFT domain features appears more promising
  - There may be issues for unseen condition or unseen microphone configurations
  - Filter learning or filter prediction

References (Back-end 1/3)

(Chen'15)	Chen, Z., et al, "Integration of Speech Enhancement and Recognition using Long-Short Term Memory Recurrent Neural Network," Proc. Interspeech (2015).
(Chunyang'15)	Chunyang, W., et al. "Multi-basis adaptive neural network for rapid adaptation in speech recognition," Proc. ICASSP (2015).
(Delcroix'15a)	Delcroix, M., et al. "Strategies for distant speech recognition in reverberant environments," Proc. CSL (2015).
(Delcroix'15b)	Delcroix, M., et al. "Context adaptive deep neural networks for fast acoustic model adaptation," Proc. ICASSP (2015).
(Delcroix'16a)	Delcroix, M., et al. "Context adaptive deep neural networks for fast acoustic model adaptation in noise conditions," Proc. ICASSP (2016).
(Delcroix'16b)	Delcroix, M., et al. "Context adaptive neural network for rapid adaptation of deep CNN based acoustic models," Proc. Interspeech (2016).
(ETSI'07)	Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Adv. Front-end Feature Extraction Algorithm; Compression Algorithms, ETSI ES 202 050 Ver. 1.1.5 (2007).
(Gemmello'06)	Gemello, R., et al. "Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training," Proc. ICASSP (2006).
(Giri'15)	Giri, R., et al. "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," Proc. ICASSP (2015).
(Hori'15)	Hori, T., et al, "The MERL/SRI system for the 3rd CHiME challenge using beamforming, robust feature extraction, and advanced speech recognition," Proc. ASRU (2015).
(Hoshen'15)	Hoshen, Y., et al. "Speech Acoustic Modeling from Raw Multichannel Waveforms," Proc. ICASSP (2015).
(Kim'12)	Kim, C., et al. "Power-normalized cepstral coefficients (PNCC) for robust speech recognition." Proc. ICASSP (2012).
(Kundu'15)	Kundu, S., et al. "Joint acoustic factor learning for robust deep neural network based automatic speech recognition," Proc. ICASSP (2016).
(Li'16)	Li, B., et al. "Neural network adaptive beamforming for robust multichannel speech recognition," Proc. Interspeech (2016).

References (Back-end 2/3)

(Liao'13)	Liao, H. "Speaker adaptation of context dependent deep neural networks," Proc. ICASSP (2013).
(Liu'14)	Liu, Y., et al. "Using neural network front-ends on far field multiple microphones based speech recognition," Proc. ICASSP (2014).
(Mitra'14)	Mitra, V., et al. "Damped oscillator cepstral coefficients for robust speech recognition," Proc. Interspeech (2013).
(Marino'11)	Marino , D., et al. "An analysis of automatic speech recognition with multiple microphones.," Proc. Interspeech (2011).
(Neto'95)	Neto, J., et al. "Speaker adaptation for hybrid HMM-ANN continuous speech recognition system," Proc. Interspeech (1995).
(Ochiai'14)	Ochiai, T., et al. "Speaker adaptive training using deep neural networks," Proc. ICASSP (2014).
(Peddinti '15)	Peddinti, V., et al, "A time delay neural network architecture for efficient modeling of long temporal contexts." Proc. Interspeech (2015).
(Sainath'16)	Sainath, T. N., et al. "Factored spatial and spectral multichannel raw waveform CLDNNS," Proc. ICASSP (2016).
(Saon'13)	Saon, G., et al. "Speaker adaptation of neural network acoustic models using i-vectors," Proc. ASRU (2013).
(Shluter'07)	Schluter, R., et al. "Gammatone features and feature combination for large vocabulary speech recognition." Proc. ICASSP (2007).
(Seltzer'13)	Seltzer, M.L., et al. "An investigation of deep neural networks for noise robust speech recognition," Proc. ICASSP (2013).
(Swietojanski'13)	Swietojanski, P., et al. "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," Proc. ASRU (2013).
(Swietojanski'14a)	Swietojanski, P., et al. "Convolutional neural networks for distant speech recognition," IEEE Sig. Proc. Letters (2014).
(Swietojanski'14b)	Swietojanski, P., et al. "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models, " Proc. SLT (2014).

References (Back-end 3/3)

(Tachioka'13)	Tachioka, Y., et al. "Discriminative methods for noise robust speech recognition: A CHiME Challenge Benchmark," Proc. CHiME, (2013).
(Tachioka'14)	Tachioka, Y., et al. "Dual System Combination Approach for Various Reverberant Environments with Dereverberation Techniques," Proc. REVERB Workshop (2014).
(Tan'15)	Tan, T., et al. "Cluster adaptive training for deep neural network," Proc. ICASSP (2015).
(Waibel'89)	Waibel, A., et al. "Phoneme recognition using time-delay neural networks." IEEE transactions on acoustics, speech, and signal processing (1989).
(Weng'14)	Weng, C., et al. "Recurrent Deep Neural Networks for Robust Speech Recognition," Proc. ICASSP (2014).
(Weninger'14)	Weninger, F., et al. "The MERL/MELCO/TUM system for the REVERB Challenge using Deep Recurrent Neural Network Feature Enhancement," Proc. REVERB Workshop (2014).
(Weninger'15)	Weninger, F., et al, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in Proc. Latent Variable Analysis and Signal Separation (2015).
(Xiao'16)	Xiao, X., et al. "Deep beamforming networks for multi-channel speech recognition," Proc. of ICASSP (2016).
(Yoshioka'15a)	Yoshioka, T., et al. "Far-field speech recognition using CNN-DNN-HMM with convolution in time," Proc. ICASSP (2015).
(Yoshioka'15b)	Yoshioka, T., et al. "The NTT CHiME-3 system: advances in speech enhancement and recognition for mobile multi-microphone devices," Proc. ASRU (2015).
(Yu'13)	Yu, D., et al. "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," Proc. ICASSP (2013).

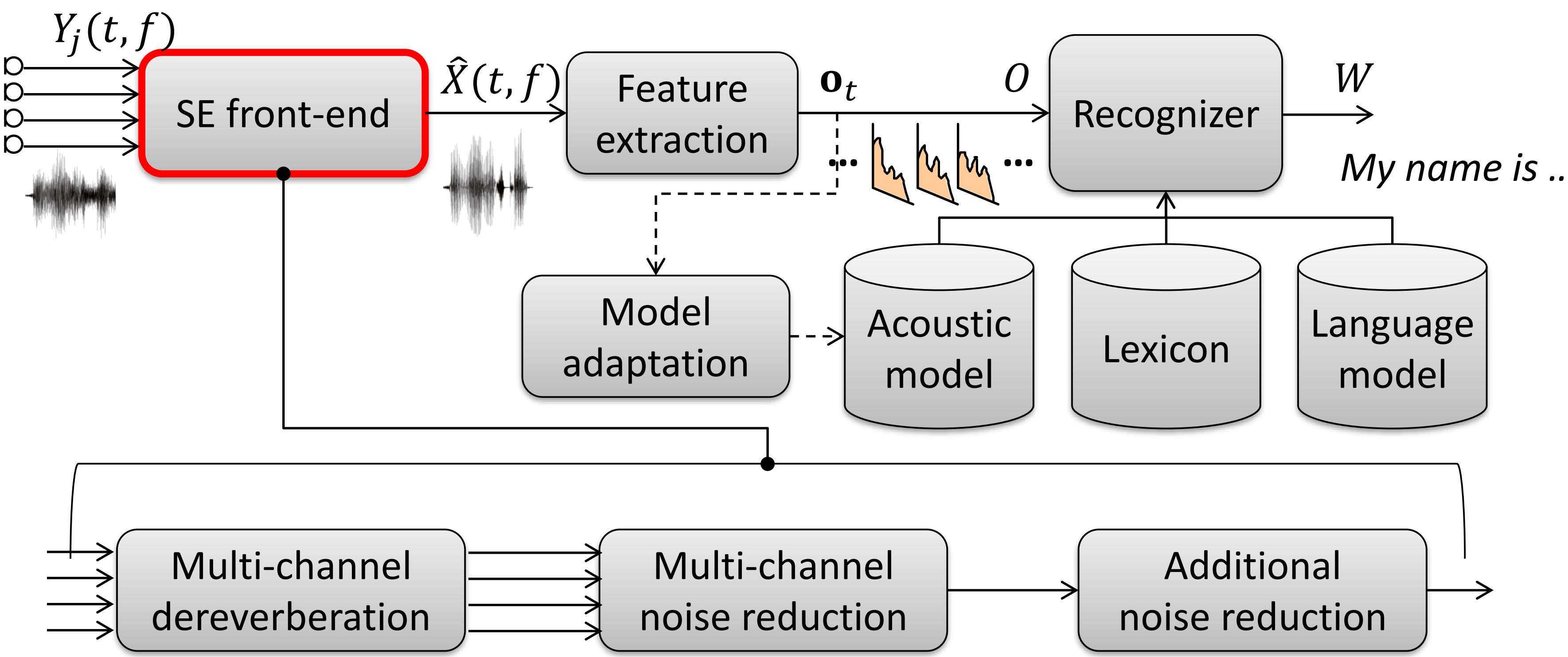
4. Building robust ASR systems

4.1 Overview of some successful systems at CHiME and REVERB

REVERB: NTT system

REVERB challenge system

(Delcroix'15)



