The MERL/MELCO/TUM system for the REVERB Challenge using Deep Recurrent Neural Network Feature Enhancement

Felix Weninger[1,2], *Shinji Watanabe*[1], *Jonathan Le Roux*[1], *John R. Hershey*[1], *Yuuki Tachioka*[3], *Jürgen Geiger*[2], *Björn Schuller*[2], *Gerhard Rigoll*[2]

[1] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
[2] MMK, Technische Universität München, Munich, Germany
[3] IT R&D Center, Mitsubishi Electric Corporation, Kamakura, Japan

REVERB CHALLENGE Workshop

Florence, Italy
May 10, 2014

Mitsubishi Electric Changes for the Better

Motivation

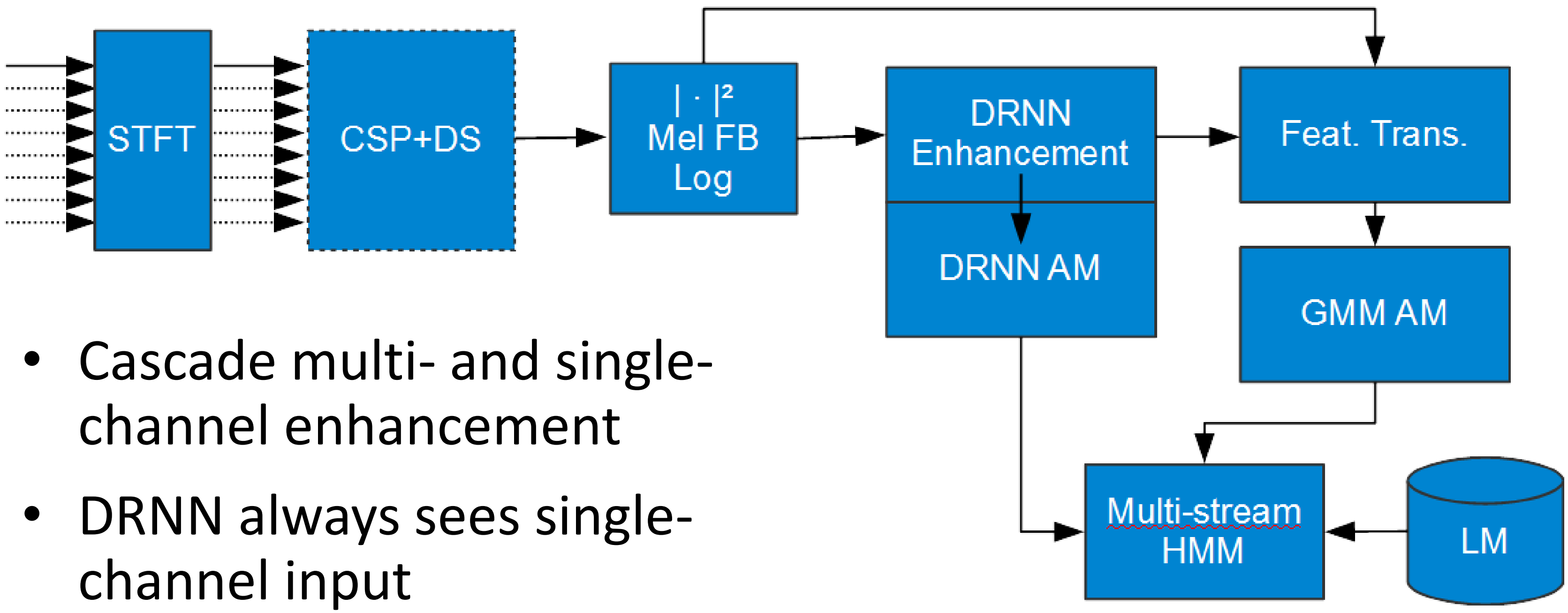
- Deep recurrent neural network (DRNN) feature enhancement: promising for reverberated ASR
- Potential performance improvement by additional:
    - Discriminative GMM training
    - DRNN acoustic modeling
    - Integration of multi- and single-channel enhancement

F. Weninger et al., Deep Recurrent De-Noising Auto-Encoder and Blind De-Reverberation for Reverberated Speech Recognition, ICASSP 2014

Y. Tachioka et al., Effectiveness of discriminative training for recognition of reverberated and noisy speech, ICASSP 2013

