Voice Controlled Hearing Assistive Devices

Iván López-Espejo

Centre for Acoustic Signal Processing Research (CASPR)

ivl@es.aau.dk

Wednesday 19th May, 2021

Overview

- Manual operation of **hearing assistive devices** (**HADs**) is cumbersome in a number of situations.

- To assist in addressing this issue, **voice interfaces** are envisioned as a means **for handling and operating HADs** in a practical manner.

- **Keyword spotting** (**KWS**) is the technology dealing with the identification of keywords in audio streams comprising speech.

- KWS can be applied to controlling HADs:
  1. Personalization (utilization of **user-specific aspects**, e.g., voice characteristics or head-related acoustics of the specific user).
  2. Robustness against noise.
  3. Low memory and low computational complexity.

- It is expected that this technology contributes to enhance the life quality of hearing-impaired people.

Deep Spoken Keyword Spotting

A Little of Context...

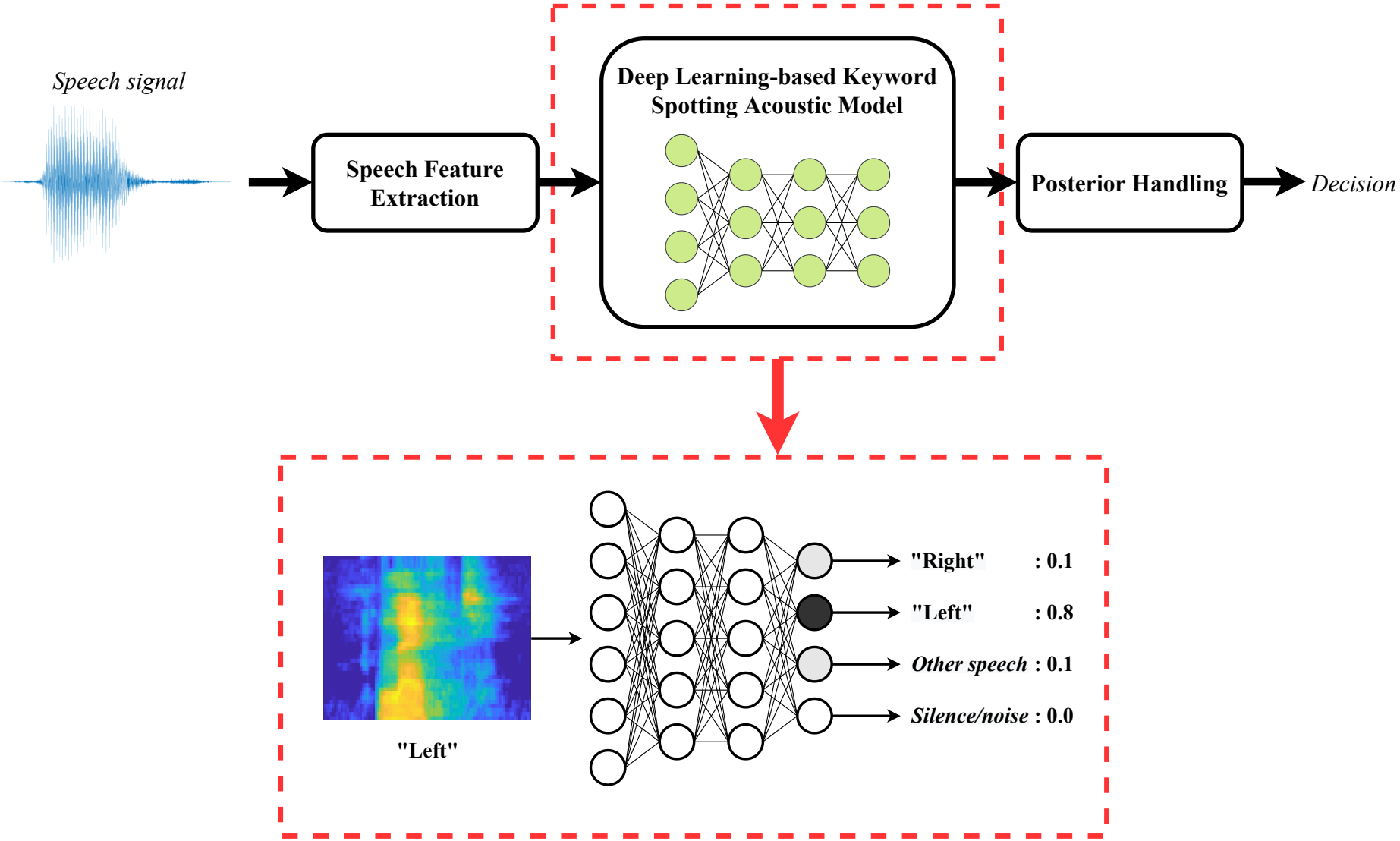
1. Large-vocabulary continuous speech recognition (LVCSR)-based KWS
   - Generation of rich lattices (high computational resources and latency).
2. Keyword/filler hidden Markov model (HMM)-based KWS (∼89-90)
   - Lighter alternative to LVCSR.
   - Viterbi decoding is still required.
3. **Deep spoken KWS** (2014[1])
   - Simple posterior handling instead of Viterbi decoding.
   - DNN complexity adjustable to fit the computational constraints.
   - Better KWS performance.

Deep spoken KWS is very appealing to deploy popular KWS technology on a variety of consumer electronics with limited resources!