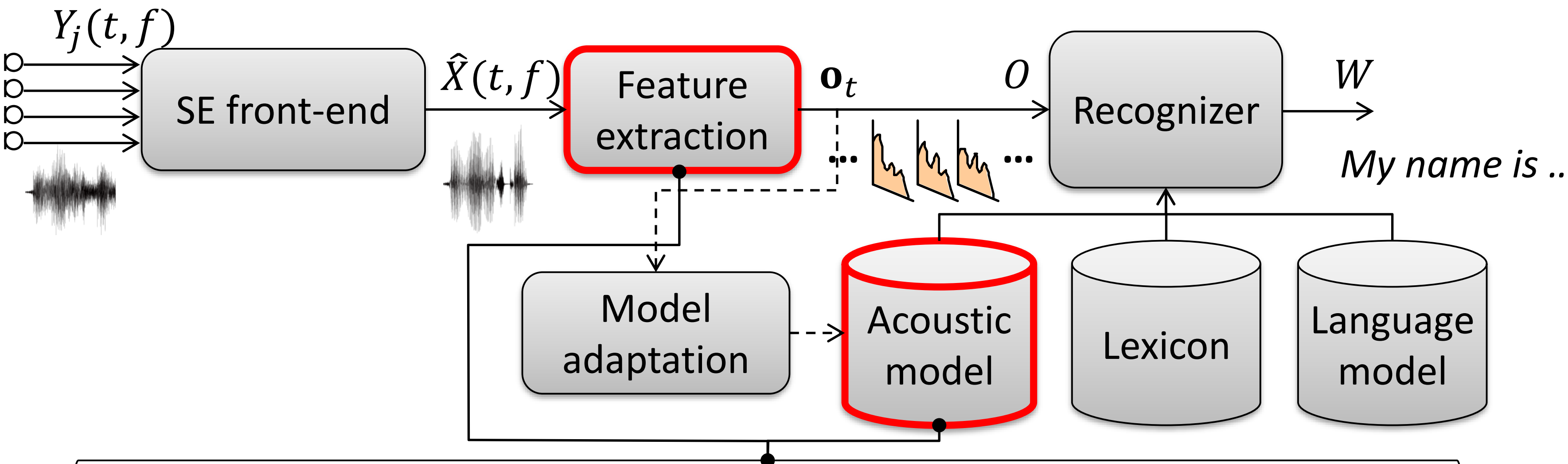
WPE

MVDR
Noise spatial correlation matrix computed from the first and last frames

DOLPHIN (Nakatani'13)
Spectral and spatial model combination based enhancement

REVERB challenge system

(Delcroix'15)



Features

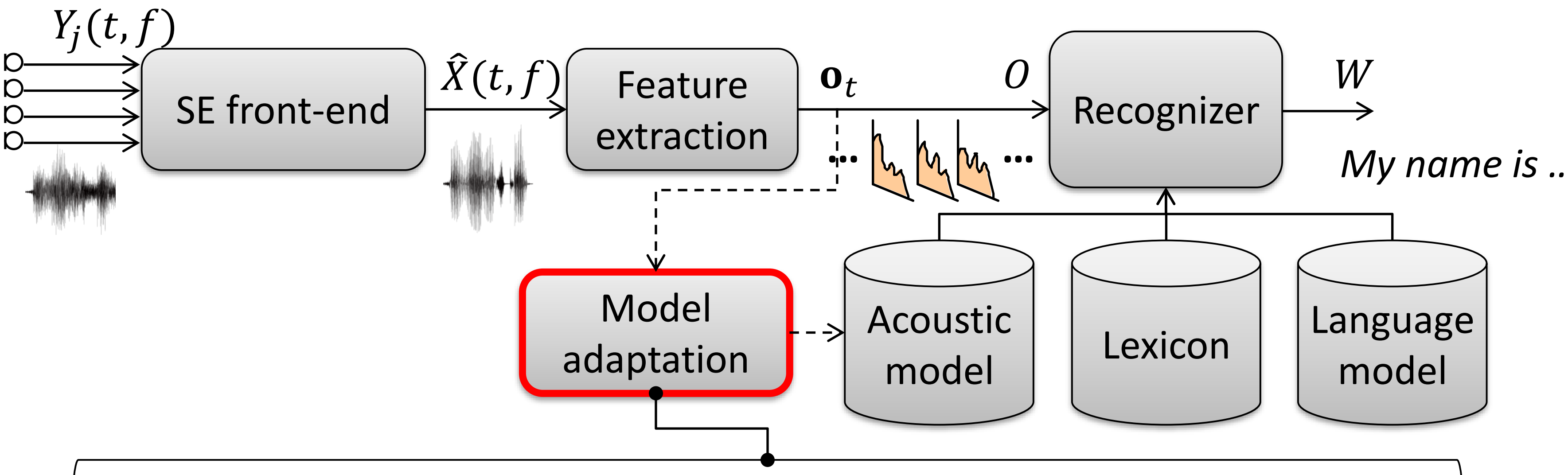
- 40 Log mel filter-bank coefficients + Δ + ΔΔ (120)
- 5 left+5 right context (11 frames)

Acoustic model

- DNN-HMM (7 hidden layers)
- RBM pre-training
- Training with data augmentation without SE front-end

REVERB challenge system

(Delcroix'15)

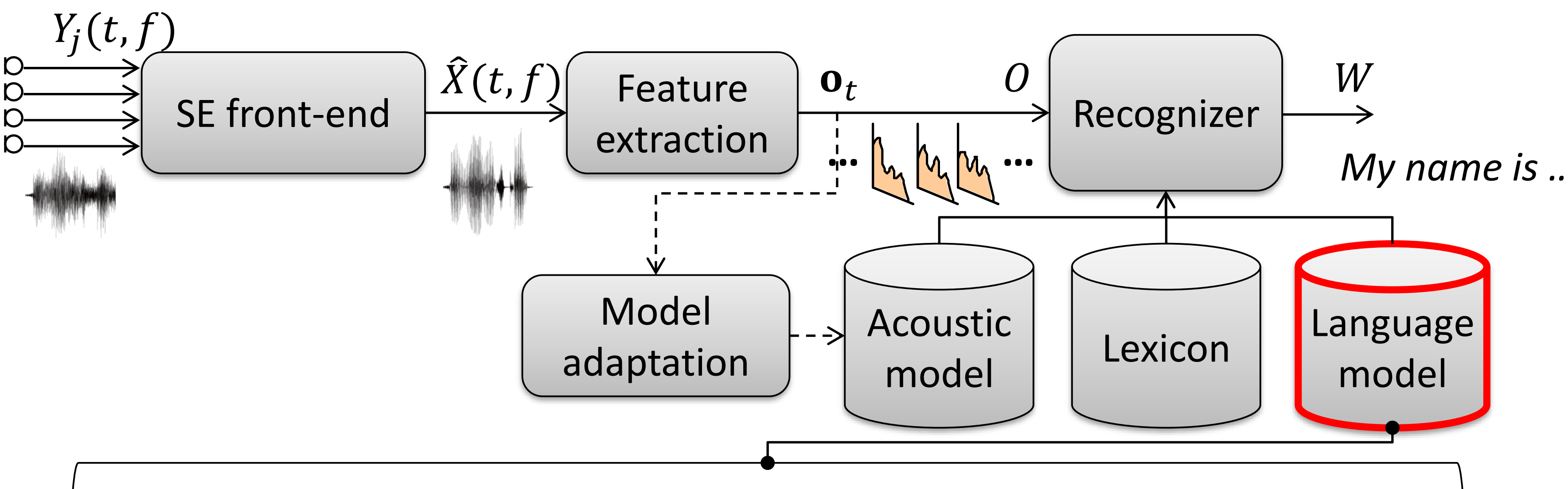


Unsupervised environmental adaptation

- Retrain 1st layer of DNN-HMM w/ small learning rate using
- Labels obtained from a 1st recognition pass

REVERB challenge system

(Delcroix'15)



Language model (LM)

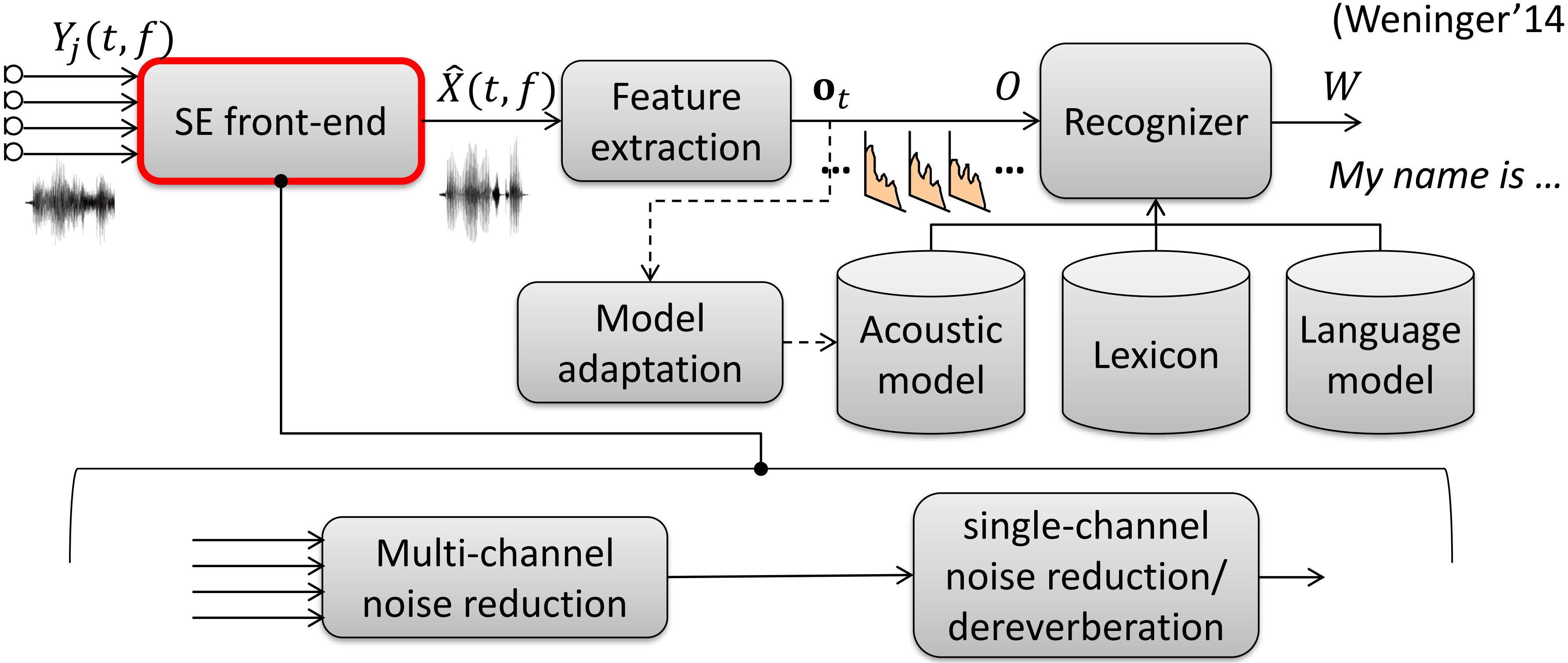
- Recurrent neural net (RNN) based LM w/ on-the-fly rescoring (Hori'14)