J. Geiger et al., Memory-Enhanced Recurrent Neural Networks and NMF for Robust ASR, T-ASLP 2014

System Overview



- Cascade multi- and single-channel enhancement
- DRNN always sees single-channel input
- Multi-stream HMM decoding
  - Cf. CHiME Challenge (Geiger et al., T-ASLP, 2014)

Multi-Channel Processing

- Cross-spectrum phase (CSP) + delay-and-sum (DS) beam-forming in the spectral domain

$$\tau_{1,m} = \arg\max \mathcal{S}^{-1}\left[\frac{\mathbf{z}_t(1) \odot \mathbf{z}_t(m)^*}{|\mathbf{z}_t(1)||\mathbf{z}_t(m)|}\right]$$

$$\hat{\mathbf{z}}_t = \sum_m \mathbf{z}_t(m) \odot \exp(-\jmath\boldsymbol{\omega}\tau_{1,m})$$

- Peak-hold process
- Noise component suppression

Single-channel DRNN-DAE enhancement

- Enhancement by de-noising auto-encoder (DAE)
  - Supervised training of mapping from reverberated and noisy to clean speech features (Log Mel)
  - Trained on simulated parallel data – does it generalize?
- Implement DAE as deep recurrent neural network (RNN) with Long Short-Term Memory (LSTM) architecture
- Successful in ASR feature enhancement task
  - Outperforms DNN on CHiME
- LSTM-RNN:
  - Adaptive context size
  - Models output dynamics

(Weninger et al., CSL, 2014)

LSTM de-reverberation

Noisy + reverberated features

Matrices obtained from supervised training

Compute input / forget gate activation based on feed-forward and recurrent part

Update cell state

Output cell state to hidden activation

Estimated clean speech features

$$
\begin{aligned}
\mathbf{h}_t^{(0)} &:= \tilde{\mathbf{x}}_t, \\
\mathbf{f}_t^{(n)} &:= \sigma(\mathbf{W}^{f,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_{t-1}^{(n)}; 1]) \\
\mathbf{i}_t^{(n)} &:= \sigma(\mathbf{W}^{i,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_{t-1}^{(n)}; 1]) \\
\mathbf{c}_t^{(n)} &:= \mathbf{f}_t^{(n)} \otimes \mathbf{c}_{t-1}^{(n)} \\
&\quad + \mathbf{i}_t^{(n)} \otimes \tanh\left(\mathbf{W}^{c,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; 1]\right), \\
\mathbf{o}_t^{(n)} &:= \sigma(\mathbf{W}^{o,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_t^{(n)}; 1]) \\
\mathbf{h}_t^{(n)} &:= \mathbf{o}_t^{(n)} \otimes \tanh(\mathbf{c}_t^{(n)}), \\
\tilde{\mathbf{y}}_t &:= \mathbf{W}^{(N+1)}[\mathbf{h}_t^{(N)}; 1].
\end{aligned}
$$

- Can learn long-term dependencies without blowing up input layer → More concise model
- Context size depends on history → useful for varying acoustic conditions

DAE training

- Training tasks:
  - 1-channel system: Map REVERB multi-condition training set to WSJCAM0 clean training set
  - 8-channel system: Map CSP+DS processed REVERB multi-condition training set to WSJCAM0 clean tr. set
- Dimension:
  - 1-channel: 3 bidirectional LSTM layers w/ 128 units
  - 8-channel: 2 bidirectional LSTM layers w/ 128 units
- Stochastic gradient descent with momentum and input noise
- Parallel GPU training in mini-batch learning
  - CURRENNT toolkit (http://currennt.sf.net)

Baseline recognizer

- ASR features:
  - 23 Mel filterbank outputs
  - 13 MFCCs (0-12)
  - Mean normalized Log Mel features → gain-independent
- Re-implemented REVERB HTK baseline in Kaldi toolkit
- Improvements:
  - LDA-STC (MLLT) instead of $\Delta + \Delta\Delta$
    - Feature-level context
  - Basis fMLLR adaptation *per utterance*
    - Similar or better performance than fMLLR with less adaptation data