[1] G. Chen *et al.*, "Small-footprint keyword spotting using deep neural networks", in *Proc. of ICASSP 2014*

Deep Spoken Keyword Spotting

General Pipeline



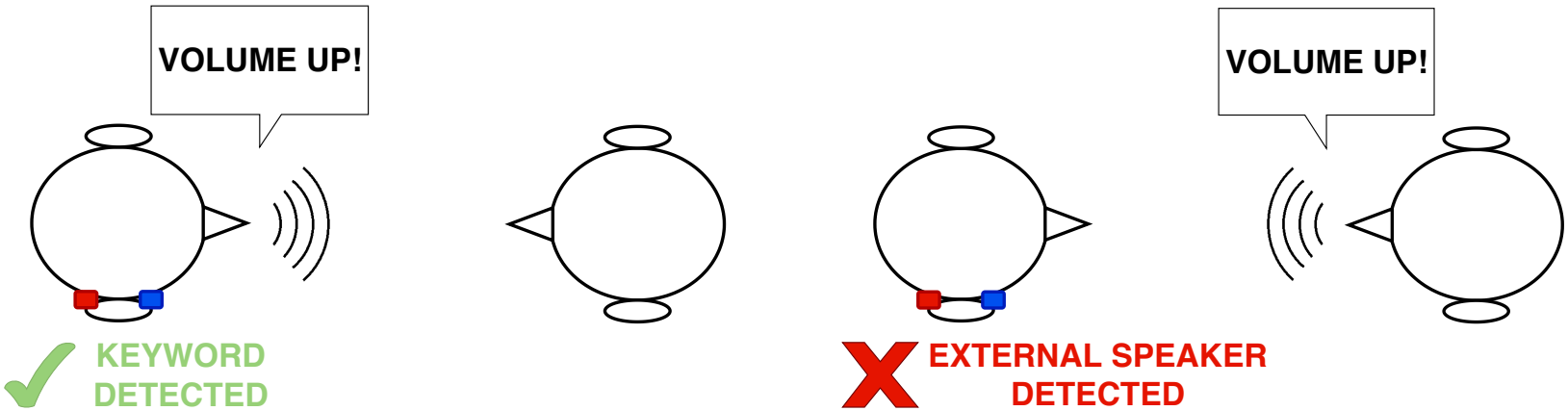
Paths Explored in CASPR

1. **Personalization for HADs:**
   - I. López-Espejo, Z.-H. Tan and J. Jensen, "Keyword Spotting for Hearing Assistive Devices Robust to External Speakers", in *Proc. of INTERSPEECH 2019*, Graz (Austria), 2019.
   - I. López-Espejo, Z.-H. Tan and J. Jensen, "Improved External Speaker-Robust Keyword Spotting for Hearing Assistive Devices", *IEEE Transactions on Audio, Speech and Language Processing*, 2020.

2. **Speech representation learning:**
   - I. López-Espejo, Z.-H. Tan and J. Jensen, "Exploring Filterbank Learning for Keyword Spotting", in *Proc. of EUSIPCO 2020*, Amsterdam (The Netherlands), 2021.

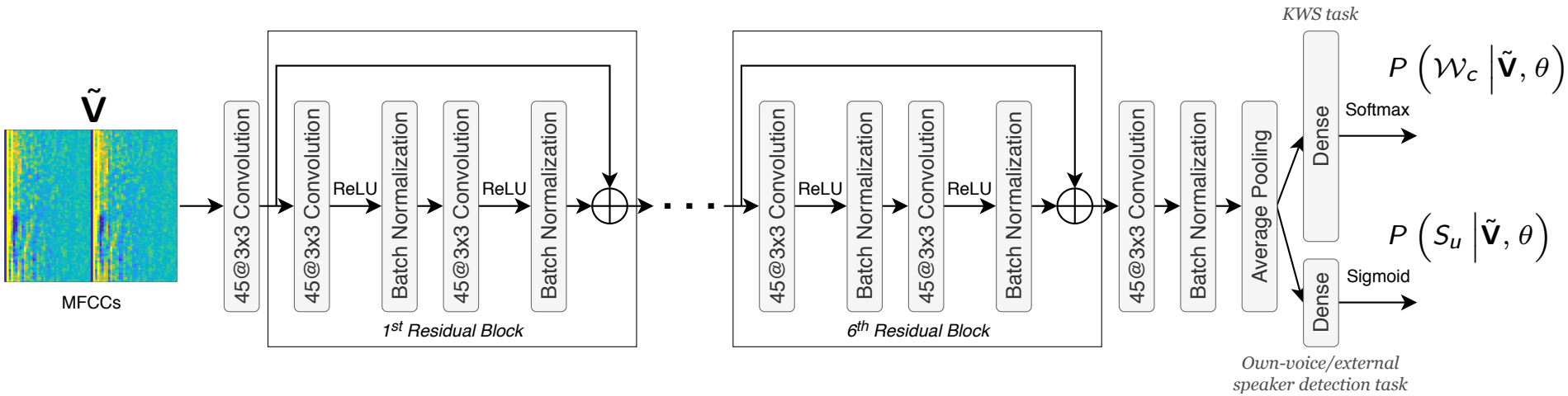
3. **Robustness against acoustic noise:**
   - I. López-Espejo, Z.-H. Tan and J. Jensen, "A Novel Loss Function and Training Strategy for Noise-Robust Keyword Spotting", *submitted to IEEE Transactions on Audio, Speech and Language Processing*.

- **KWS** systems **for HADs must be robust against external speakers**, that is, the user must be the only one allowed to trigger actions on her/his HAD.



- **We proposed HAD user (speaker)-dependent KWS** drawing from a state-of-the-art small-footprint KWS system based on deep residual learning and dilated convolutions (`res15`[2]).

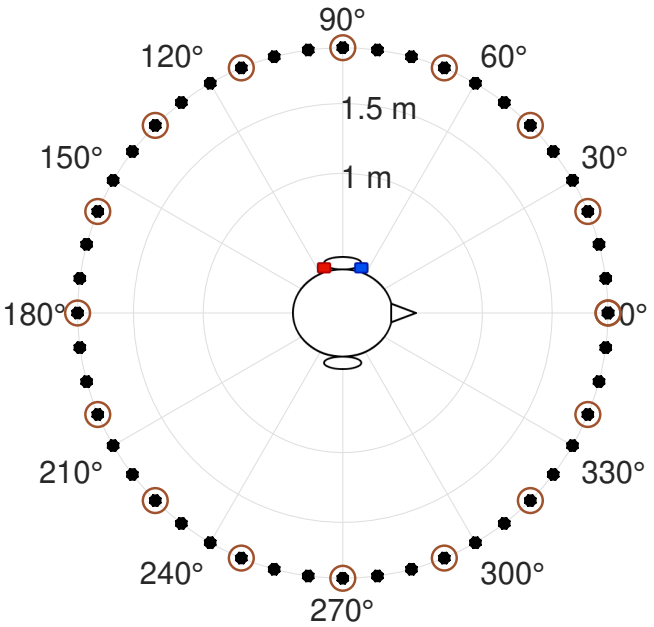
[2] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting", in *Proc. of ICASSP 2018*, Calgary (Canada), 2018