REVERB CHALLENGE

REVERB: MERL/MELCO/TUM system

REVERB challenge system

(Tachioka'14)
(Weninger'14)

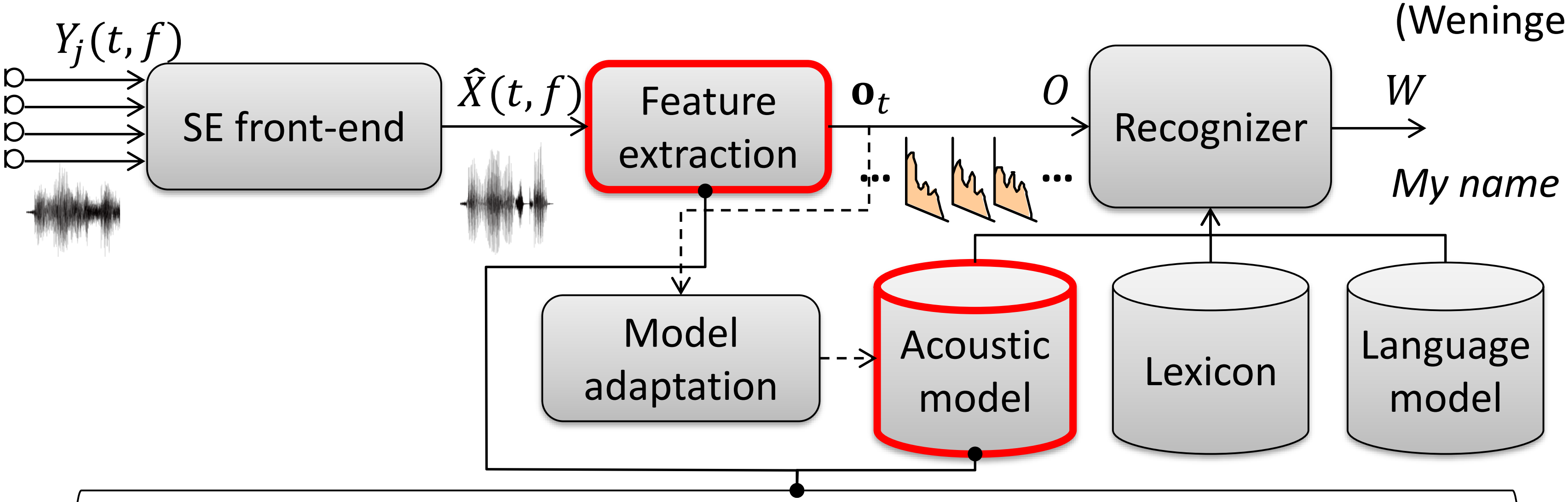


Delay and Sum Beamformer
GCC to compute TDOA

LSTM based dereverberation
2 or 3-layer LSTM (128 units)
Spectrum Subtraction based Dereverberation
Late reverberation estimation based on the reverberation time estimation

REVERB challenge system

(Tachioka'14)
(Weninger'14)



Acoustic model (GMM)

- 40 MFCC/PLP, LDA, MLLT, and fMLLR
- Feature-space MMI, boosted MMI

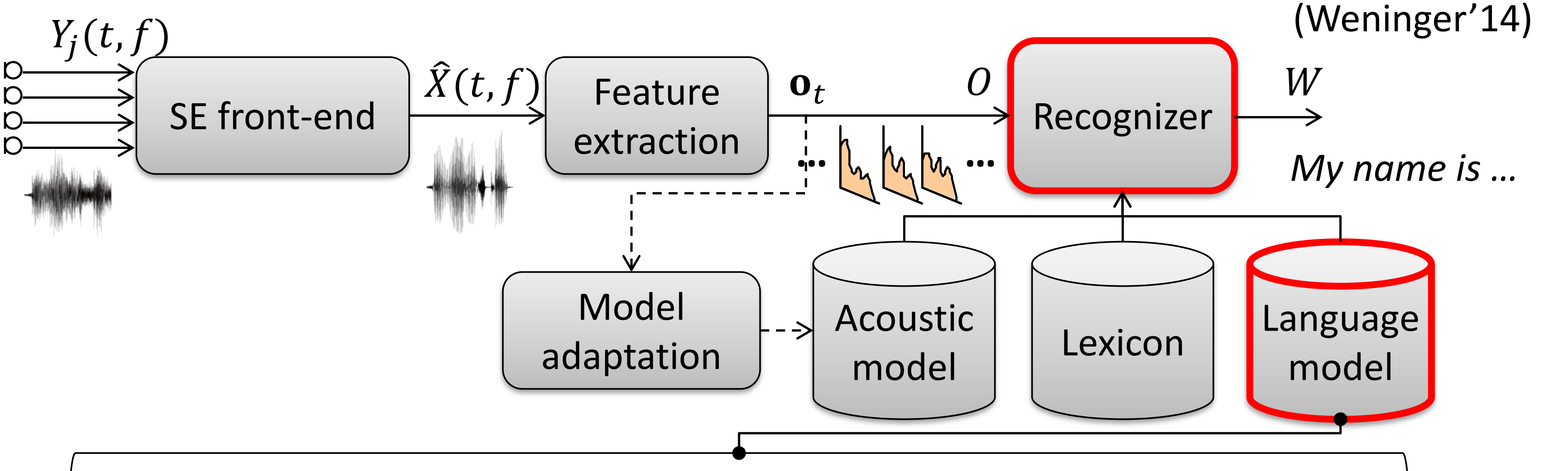
Acoustic model (LSTM)

- LSTM output corresponds to 23 Log mel filter-bank coefficients
- 3-layer LSTM (50 units)

Multi-Stream integration

REVERB challenge system

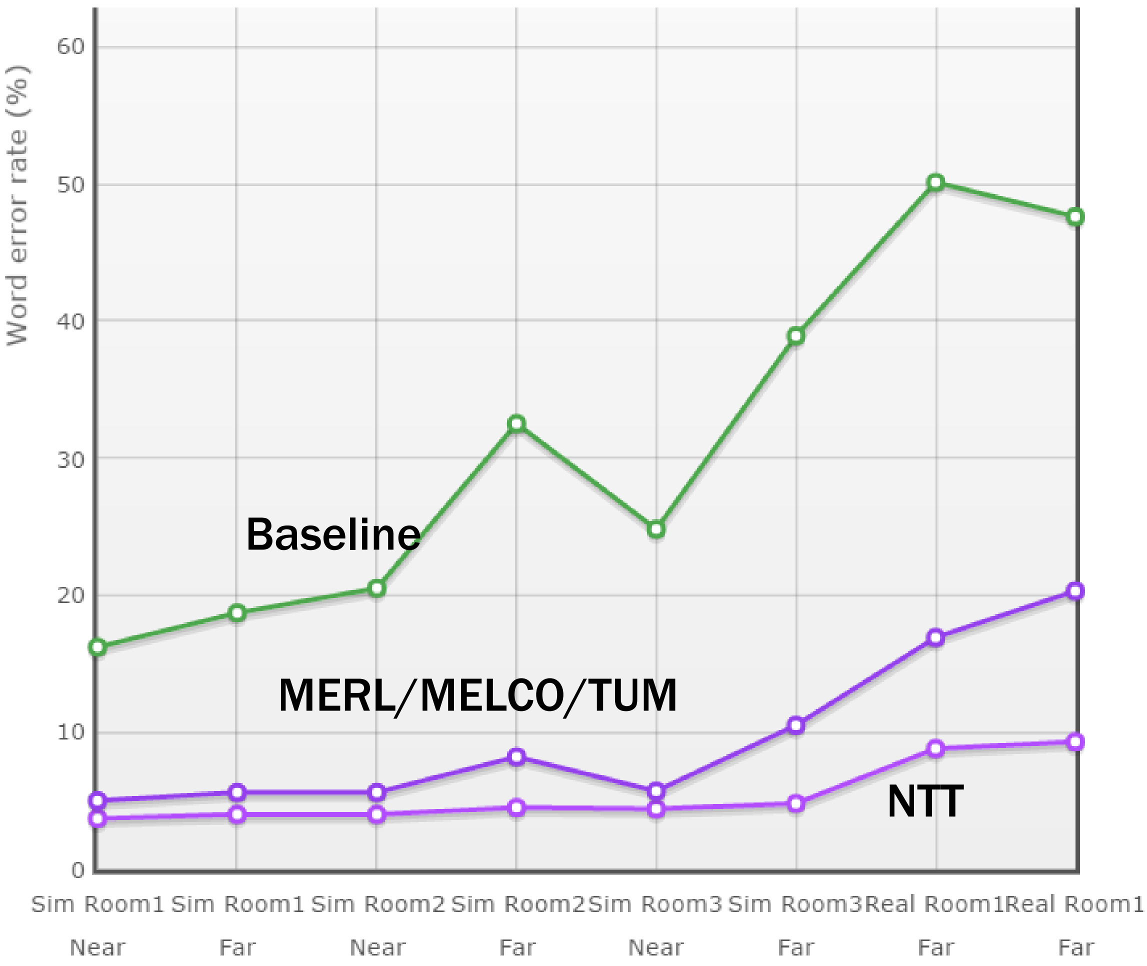
(Tachioka'14)
(Weninger'14)