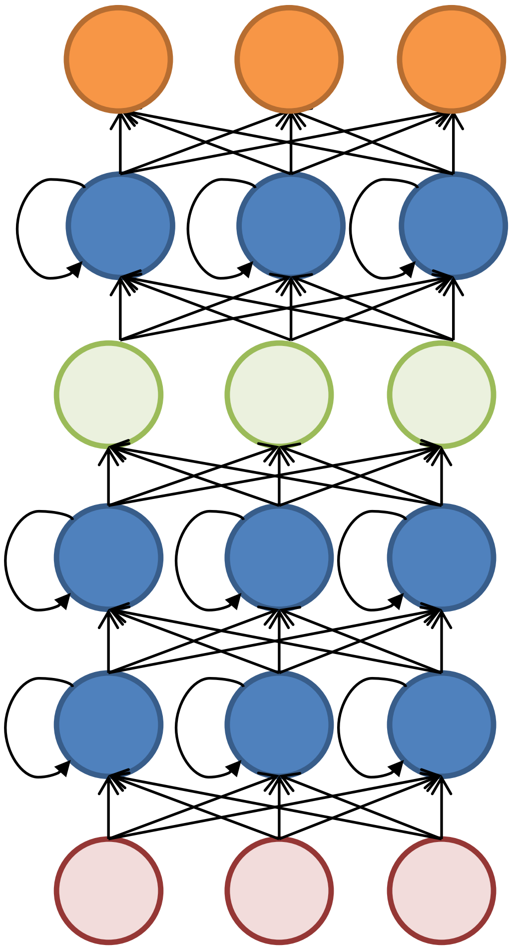
Baseline improvements (2)

- Discriminative training of GMM-HMM
  - Boosted MMI criterion:

$$f_b(\lambda) = \log \sum_u \frac{p(\mathbf{X}^u|\lambda, h^{w_u^*})^\alpha p_L(w_u^*)}{\sum_{w_u} p(\mathbf{X}^u|\lambda, h^{w_u})^\alpha p_L(w_u) e^{-b\varrho(w_u, w_u^*)}}$$

- Tri-gram language model
- Minimum Bayes Risk (MBR) decoding
  - Don't choose hypothesis far from the N-best
  - Minimize expected WER instead of SER (in case of MAP)

DRNN acoustic modeling



$\boldsymbol{b}_t$

Estimated phoneme posteriors

$\tilde{\mathbf{y}}_t \approx f(\mathbf{y}_t)$

DAE pre-training

$\tilde{\mathbf{x}}_t = f(\mathbf{x}_t)$

Multi-Stream DRNN+GMM-HMM

- Tandem decoding approach
- Discrete DRNN phoneme prediction:

$$b_t = \arg\max_i \tilde{y}_{t,i}$$

- Multi-stream emission probability:

$$p(\mathbf{x}_t, b_t | s_t) = p(\mathbf{x}_t | s_t)^{\mu} p(b_t | s_t)^{2-\mu}$$

- Stream weight μ for GMM likelihood of acoustic feature vector $\mathbf{x}_t$
- DRNN phoneme confusions modeled by $p(b_t/s_t)$

Baseline ASR results

	SimData	RealData
REVERB baselines (HTK)		
Clean	51.86	88.38
Multi-condition	28.94	52.29
fMLLR	25.16	47.23
Our baselines (Kaldi)		
Clean	51.23	88.81
Multi-condition	28.62	54.04
Basis fMLLR	23.60	47.14

Baseline ASR results (2)

	SimData	RealData
Our baselines (Kaldi)		
Clean	51.23	88.81
Multi-condition	28.62	54.04
Basis fMLLR	23.60	47.14
+LDA-STC	19.42	41.42
+DT	15.53	40.60
+Tri-gram	12.28	31.05
+MBR	**12.05**	**30.73**

