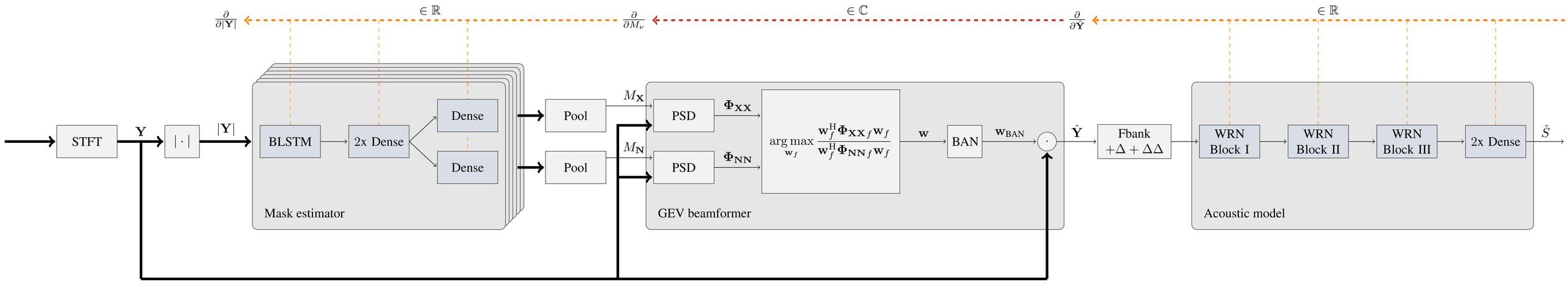
BEAMNET: End-to-End Training of a Beamformer-Supported Multi-Channel ASR System

Jahn Heymann, Lukas Drude, Christoph Boeddeker, Patrick Hanebrink, Reinhold Haeb-Umbach, University of Paderborn, Germany
{heymann, drude, haeb}@nt.uni-paderborn.de, http://www-nt.uni-paderborn.de

Computer Science, Electrical Engineering and Mathematics
Communications Engineering
Prof. Dr.-Ing. Reinhold Häb-Umbach

System overview



Highlights

- Complex valued backpropagation through statistical beamforming operation
  - Eliminates the need for parallel clean and noisy data and can be trained on real data only
  - Does not need heuristic masks as targets
  - Whole system works in the same STFT domain, no transformation between front-end and back-end necessary
- Agnostic to the array geometry
- Number of microphones can differ at test time

Acoustic model

- Hybrid approach, model estimates state posteriors
- Trained on whole utterances
- Based on Wide Residual Networks
  - Each block consists of two 2D convolutional layer and a residual connection
  - Number of channels increases with each block: $16 \rightarrow 80 \rightarrow 160 \rightarrow 320$
  - Normalization across time before each non-linearity

Backpropagation through GEV

- Most crucial step is to find the gradient for the PSD matrices given the gradient for the beamforming vector
- Generalized Eigenvalue problem:

$$\mathbf{\Phi}_{\mathbf{XX}}\mathbf{W} = \mathbf{\Phi}_{\mathbf{NN}}\mathbf{W}\mathbf{\Lambda}$$

- Beamforming vector is the eigenvector corresponding to the largest Eigenvalue
- Transform Generalized Eigenvalue problem to standard Eigenvalue problem

$$\mathbf{\Phi}\mathbf{W} = \mathbf{W}\mathbf{\Lambda} \quad \text{with } \mathbf{\Phi} = \mathbf{\Phi}_{\mathbf{NN}}^{-1}\mathbf{\Phi}_{\mathbf{XX}}$$

- Normalize the eigenvector to unit norm
- Gradient can now be calculated as

$$\frac{\partial J}{\partial \mathbf{\Phi}^*} = \mathbf{W}^{-\mathrm{H}} \left(\frac{\partial J}{\partial \mathbf{\Lambda}^*} + \mathbf{F}^* \circ \left(\mathbf{W}^\mathrm{H} \frac{\partial J}{\partial \mathbf{W}^*} \right) \right) \mathbf{W}^\mathrm{H} - \mathbf{W}^{-\mathrm{H}} \left(\mathbf{F}^* \circ \mathbf{W}^\mathrm{H}\mathbf{W} \left(\mathrm{Re}\left\{ \mathbf{W}^\mathrm{H} \frac{\partial J}{\partial \mathbf{W}^*} \right\} \circ \mathbf{I} \right) \right) \mathbf{W}^\mathrm{H}$$

with $F_{ij} = (\lambda_j - \lambda_i)^{-1}\ \forall_{i \neq j}$ and $F_{ii} = 0$

Results

- System was evaluated on the CHiME 4 dataset with different pre-training configurations:
  - **Fixed**: Model pre-trained separately. The parameters are kept fixed during joint training.
  - **Scratch**: Model parameters initialized randomly and the trained jointly.
  - **Finetune**: Model pre-trained separately. The parameters can be adjusted during joint training.

Training		Dev		Test	
BF	AM	real	simu	real	simu
BFIT+Kaldi		5.76	6.77	11.51	10.90
BFIT+WRN		5.53	6.67	9.44	10.18
fixed	fixed	4.26	4.29	5.85	4.59
scratch	scratch	5.51	5.19	8.76	5.61
scratch	finetune	4.14	4.09	5.86	4.06
fixed	finetune	4.09	3.96	5.56	**3.9**
finetune	finetune	**3.77**	**3.89**	**5.42**	3.95

ICASSP 2017, New Orleans

Work supported by DFG under contract no. Ha3455/11-1