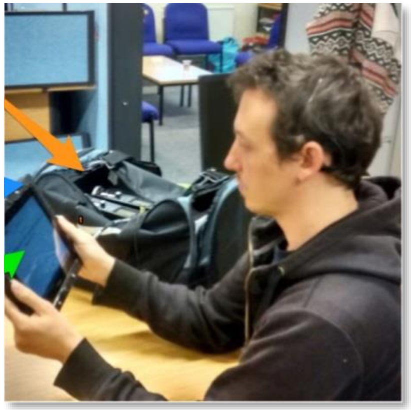
Language model (LM)
- 3-gram LM

Minimum Bayes Risk decoding
System combination

Results of top 2 systems



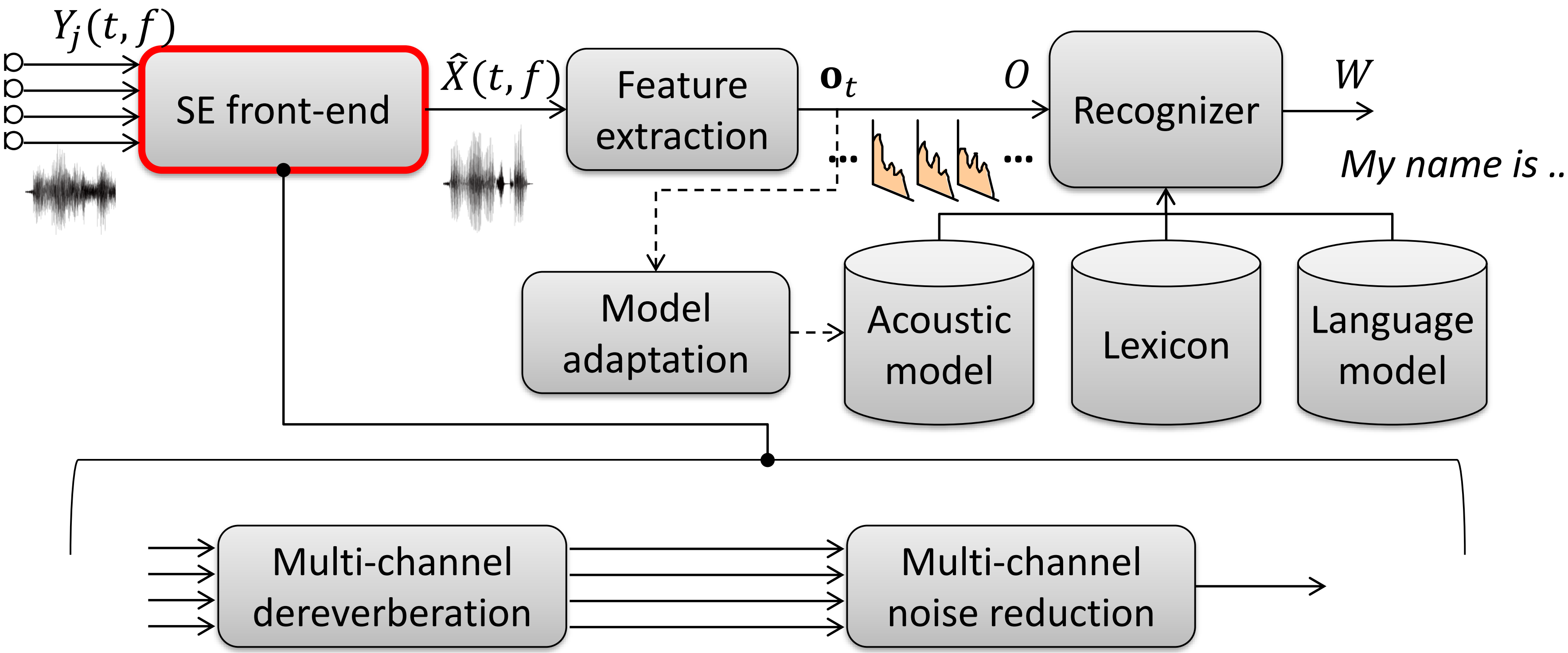
- Two systems significantly improve the performance from the baseline



CHiME 3: NTT system

CHiME3 challenge system

(Yoshioka'15)

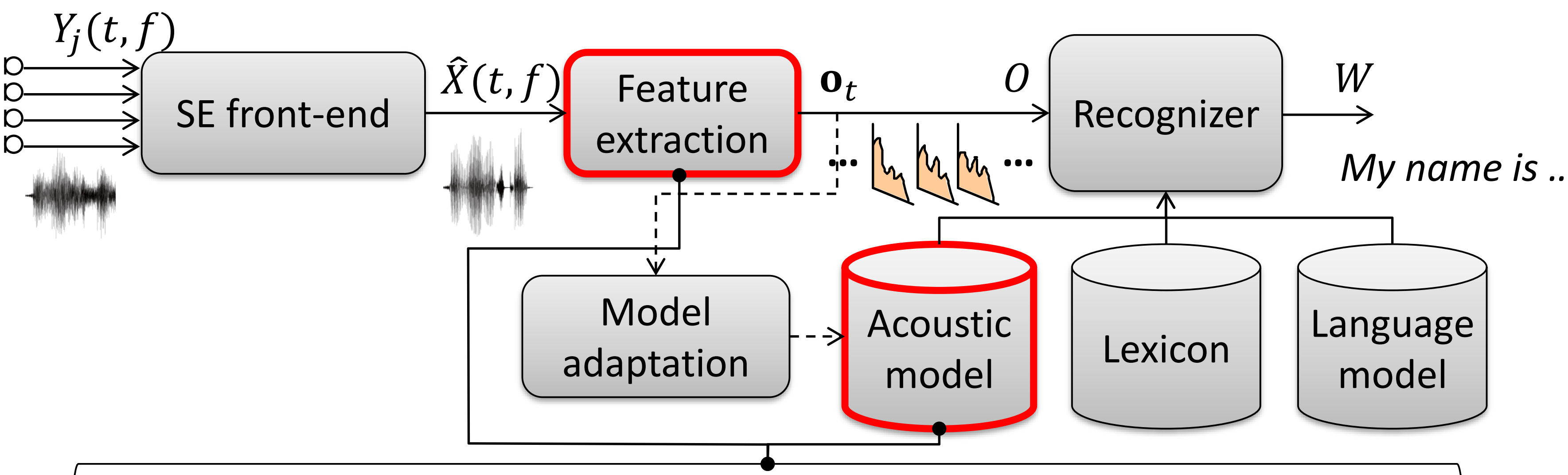


WPE

MVDR (Higuchi'16)
Spatial correlation matrix derived from time-frequency mask obtained by Clustering of spatial features

CHiME3 challenge system

(Yoshioka'15)



Features

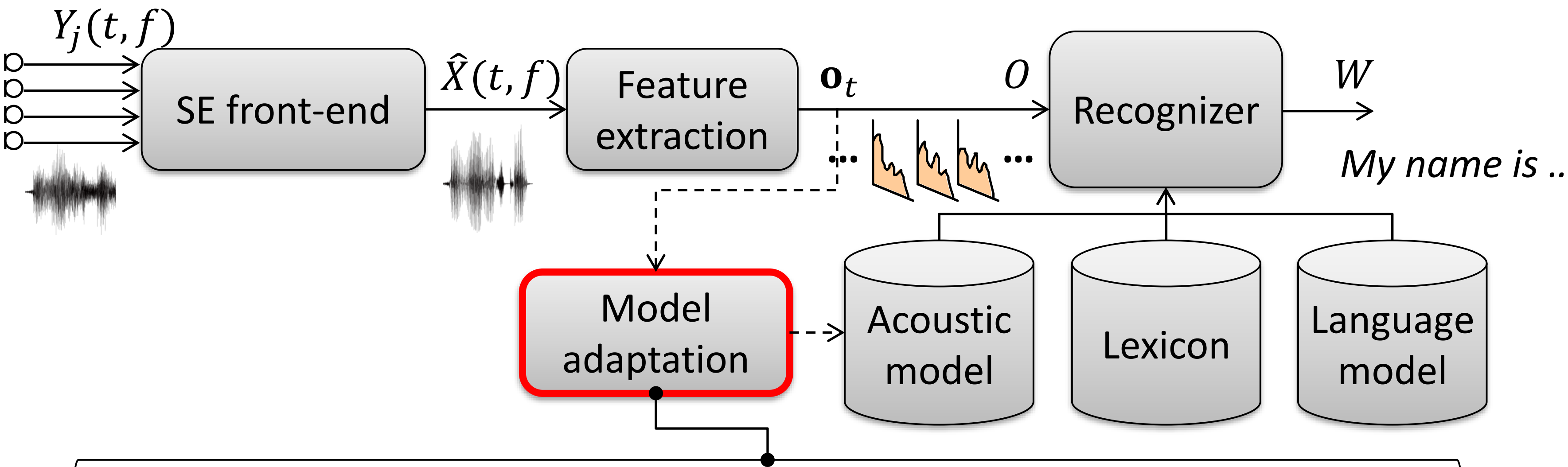
- 40 Log mel filter-bank coefficients + Δ + ΔΔ (120)
- 5 left+5 right context (11 frames)

Acoustic model

- Deep CNN using Network-in-Network
- Multi-channel training data (treat each channel training utterance as a separate training sample)
- Training without SE front-end

CHiME3 challenge system

(Yoshioka'15)