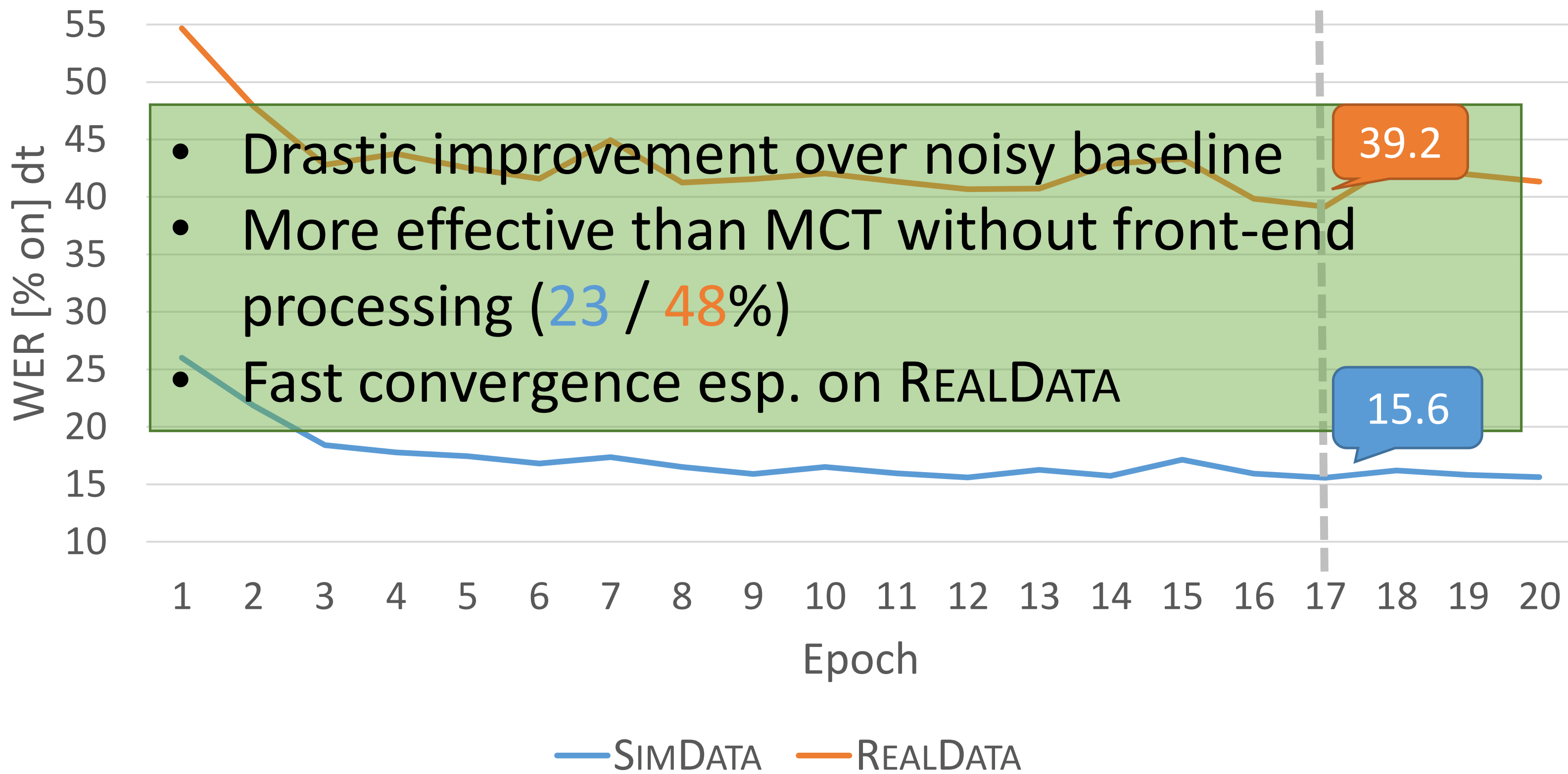
Kaldi recipe available on REVERB homepage