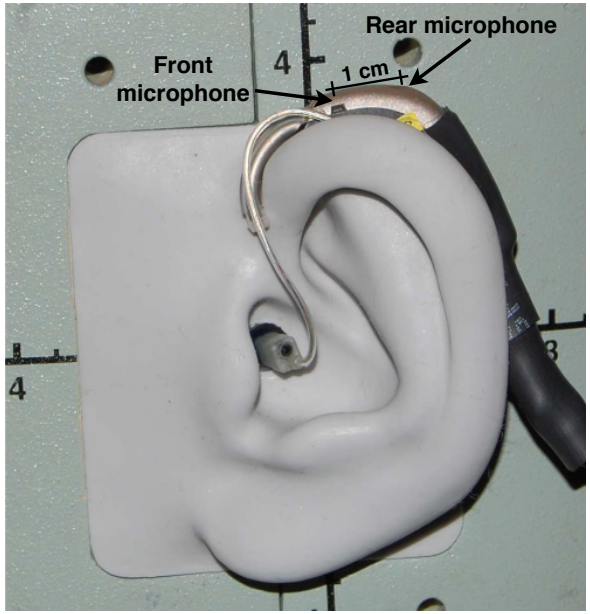


- **Two tasks:** KWS and own-voice/external speaker detection.
- The sigmoid layer outputs a probability $P\left(S_u \middle| \tilde{\mathbf{V}}, \theta\right)$ that the input $\tilde{\mathbf{V}}$ corresponds to an utterance said by the HAD user S_u.
- KWS prediction $P\left(\mathcal{W}_c \middle| \tilde{\mathbf{V}}, \theta\right)$ from $\tilde{\mathbf{V}}$ is considered if $P\left(S_u \middle| \tilde{\mathbf{V}}, \theta\right) > P_{THR}$ ($P_{THR} = 0.5$).

Personalization

Experimental Framework

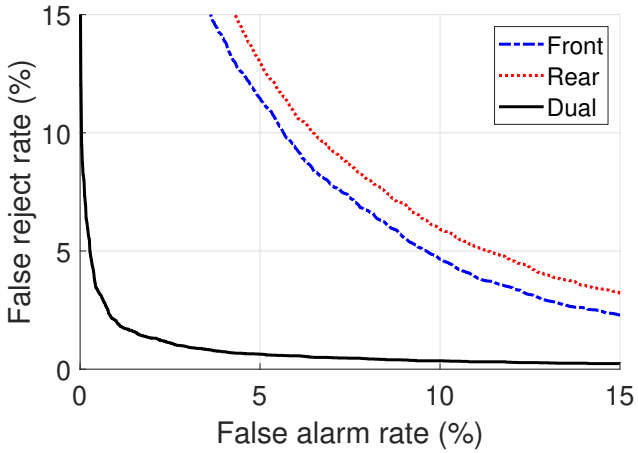




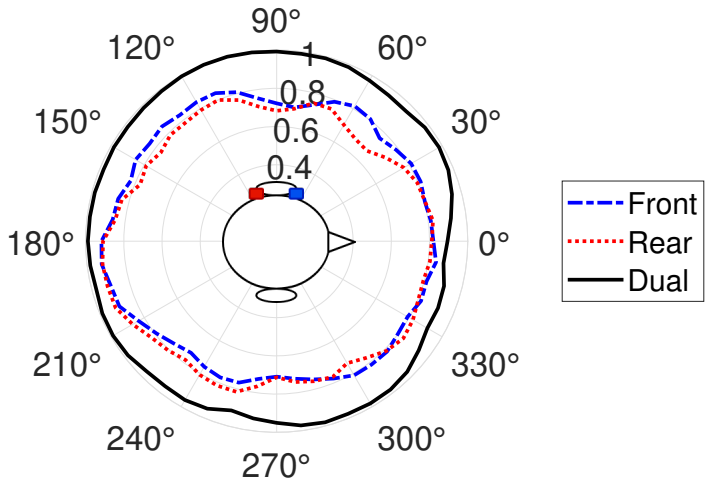
- We created a **two-microphone hearing aid speech database** from the Google Speech Commands Dataset (**GSCD**).
- HAD user own-voice signals were generated by filtering 75% of the GSCD through a single own-voice transfer function (**OVTF**).
- External speaker signals were created by filtering the remaining 25% of the GSCD through head-related transfer functions (**HRTFs**).
- Apart from the *unknown word* class, **10 keywords** were considered: "yes", "no", "up", "down", "left", "right", "on", "off", "stop" and "go".

- **Accuracy results (%)** with 95% confidence intervals:

				Own-voice/External speaker detection			Keyword spotting	
	Architecture	**Training data**	**Input type**	*Own-voice subset*	*External speaker subset*	*Overall*	*Own-voice subset*	*Overall*
Baseline	`res15` (KWS only)	Own voice	Front and rear mics	—	—	—	94.21 ± 0.39	71.87 ± 0.30
Front	Multi-task	Own and external voice	Front mic	97.49 ± 1.02	80.38 ± 5.23	93.02 ± 0.76	94.28 ± 0.37	89.48 ± 0.74
Rear	Multi-task	Own and external voice	Rear mic	97.28 ± 1.08	79.03 ± 5.06	92.51 ± 0.68	94.48 ± 0.25	89.29 ± 0.55
Dual	Multi-task	Own and external voice	Front and rear mics	99.60 ± 0.22	96.22 ± 1.61	98.72 ± 0.29	94.59 ± 0.32	94.86 ± 0.39



DET curves for own-voice/external speaker detection.