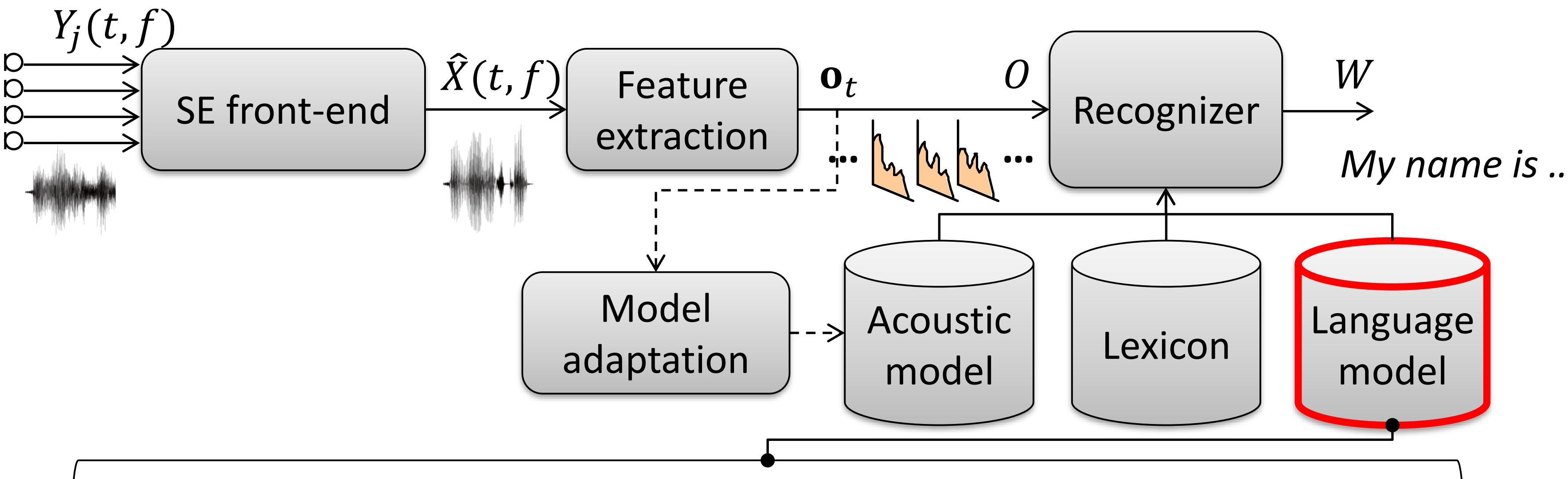


Unsupervised speaker adaptation

- Retrain all layers of CNN-HMM
- Labels obtained from a 1st recognition pass with DNN based system → cross adaptation (system combination)

CHiME3 challenge system

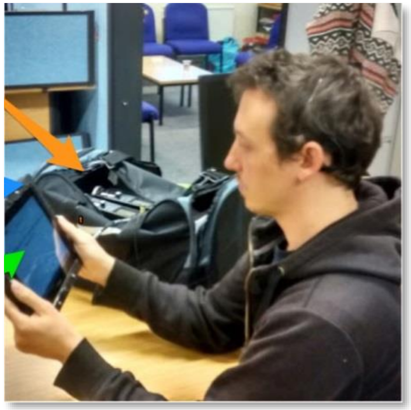
(Yoshioka'15)



Language model (LM)

- Recurrent neural net (RNN) based LM w/ on-the-fly rescoring (Hori'14)

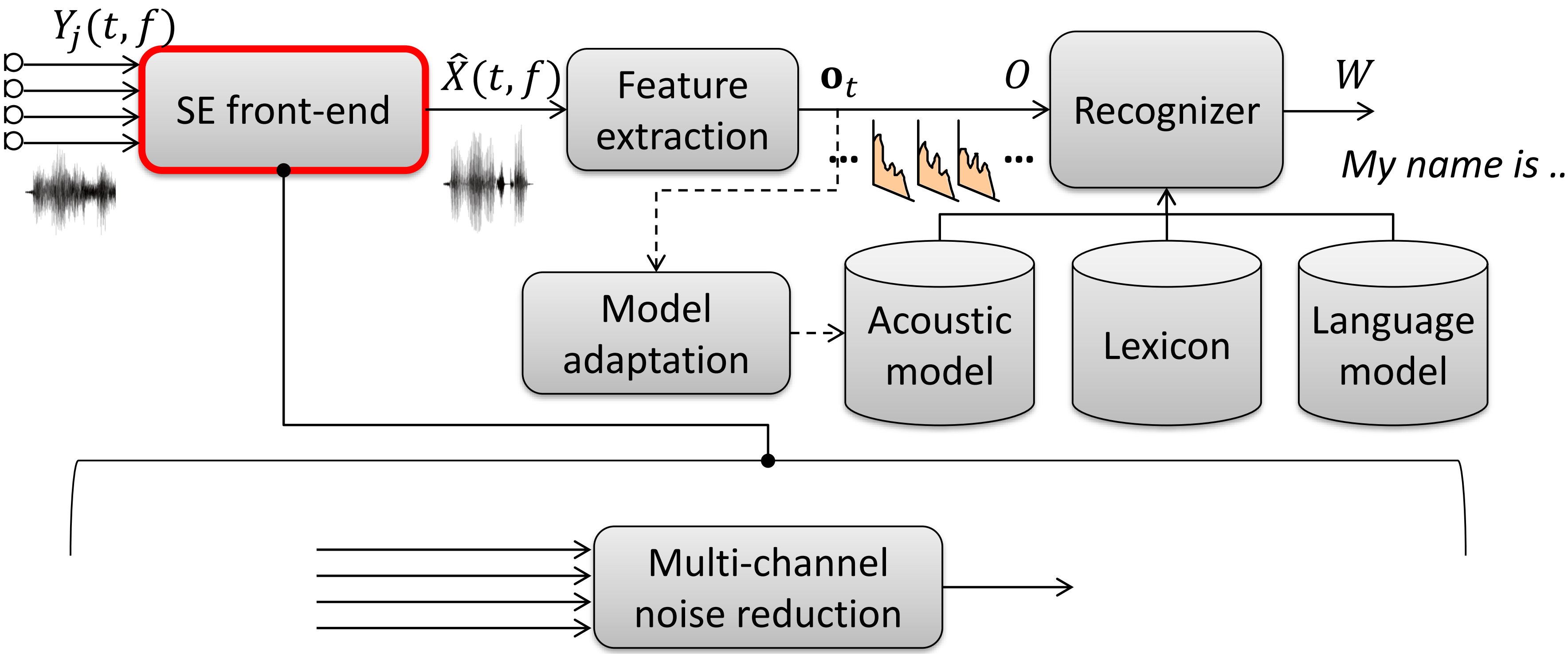
CHiME CHALLENGE



CHiME 3: MERL-SRI system

CHiME3 challenge system

(Hori'15)

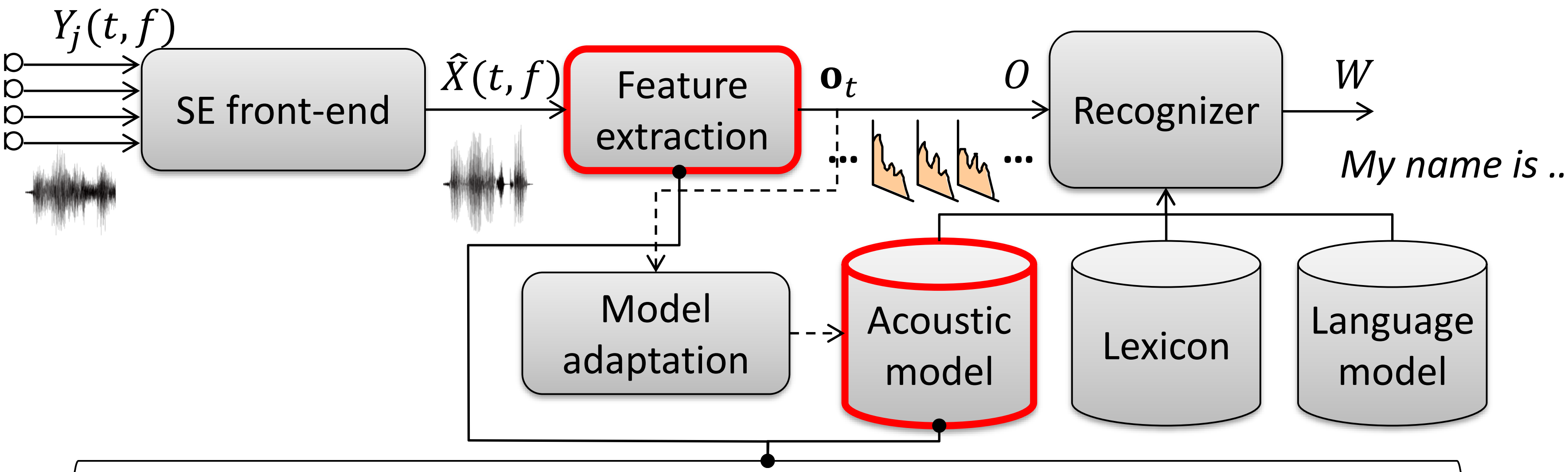


BeamformIt (Anguera'07)
LSTM Mask-based MVDR (Erdogan'16)

Both methods are integrated at system combination

CHiME3 challenge system

(Hori'15)