DRNN enhancement training epochs

Clean recognizer, LDA-STC, ML trained, Trigram

Base: 43.4 / 89.6

Input: 1st channel

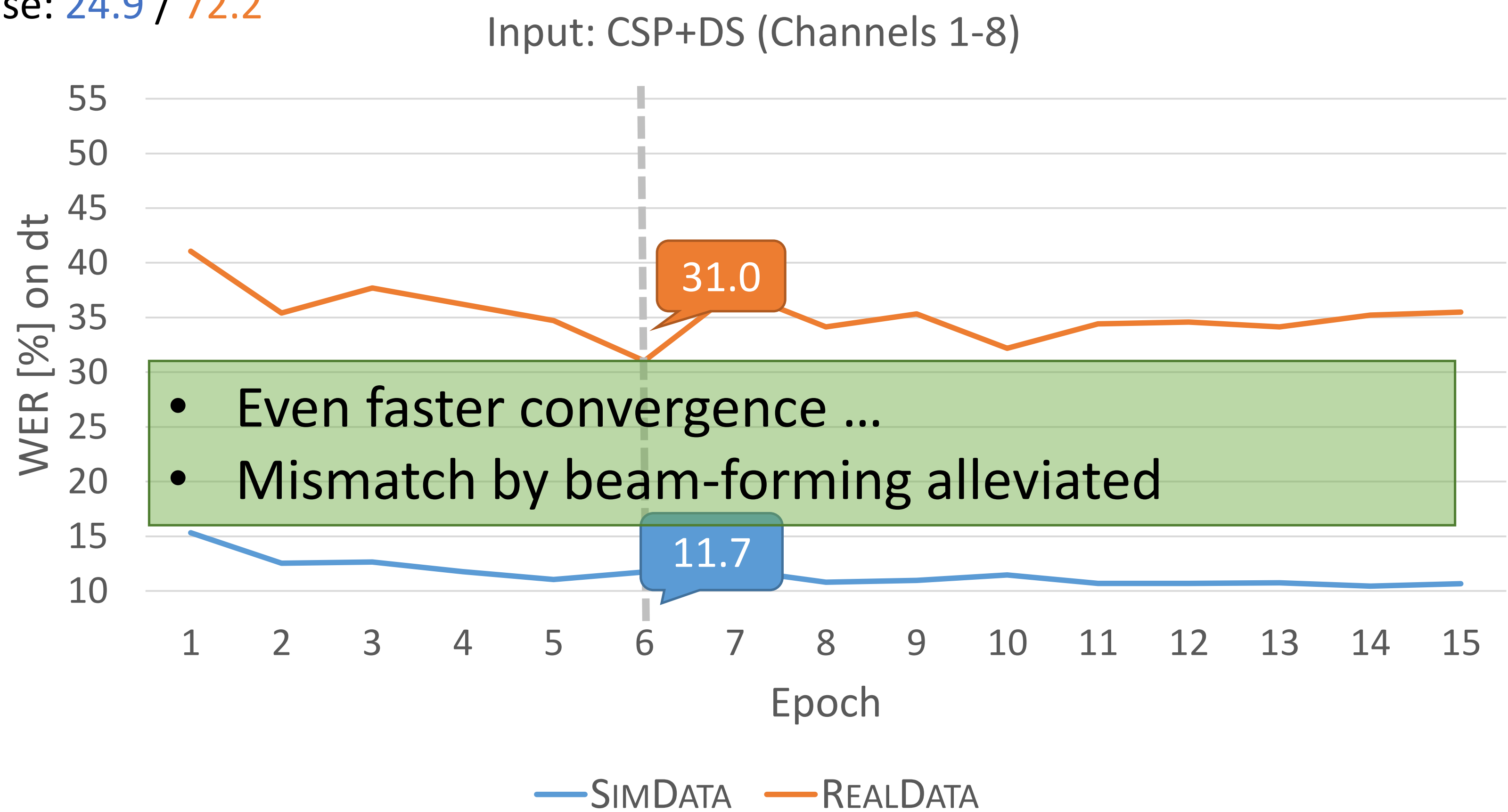


- Drastic improvement over noisy baseline
- More effective than MCT without front-end processing (23 / 48%)
- Fast convergence esp. on REALDATA

DRNN enhancement training epochs

Clean recognizer, LDA-STC, ML trained, Trigram

Base: 24.9 / 72.2



- Even faster convergence …
- Mismatch by beam-forming alleviated

Enhancement results: Clean training w/ fMLLR adaptation

# channels	DRNN enh.?	SimData	RealData
1	✗	33.2	77.8
1	✓	14.0	35.0
8	✗	16.4	54.5
8	✓	**9.7**	**26.5**
Oracle		6.0	10.1

Best result without using the multi-condition set!

Enhancement results: bMMI MCT recognizer

- Tuning of search parameters
- Discriminative training (booste
(processed) multi-condition se