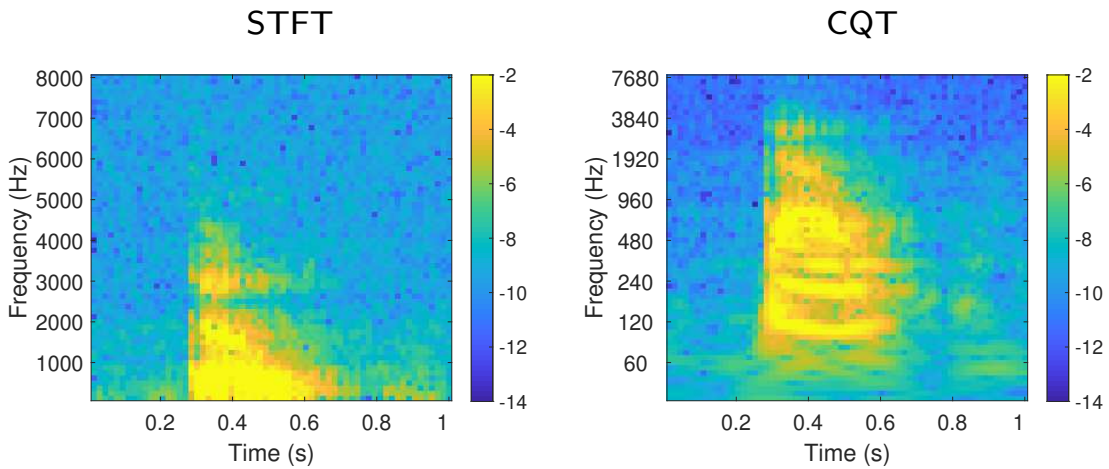
External speaker detection accuracy as a function of the angle of external speakers.

- The OVTF and HRTFs are more similar (in terms of MFCC Euclidean distance) at angles where we see a relative drop in performance.

- While the created two-microphone hearing aid speech database comprises speech signals uttered by many different speakers, impulse responses for its generation were only measured on a single actual person.
- Impulse responses are user-dependent, as these characterize physical features, e.g., head size and shape.
- We created a new speech corpus with impulse responses measured on multiple persons wearing a hearing aid: **multi-user database**.
- **Problem!** Performance loss in terms of KWS accuracy: from 94.86% $\pm$ 0.39 to 80.45% $\pm$ 0.55.

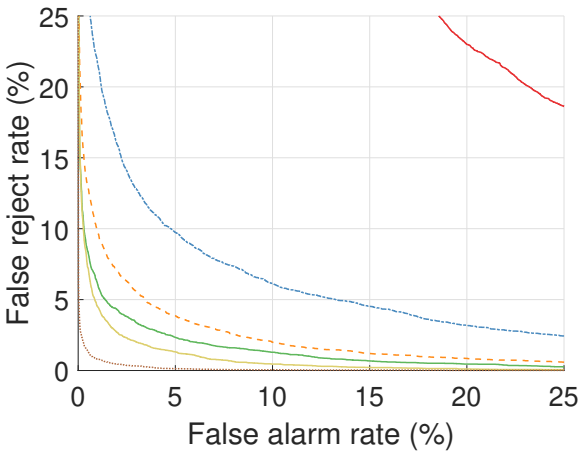
Towards reducing the performance loss:

- The relative position of the users' mouth w.r.t. the hearing aid microphones is virtually time-invariant and different from that of an external speaker:
  - Spectral magnitude features for KWS.
  - Phase difference information (**GCC-PHAT**-based coefficients) for own-voice/external speaker detection.
- Use of the perceptually-motivated **constant-Q transform**: at lower (higher) frequencies the frequency (time) resolution is higher.

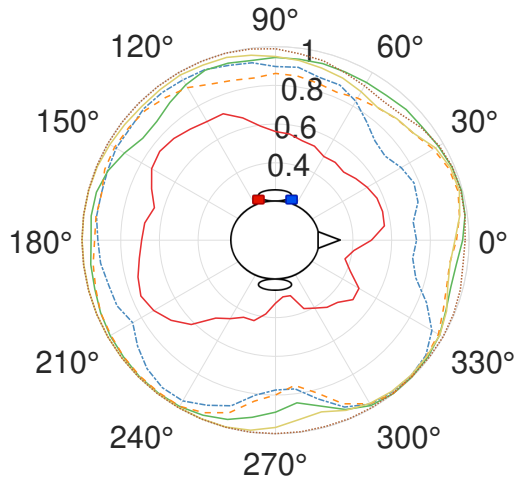


Accuracy results (%) with 95% confidence intervals:

		Own-voice/External speaker detection			Keyword spotting	
		Own-voice subset	*External speaker subset*	*Overall*	*Own-voice subset*	*Overall*
Multi-user database	Baseline	—	—	—	93.81 ± 0.27	73.88 ± 0.23
	MFCC-80×1	92.64 ± 1.39	55.36 ± 4.43	84.26 ± 0.45	93.27 ± 0.30	**80.45 ± 0.55**
	MFCC-40×2	97.03 ± 1.81	87.18 ± 2.06	94.81 ± 1.20	94.32 ± 0.21	90.78 ± 1.16
	STFT-S	98.60 ± 0.95	95.03 ± 1.10	97.80 ± 0.53	94.30 ± 0.34	93.59 ± 0.64
	CQT-S	98.44 ± 0.87	92.12 ± 2.39	97.02 ± 0.44	94.60 ± 0.31	93.19 ± 0.52
	STFT-S+GCC	98.61 ± 1.30	96.40 ± 1.21	98.11 ± 0.93	94.23 ± 0.57	93.77 ± 0.99
	CQT-S+GCC	99.49 ± 0.47	98.67 ± 0.36	99.31 ± 0.33	94.81 ± 0.26	**95.34 ± 0.32**



DET curves for own-voice/external speaker detection.