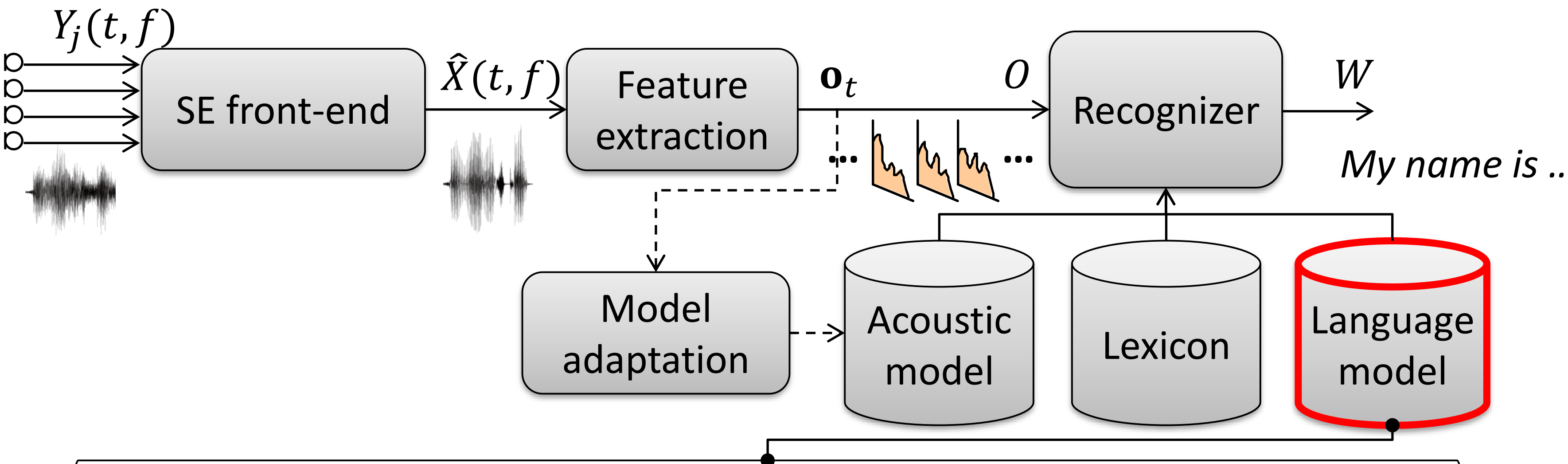


Features (3 type features. Integrated at system combination)
1) 40 Log mel filter-bank coefficients
2) Damped oscillator coefficients (DOC) (Mitra'14a)
3) Modulation of medium duration speech amplitudes (MMeDuSA) (Mitra'14b)
- 5 left+5 right context (11 frames)
- LDA, MLLT, fMLLR feature transformation

Acoustic model
- DNN with sMBR training
- Training with SE front-end

CHiME3 challenge system

(Hori'15)

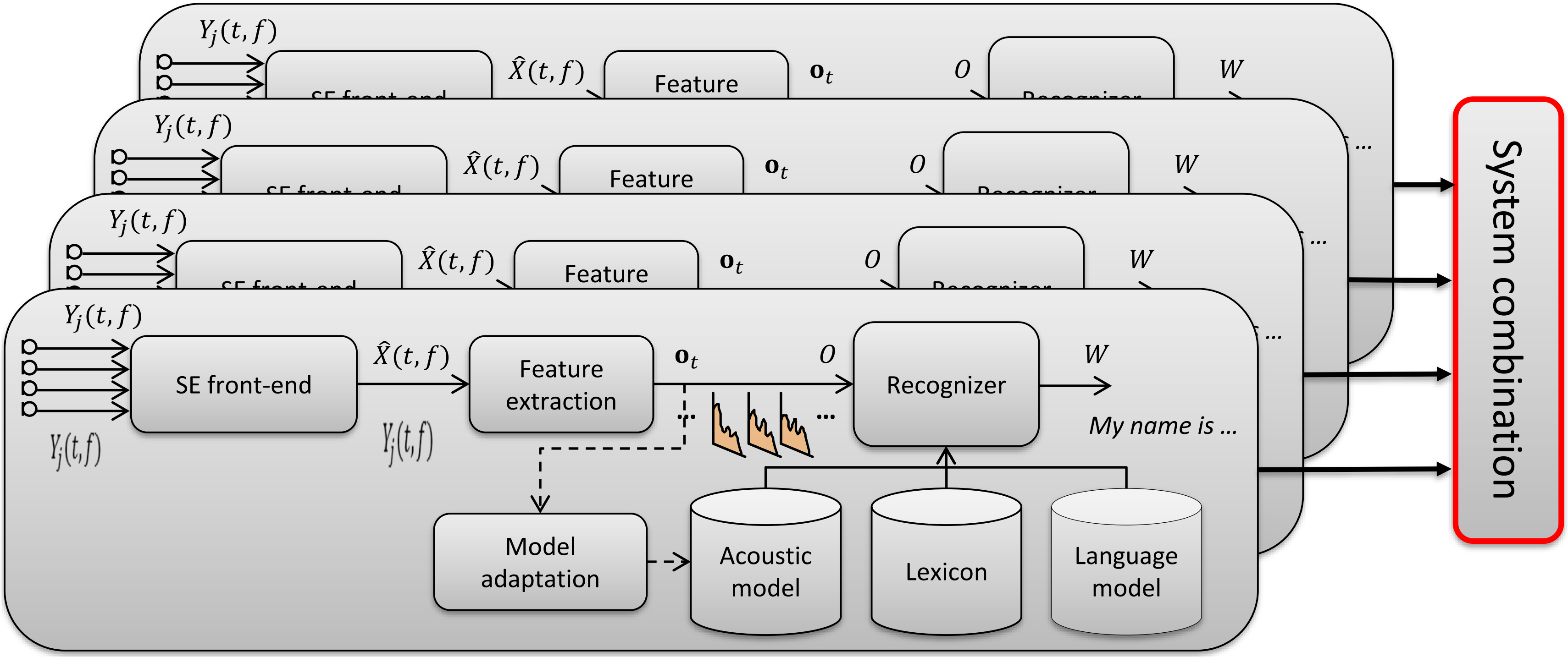


Language model (LM)

- Recurrent neural net (RNN) based LM

CHiME3 challenge system

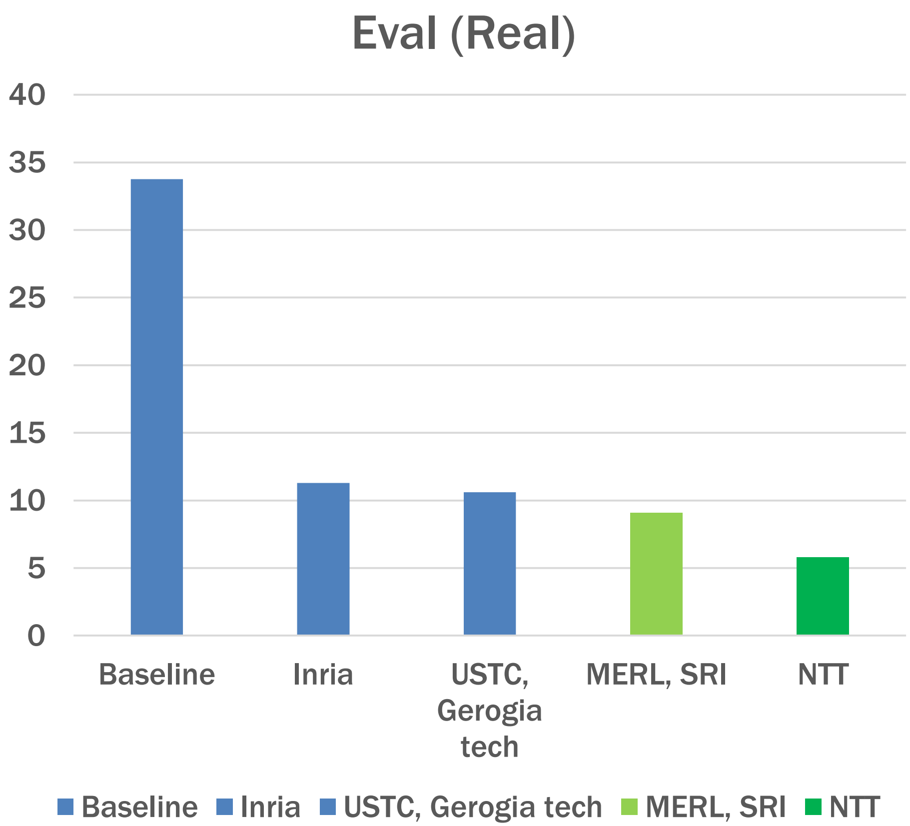
(Hori'15)



System combination

1) BeamformIt + Log mel filter-bank
2) BeamformIt + DOC
3) BeamformIt + MMeDuSA
4) Make-based MVDR + Log mel filter-bank

Results of top 4 systems



- Significant error reduction from the baseline (more than 60%)

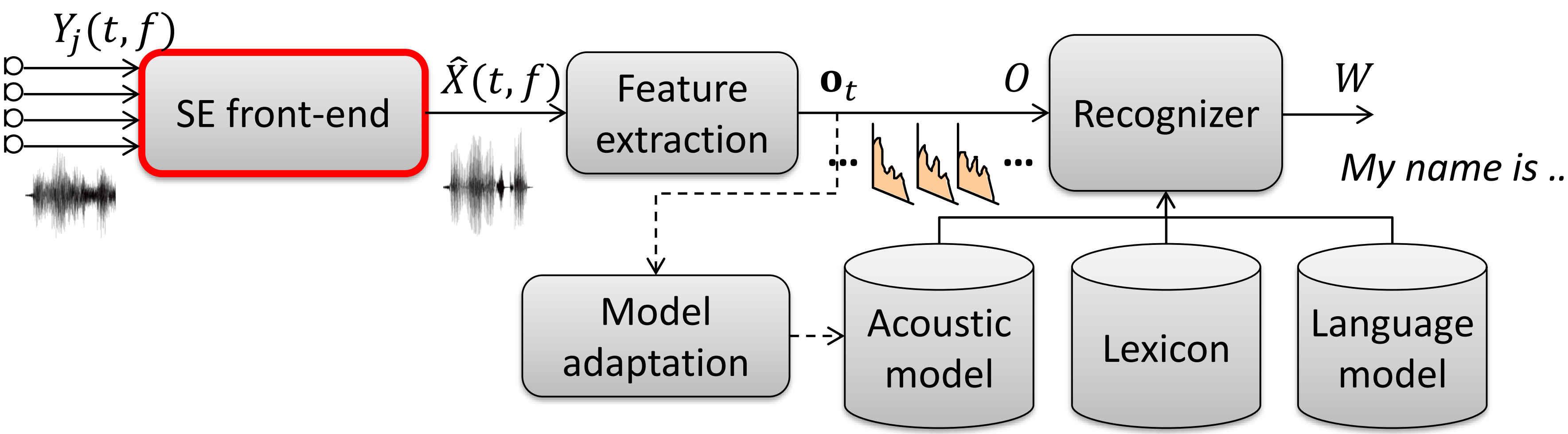
→ Top system reaches clean speech performance (~5%)