Best result with single-channel front-end

# channels	DRNN enh.?	SIMDATA	REALDATA
1	✗	11.2	30.8
1	✓	10.4	26.3
8	✗	**7.5**	23.9
8	✓	7.7	**21.4**
Oracle		5.1	9.9

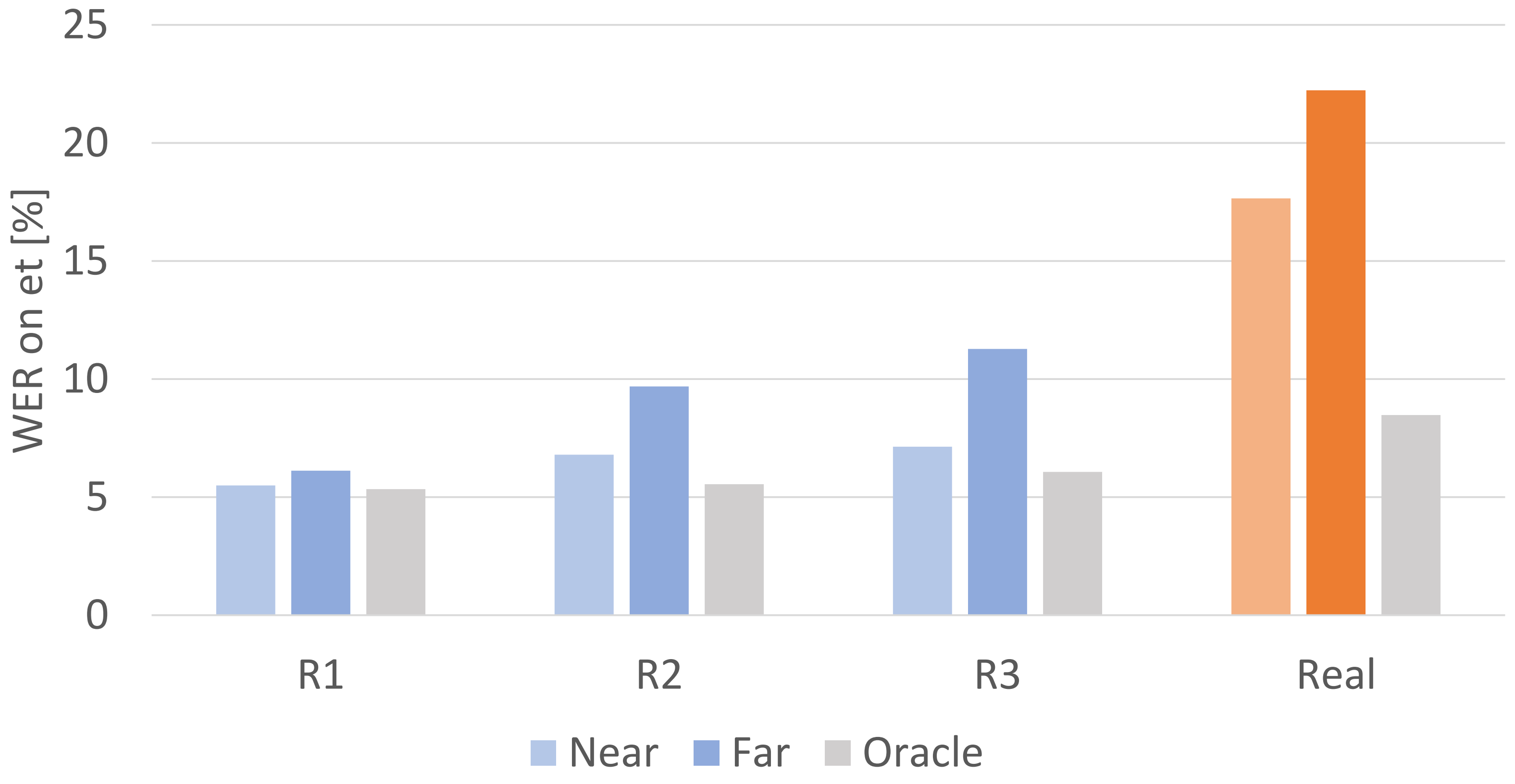
Test set evaluation: Enhancement, GMM-HMM AM

WER [%]	SimData	RealData
1-channel systems		
REVERB baseline	25.3	49.2
GMM-HMM	11.7	30.9
+ DRNN enh.	10.2	26.7
8-channel system		
+ CSP-DS	**7.8**	**20.1**

Test set evaluation: DRNN+GMM-HMM AM

WER [%]	SimData	RealData
DRNN+GMM-HMM	7.28	21.69
GMM-HMM w/ DRNN enh.	7.75	20.09
ROVER	**7.02**	**19.61**
GMM-HMM w/ Oracle enh.	5.65	8.47

Results with GMM-HMM and DRNN enhancement by room



Conclusions and Outlook

- Supervised training of de-reverberation with RNN is effective for ASR
  - Works on real data
  - Particularly promising for single-channel scenario
  - Can be efficiently combined with beam-forming
  - Some over-fitting observed (less than RNN-AM)
- Future work:
  - Effectiveness of supervised training for multi-channel de-reverberation
  - Use phase information

Thank you.

felix@weninger.de