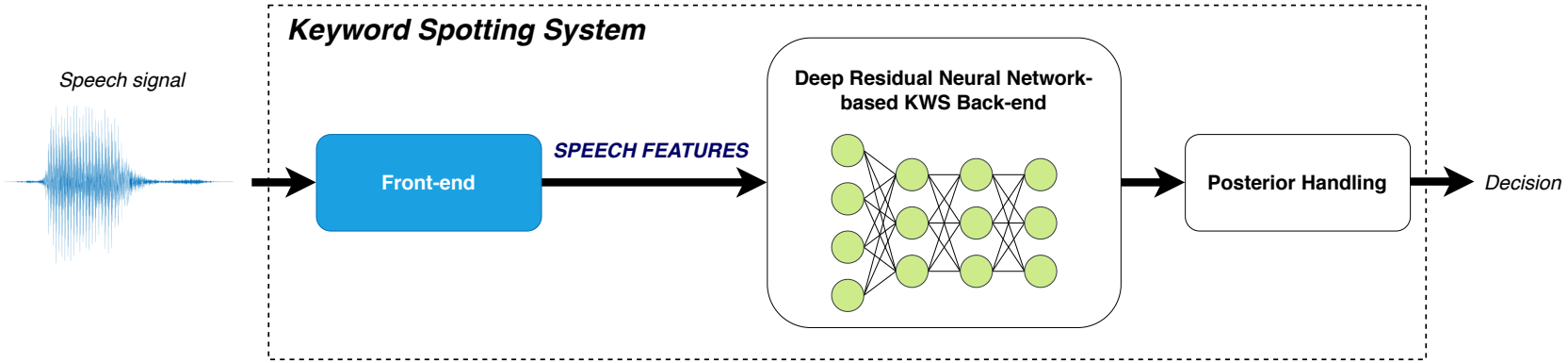
External speaker detection accuracy as a function of the angle of external speakers.

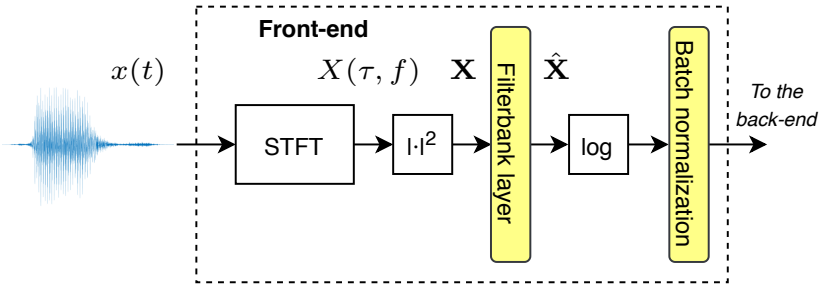
MFCC-80x1 MFCC-40x2 STFT-S CQT-S STFT-S+GCC CQT-S+GCC

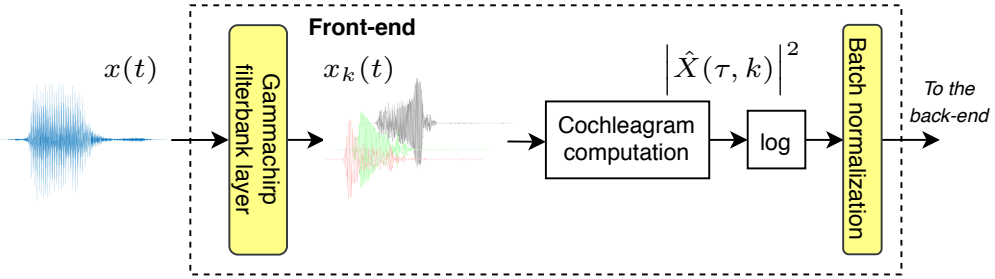
- Handcrafted speech features like MFCCs are not necessarily optimal for any particular speech processing task.
- **Recent trend:** development of end-to-end deep learning systems where the feature extraction process is optimal according to the task and training criterion.
- We explored the possible advantages of employing *learned filterbanks over handcrafted speech features* for KWS.

Speech Representation Learning

Methods







- Filterbank matrix weight learning in the power spectral domain
- Psychoacoustically-motivated gammachirp filterbank parameter learning

For both front-ends, the learnable filterbank parameters are optimized by backpropagation jointly with the KWS back-end!

We used the Google Speech Commands Dataset to experiment (**10 keywords**)

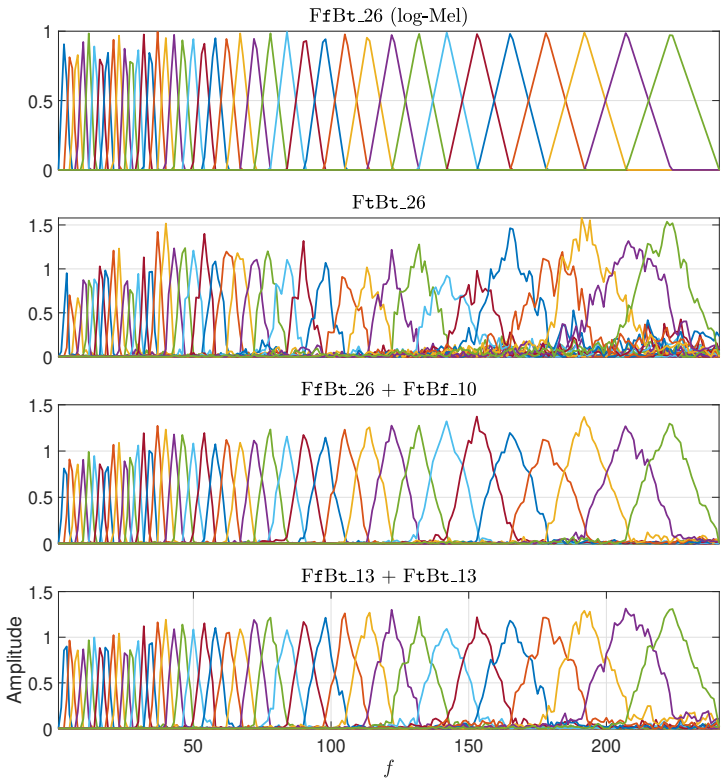
Accuracy results (%) with 95% confidence intervals.