- All systems are very complex ☹ (reproducibility)

- We will discuss how to build such systems with existing tools

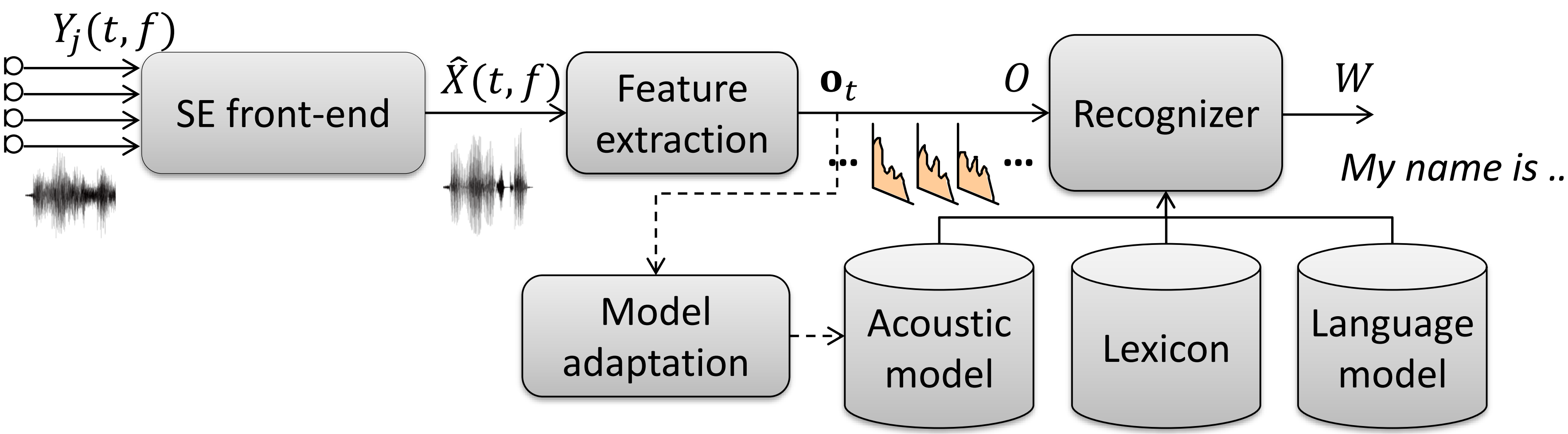
4.2 Overview of existing tools

SE front-end



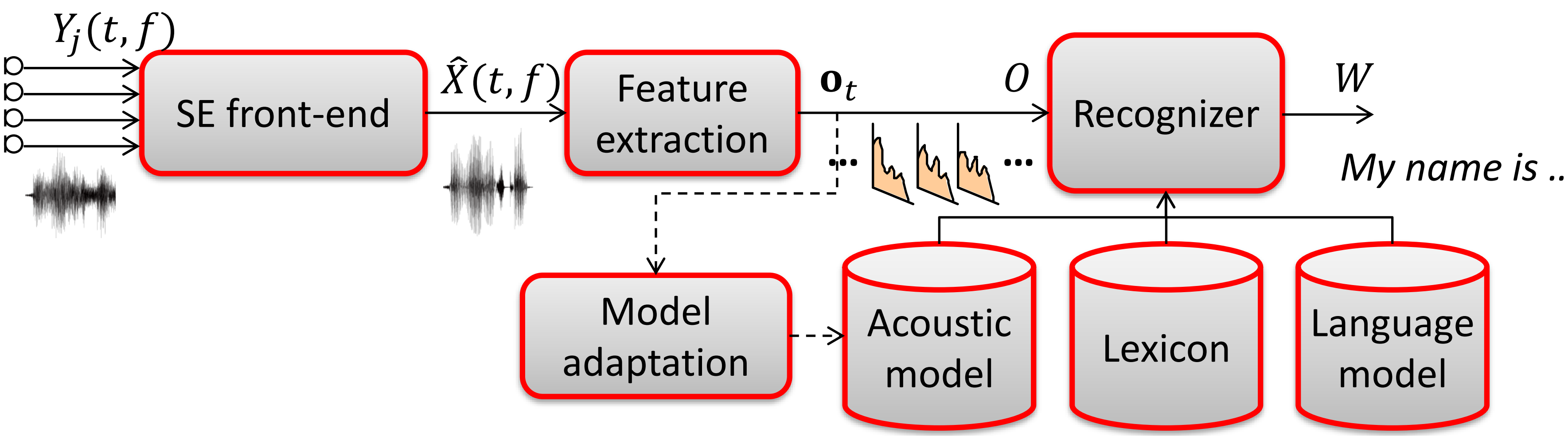
Tool	Institute	Function	Language	License
WPE	NTT	Dereverberation	Matlab	Proprietary
BeamformIt	ICSI/X. Anguera	Beamforming	C++	Apache 2.0
SRP-PHAT MVDR	Inria	Beamforming	Matlab	GPL
FASST	Inria	Multi-channel NMF	C++	GPL
NN-based GEV beamformer	U. Paderborn	Beamforming	Python	Non-commercial Educational

Whole system: Kaldi recipes



Recipe	Enhancement	Acoustic modeling	Language modeling	Main developers
REVERB	n/a	GMM	N-gram	F. Weninger, S. Watanabe
CHiME2	n/a	DNN, sMBR	N-gram	C. Weng, S. Watanabe
CHiME3	BeamformIt	DNN, sMBR	RNNLM	S. Watanabe
CHiME4	BeamformIt	DNN, sMBR	RNNLM	S. Watanabe
AMI	BeamformIt	DNN, sMBR, LSTM, TDNN	N-gram	P. Swietojanski, V. Peddinti
ASpIRE	n/a	DNN, sMBR, LSTM, TDNN	N-gram	V. Peddinti

Whole system: Kaldi recipes



Recipe	Enhancement	Acoustic modeling	Language modeling	Main developers
REVERB	n/a	GMM	N-gram	F. Weninger, S. Watanabe
CHiME2	n/a	DNN, sMBR	N-gram	C. Weng, S. Watanabe
CHiME3	BeamformIt	DNN, sMBR	RNNLM	S. Watanabe
CHiME4	BeamformIt	DNN, sMBR	RNNLM	S. Watanabe
AMI	BeamformIt	DNN, sMBR, LSTM, TDNN	N-gram	P. Swietojanski, V. Peddinti
ASpIRE	n/a	DNN, sMBR, LSTM, TDNN	N-gram	V. Peddinti

CHiME4 Kaldi recipe based on free software

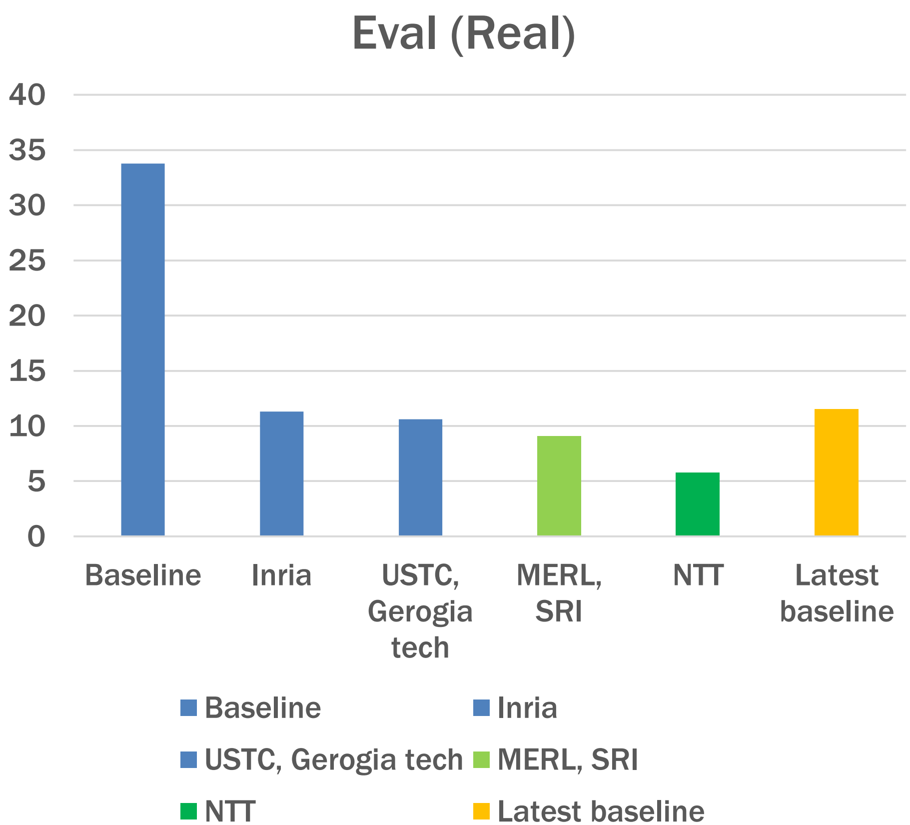
1. Get CHiME4 data
   http://spandh.dcs.shef.ac.uk/chime_challenge/software.html
   – Registration → LDC license confirmation step → credentials
2. Get Kaldi
   https://github.com/kaldi-asr/kaldi
3. Install Kaldi tools
   – In addition to default Kaldi tools, you have to install BeamformIt, IRSTLM, SRILM, and Milonov's RNNLM (all are prepared in kaldi/tools/extras
   – For SRILM, you need to get source (srilm.tgz) at http://www.speech.sri.com/projects/srilm/download.html
4. Install Kaldi
5. Specify CHiME4 data root paths in kaldi/egs/s5_6ch/run.sh
6. Execute ./run.sh

kaldi/egs/s5_6ch/run.sh

```
#!/bin/bash

chime4_data=/db/laputa1/data/processed/public/CHiME4
local/run_init.sh $chime4_data

enhancement_method=beamformit_5mics
enhancement_data=`pwd`/enhan/$enhancement_method
local/run_beamform_6ch_track.sh --cmd "$train_cmd" --nj 20 \
  $chime4_data/data/audio/16kHz/isolated_6ch_track $enhancement_data

local/run_gmm.sh $enhancement_method $enhancement_data $chime4_data

local/run_dnn.sh $enhancement_method

local/run_lmrescore.sh $chime4_data $enhancement_method
```

- **run_init.sh**: creates 3-gram LM, FSTs, and basic task files
- **run_beamform_6ch_track.sh**: beamforming with 5 channel signals
- **run_gmm.sh**: LDA, MLLT, fMLLR based GMM
- **run_dnn.sh**: DNN + sMBR
- **run_lmrescore.sh**: 5-gram and RNNLM rescoring

Result and remarks



- Already obtain top level performance (11.5%)
- Everyone can **reproduce** the same results!
  - Concentrate on developing a new technology
- Still have a gap
- **Contribute** to DSR recipes to improve/standardize DSR pipeline for the community, e.g.
  - Advanced beamforming
  - Advanced acoustic modeling
  - Data simulation
  - DNN enhancement

References (Building systems)

(Anguera'07)	Anguera, X., et al. "Acoustic beamforming for speaker diarization of meetings," IEEE Trans. ASLP (2007).
(Barker'15)	Barker, J., et al, "The third `CHiME' Speech Separation and Recognition Challenge: Dataset, task and baselines," Proc. ASRU (2015).
(Delcroix'15)	Delcroix, M., et al. "Strategies for distant speech recognition in reverberant environments," CSL (2015).
(Erdogan'16)	Erdogan, H., et al. Improved MVDR beamforming using single-channel mask prediction networks," Proc. Interspeech (2016).
(Hori'14)	Hori, T., et al. "Real-time one-pass decoding with recurrent neural network language model for speech recognition," Proc. ICASSP (2014).
(Hori'15)	Hori, T., et al. "The MERL/SRI system for the 3rd CHiME challenge using beamforming, robust feature extraction, and advanced speech recognition," Proc. ASRU (2015).
(Mitra'14a)	Mitra, V., et al. "Damped oscillator cepstral coefficients for robust speech recognition," Proc. Interspeech (2013).
(Mitra'14b)	Mitra, V., et al. "Medium duration modulation cepstral feature for robust speech recognition," Proc. ICASSP (2014).
(Nakatani'13)	Nakatani, T. et al. "Dominance based integration of spatial and spectral features for speech enhancement," IEEE Trans. ASLP (2013).
(Tachioka'14)	Tachioka, Y., et al. "Dual System Combination Approach for Various Reverberant Environments with Dereverberation Techniques," Proc. REVERB Workshop (2014).
(Wang'16)	Wang, Z.-Q. et al. "A Joint Training Framework for Robust automatic speech recognition," IEEE/ACM Trans. ASLP (2016).
(Weninger'14)	Weninger, F., et al. "The MERL/MELCO/TUM system for the REVERB Challenge using Deep Recurrent Neural Network Feature Enhancement," Proc. REVERB Workshop (2014).
(Yoshioka'15)	Yoshioka, T., et al. "The NTT CHiME-3 system: advances in speech enhancement and recognition for mobile multi-microphone devices," Proc. ASRU (2015).

6. Conclusion and future research directions

Conclusion

- Combining SE and ASR techniques greatly improves performance in severe conditions
  - SE front-end technologies
    - Microphone array,
    - Neural network-based speech enhancement, …
  - ASR back-end technologies
    - Feature extraction/transformation
    - RNN/LSTM/TDNN/CNN based acoustic modeling
    - Model adaptation, …

- Introduction of deep learning had a great impact on DSR
  - Large performance improvement
  - Reshuffling the importance of technologies

- There remains many challenges and opportunities for further improvement

Toward joint optimization?

Separate optimization

SE front-end → ASR back-end

- Both components are designed with different objective functions

☺ Potentially SE front-end can be made more robust to unseen acoustic conditions (noise types, different mic configurations)

☹ Not optimal for ASR

Joint optimization

SE front-end ASR back-end

- Both components are optimized with the same objective functions

☹ Potentially more sensitive to mismatch between training and testing acoustic conditions

☺ Optimal for ASR

- Joint training is a recent active research topic
  - Currently integrate front-end and acoustic model
  - Combined with *end-to-end* approaches it could introduce higher level cues to the SE front-end (linguistic info…)

Dealing with uncertainties

- Advanced GMM-based systems exploited the uncertainty of the SE front-end during decoding (Uncertainty decoding)
  - Provided a way to interconnect speech enhancement front-end and ASR back-end optimized with different criteria

- Exploiting uncertainty within DNN-based ASR systems has not been sufficiently explored yet
  - Joint training is one option
  - Are there other?

More severe constraints

- Limited number of microphones
  - Best performances are obtained when exploiting multi-microphones

1ch	2ch	8ch	Lapel	Headset
17.4 %	12.7 %	9.0 %	8.3 %	5.9 %

REVERB challenge

  - Remains a great gap between performance with a single-microphone

→ Developing more powerful single-channel approaches remains an important research topic

- Many systems assume batch processing or utterance batch processing

  → Need further research for online & real-time processing

More diverse acoustic conditions

- More challenging situations are waiting to be tackled
  - Dynamic conditions
    - Multiple speakers
    - Moving speakers, …
  - Various conditions
    - Variety of microphone types/numbers/configurations
    - Variety of acoustic conditions, rooms, noise types, SNRs, …
  - More realistic conditions
    - Spontaneous speech
    - Unsegmented data
    - Microphone failures, …
  - New directions
    - Distributed mic arrays, …

→ New technologies may be needed to tackle these issues
→ New corpora are needed to evaluate these technologies

Larger DSR corpora

- Some industrial players have access to large amount of field data…
  … most publicly available DSR corpora are relatively small scale
- It has some advantages,
  ☺ Lower barrier of entry to the field
  ☺ Faster experimental turnaround
  ☺ New applications start with limited amount of available data

But…

Are the developed technologies still relevant when training data cover a large variety of conditions?

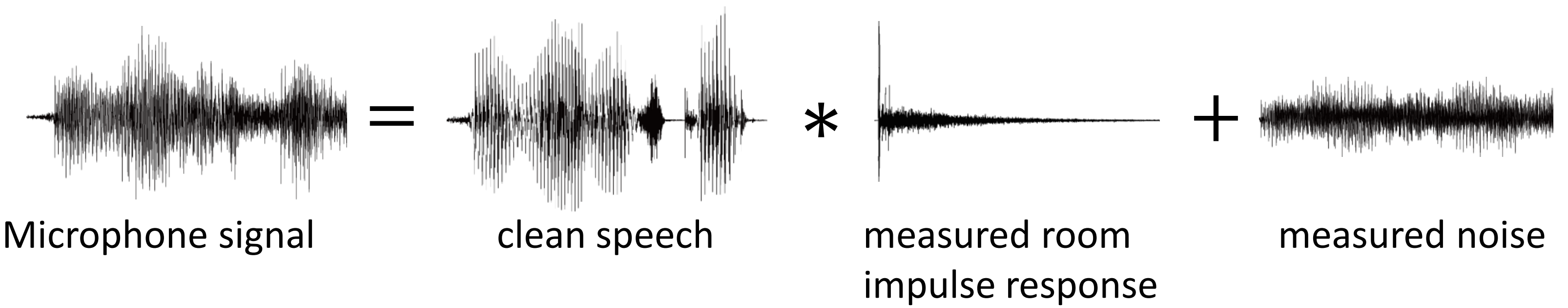
Could the absence of large corpora hinder the development of data demanding new technologies?

→ There is a need to create larger publicly available DSR corpus

DSR data simulation

- Low cost way to obtain large amount of data covering many conditions
- Only solution to obtain noisy/clean parallel corpora

- Distant microphone signals can be simulated as



  – Good simulation requires measuring the room impulse responses and the noise signals in the same rooms with the same microphone array
  – Still …
    - Some aspect are not modeled e.g. head movements
    - It is difficult to measure room impulse response in public spaces,…

DSR data simulation

- Recent challenges results showed that
  - Simulated data help for acoustic model training
    - No need for precise simulation
  - Results on simulated data do not match results on real data when using an SE front-end
    - SE models match better to simulated data → Causes overfitting

→ Need to develop better simulation techniques

Toolkits

- ASR research has long history of community developed toolkits and recipes
- Toolkits and recipes are important to
  - Lower barrier of entrance
  - Reproducibility of results
  - Speedup progress in the field
- Recent DSR recipes for REVERB and CHiME challenges include state-of-the-art back-end technologies
- Much less toolkits and recipes available for SE technologies

→Community based development of SE toolkits could contribute to faster innovation for DSR

Cross community

- DSR research requires combination of
  - SE front-end technologies
  - ASR back-end technologies

→ Cross disciplinary area of research from speech enhancement, microphone array, ASR…

→ Recent challenges (CHiME, REVERB) have contributed to increase synergy between the communities by sharing
  - Common tasks
  - Baseline systems
  - Share knowledge
    - Edit book to appear “New Era for Robust Speech Recognition: Exploiting Deep Learning,” Springer (2017)

Thank you!

Acknowledgments

Acknowledgments

- We would like to thank our colleagues at *NTT, MERL, MELCO, TUM, SRI* and at *the 2015 Jelinek summer workshop on Speech and Language technology (JSALT)* for their direct or indirect contributions to the content of this tutorial.
- In particular we are deeply grateful to *Xiong Xiao* (NTU), *Vikramjit Mitra* (SRI), *Horacio E. Franco* (SRI), *Victor Abrash* (SRI), *Jonathan Le Roux* (MERL), *Takaaki Hori* (MERL), *Christian Huemmer* (Erlangen Univ.), *Tsubasa Ochiai* (Doshisha Univ.), *Shigeki Karita* (NTT), *Keisuke Kinoshita* (NTT) and *Tomohiro Nakatani* (NTT) for their comments and suggestions on the content of the tutorial.