- Filterbank matrix learning:

Test	Accuracy (%)
FfBt_26 (log-Mel)	95.64 ± 0.33
FtBt_26	95.73 ± 0.24
FfBt_26 + FtBf_10	95.73 ± 0.38
FfBt_13 + FtBt_13	95.30 ± 0.82

- Gammachirp filterbank learning:

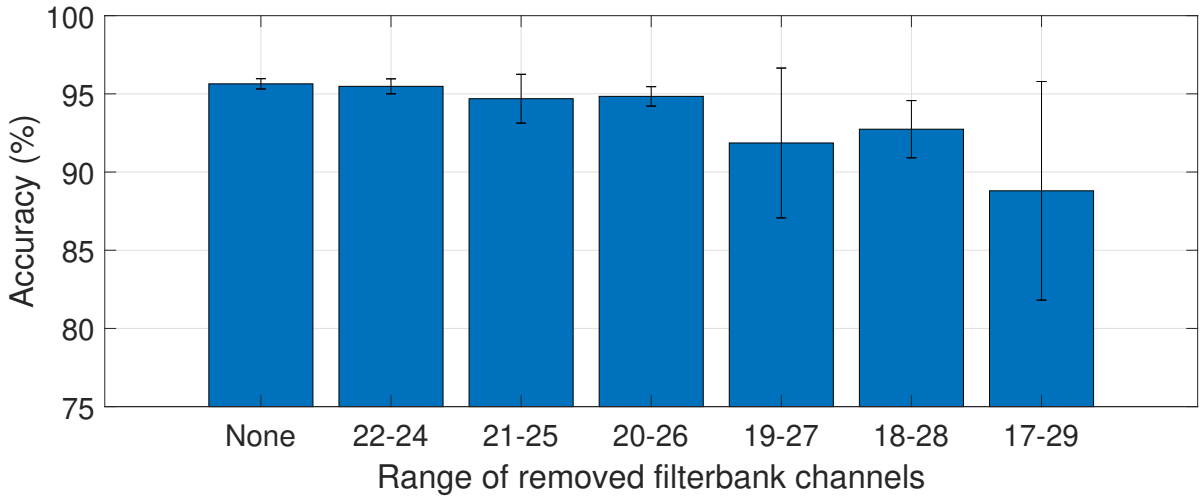
Test	Accuracy (%)	n	b	c
GT[f]_Ic-Mel	95.47 ± 0.36	4	1.019	0
GC[f]_Ic-Mel	95.45 ± 0.58	4	1.019	-1
GC[t]_Ic-Mel	95.12 ± 0.42	4.69 ± 0.07	0.976 ± 0.015	-0.84 ± 0.05
GC[t]_Ic-Linear	95.19 ± 0.52	4.44 ± 0.05	0.866 ± 0.019	-0.88 ± 0.02
GC[t]_Ir-Mel	94.68 ± 0.52	4.90 ± 0.51	0.976 ± 0.115	-0.97 ± 0.32
GC[t]_Ir-Linear	94.93 ± 0.45	4.65 ± 0.41	0.861 ± 0.075	-0.98 ± 0.38



In general, there are **no** statistically significant differences between using a learned filterbank and handcrafted speech features ⇒ the latter are still a good choice when employing modern KWS back-ends!

- Is the filterbank and, in general, the speech feature design actually a crucial part of modern KWS systems?

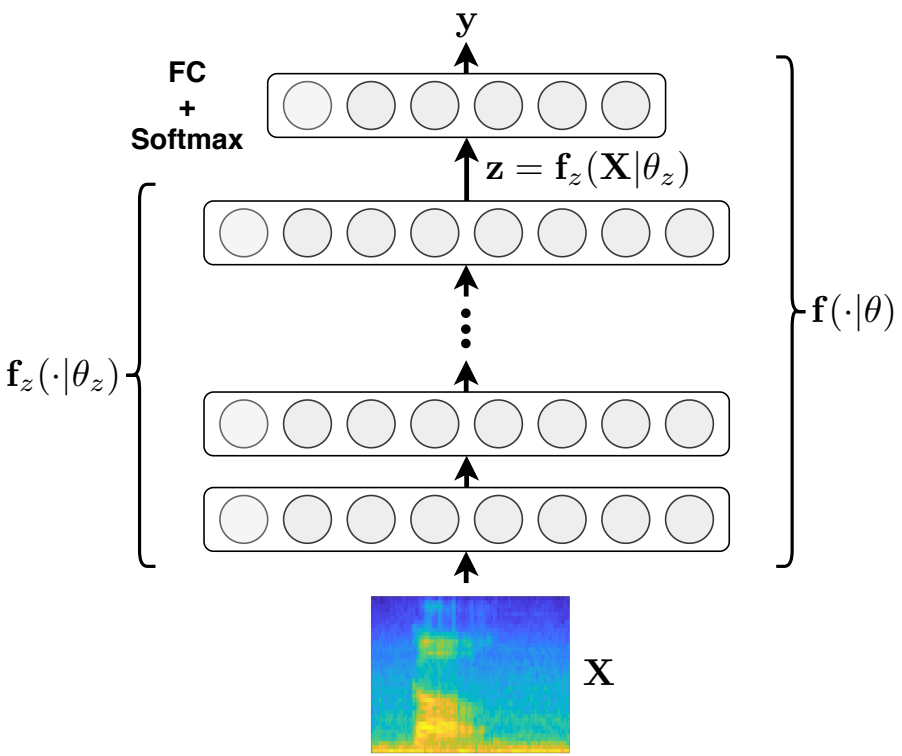
Using log-Mel features:



$f_{k=23} \approx 2,000$ Hz, $\quad [20, 26] \equiv [1,626,\ 2,564]$ Hz

KWS systems are fed with a great amount of **redundant information** $\Rightarrow$ new possibilities in the field of *small-footprint* KWS regarding the design of much more compact speech features!

- The development of KWS systems that are accurate in **noisy conditions** remains a challenge.

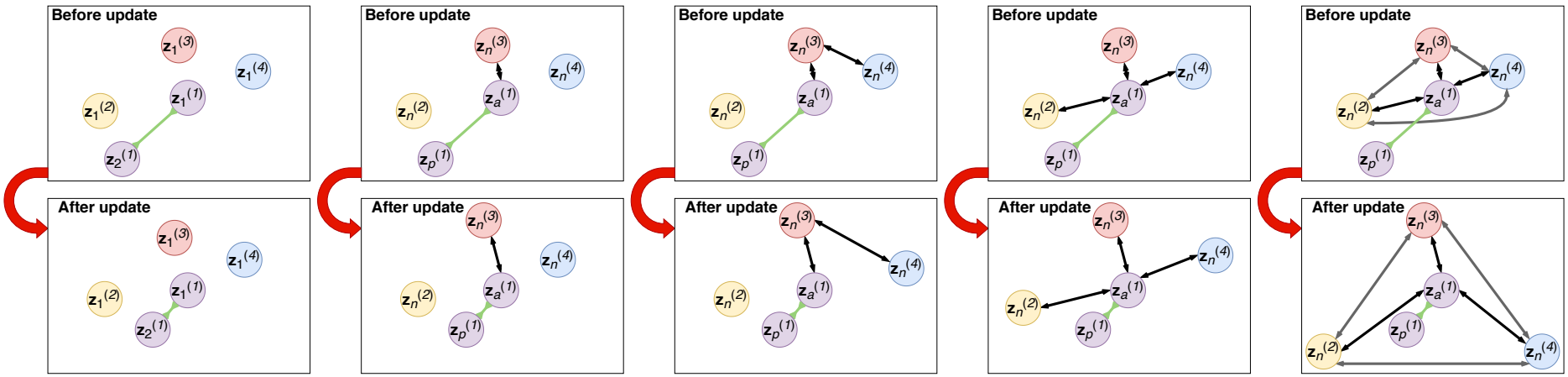


- We interpret **z** as linearly classifiable keyword embeddings.

Two-stage training strategy:

1. The keyword embedding extractor $\mathbf{f}_z(\cdot|\theta_z)$ is multi-condition trained by considering a new $(C_{N,2}+1)$-pair loss function.
2. **FC + Softmax** is trained using cross-entropy loss and multi-condition keyword embeddings extracted by $\mathbf{f}_z(\cdot|\theta_z)$.

- We suggested a $(C_{N,2}+1)$-pair loss function extending the concept behind other tuple-based loss functions like triplet and N-pair losses.
- The $(C_{N,2}+1)$-pair loss can achieve larger inter-class and smaller intra-class variation $\Rightarrow$ the *generalization ability* of an embedding extractor can be improved.



(a) Contrastive loss (b) Triplet loss (c) Quadruplet loss (d) N-pair loss (e) $(C_{N,2}+1)$-pair loss

Robustness Against Noise

Results

We used a **noisy version** of the Google Speech Commands Dataset to experiment (**10 keywords**) and the **same deep residual learning model** as before

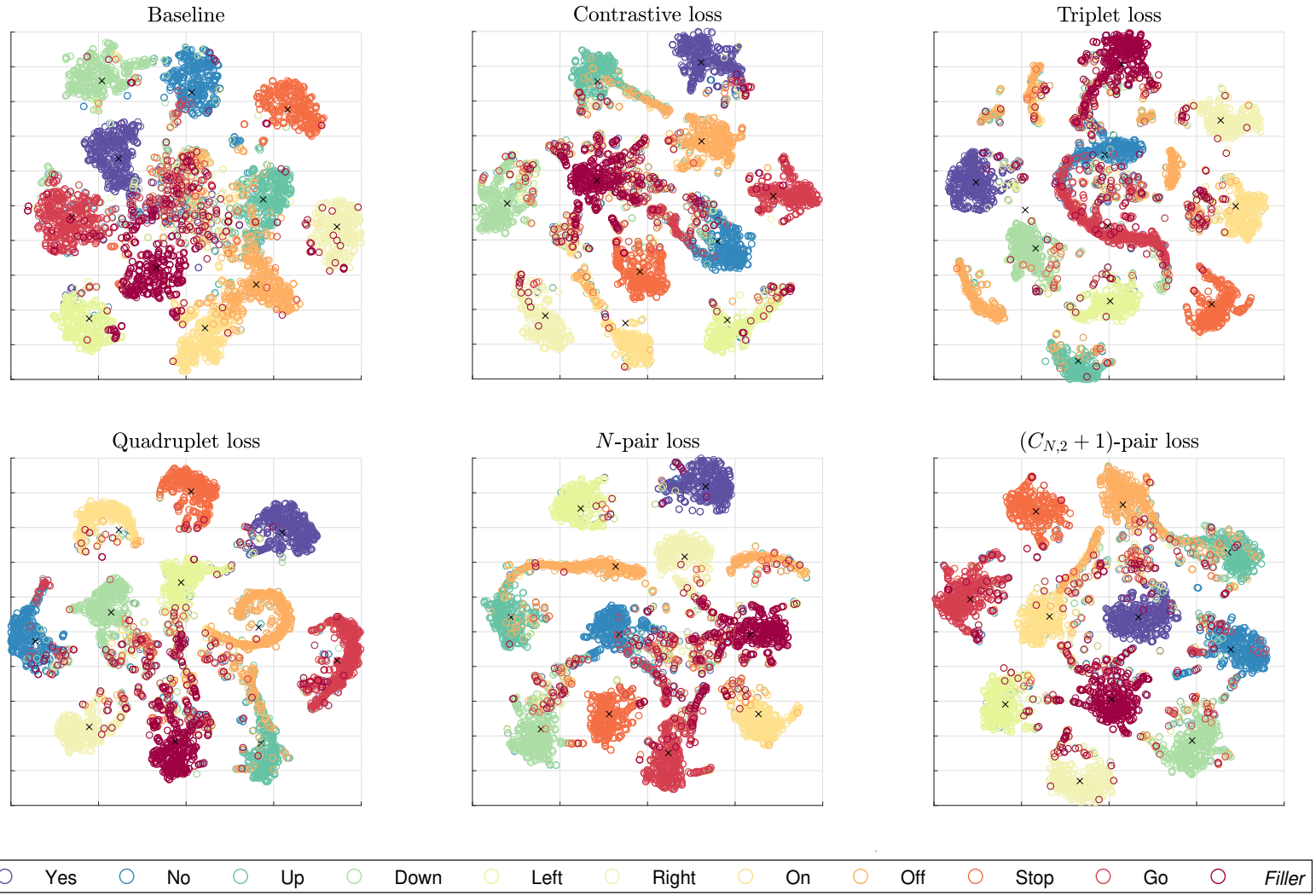
- **Accuracy results (%)** with 95% confidence intervals:

		SNR (dB)								
		-10	-5	0	5	10	15	20	Clean	Average
Seen noises	Baseline	57.51 ± 0.96	75.94 ± 1.42	88.36 ± 1.30	92.90 ± 0.43	94.59 ± 0.75	95.17 ± 0.74	96.09 ± 0.72	96.45 ± 0.87	87.13 ± 0.60
	Contrastive loss	59.73 ± 1.46	75.07 ± 1.74	86.81 ± 0.83	92.56 ± 0.64	94.37 ± 0.29	95.14 ± 0.31	95.85 ± 0.80	96.72 ± 0.36	87.03 ± 0.65
	Triplet loss	60.12 ± 2.68	75.70 ± 2.07	86.96 ± 0.78	93.04 ± 0.72	93.86 ± 0.78	95.14 ± 0.87	96.01 ± 0.76	96.67 ± 0.42	87.19 ± 0.91
	Quadruplet loss	60.36 ± 1.47	78.00 ± 1.17	87.85 ± 1.17	92.80 ± 0.66	94.40 ± 0.90	95.58 ± 0.59	96.21 ± 0.48	96.88 ± 0.94	87.76 ± 0.68
	N-pair loss	60.07 ± 1.73	77.29 ± 0.95	88.72 ± 0.61	93.41 ± 0.50	95.00 ± 0.53	95.92 ± 0.22	96.62 ± 0.64	97.00 ± 0.44	88.00 ± 0.16
	$(C_{N,2}+1)$-pair loss	61.84 ± 1.73	78.00 ± 1.97	88.82 ± 0.41	93.48 ± 0.49	95.17 ± 0.82	95.94 ± 0.57	96.76 ± 0.56	96.91 ± 0.57	88.36 ± 0.72
Unseen noises	Baseline	35.11 ± 1.87	62.06 ± 1.40	82.20 ± 1.91	89.75 ± 1.18	92.87 ± 0.92	95.11 ± 1.14	96.11 ± 0.43	96.52 ± 0.64	81.22 ± 0.69
	Contrastive loss	38.98 ± 1.74	62.61 ± 2.71	81.19 ± 1.76	89.02 ± 0.87	92.50 ± 1.22	94.29 ± 0.43	95.38 ± 0.84	96.61 ± 0.32	81.32 ± 0.86
	Triplet loss	37.78 ± 2.44	62.44 ± 1.14	81.84 ± 1.57	90.13 ± 0.57	93.45 ± 0.47	95.28 ± 0.37	96.01 ± 0.58	96.74 ± 0.59	81.71 ± 0.52
	Quadruplet loss	38.26 ± 0.95	63.94 ± 1.37	83.02 ± 1.03	90.11 ± 0.70	93.88 ± 0.61	95.26 ± 0.44	95.94 ± 0.59	96.78 ± 0.48	82.15 ± 0.27
	N-pair loss	39.93 ± 2.54	64.86 ± 1.89	83.92 ± 0.68	90.50 ± 0.27	93.81 ± 0.64	95.28 ± 0.50	96.30 ± 0.64	96.98 ± 0.54	82.70 ± 0.52
	$(C_{N,2}+1)$-pair loss	41.21 ± 2.63	65.27 ± 1.10	84.67 ± 0.99	92.26 ± 0.15	94.70 ± 0.48	96.20 ± 0.87	96.71 ± 0.49	97.19 ± 0.59	83.53 ± 0.30

- Average **intra- and inter-class Euclidean distances** with 95% confidence intervals:

	Intra-class distance			*Inter-class distance*		
	Training	Test: *Seen noises*	Test: *Unseen noises*	Training	Test: *Seen noises*	Test: *Unseen noises*
Baseline	0.327 ± 0.001	0.423 ± 0.003	0.488 ± 0.003	1.272 ± 0.021	1.234 ± 0.021	1.197 ± 0.021
Contrastive loss	0.084 ± 0.001	0.235 ± 0.005	0.324 ± 0.006	1.482 ± 0.004	1.480 ± 0.004	1.477 ± 0.004
Triplet loss	0.031 ± 0.000	0.199 ± 0.005	0.280 ± 0.005	1.471 ± 0.022	1.470 ± 0.022	1.469 ± 0.022
Quadruplet loss	0.030 ± 0.000	0.196 ± 0.005	0.278 ± 0.005	1.473 ± 0.020	1.472 ± 0.020	1.471 ± 0.020
N-pair loss ($\mathcal{D}_E^2(\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2)$)	0.052 ± 0.000	0.236 ± 0.005	0.325 ± 0.005	1.482 ± 0.002	1.478 ± 0.002	1.474 ± 0.002
N-pair loss	0.023 ± 0.000	0.196 ± 0.005	0.278 ± 0.006	1.481 ± 0.009	1.481 ± 0.009	1.480 ± 0.009
$(C_{N,2}+1)$-pair loss	0.018 ± 0.000	0.191 ± 0.005	0.268 ± 0.006	1.483 ± 0.002	1.483 ± 0.002	1.482 ± 0.002

- Test set (*unseen noises* only) **keyword embedding representation**:



This training strategy and loss function...

1. ...can be applied to most of the latest (single- and multi-channel) KWS models.
2. ...do increase neither the number of parameters nor the number of multiplications of the model.
3. ...can potentially be useful to mitigate the effect of other types of distortions in addition to acoustic noise.
4. ...might be exported to other application areas such as image classification.

Thanks for your attention!