tinyML® Research Symposium

Enabling Ultra-low Power Machine Learning at the Edge

April 22, 2024

www.tinyML.org

Boosting keyword spotting through on-device learnable user speech characteristics

Cristian Cioflan ETH Zürich
Lukas Cavigelli Huawei Technologies
Luca Benini ETH Zürich, Università di Bologna

PULP Platform
Open Source Hardware, the way it should be!

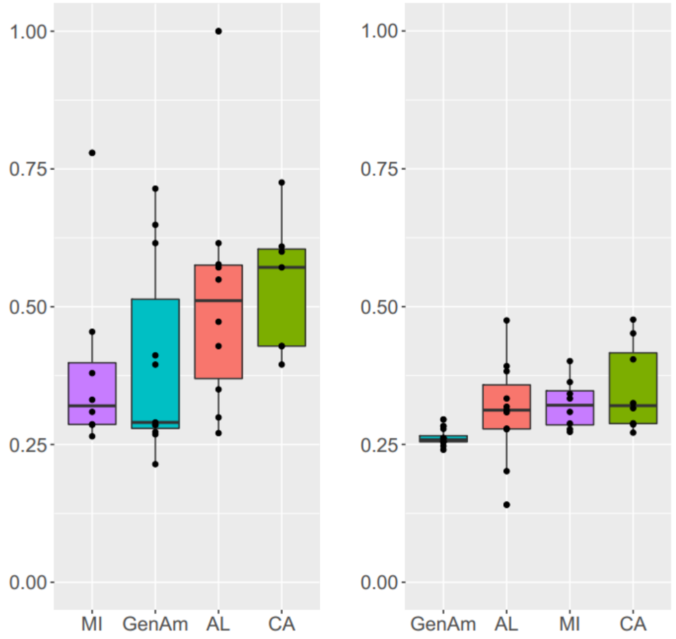
@pulp_platform
pulp-platform.org
youtube.com/pulp_platform

Keyword Spotting at the Extreme Edge

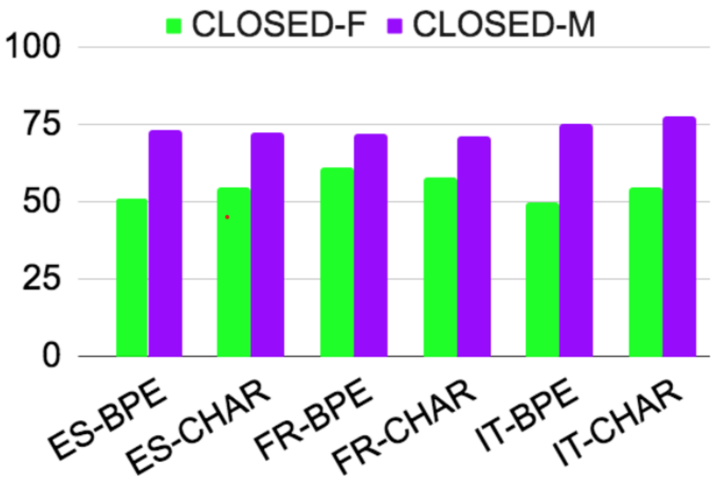
- **Voice-controlled personal assistants**
- **Drones controlled remotely to investigate hard-to-reach locations**
- **Hearing devices adapted to the environment conditions**

Accuracy degrades in real-world conditions

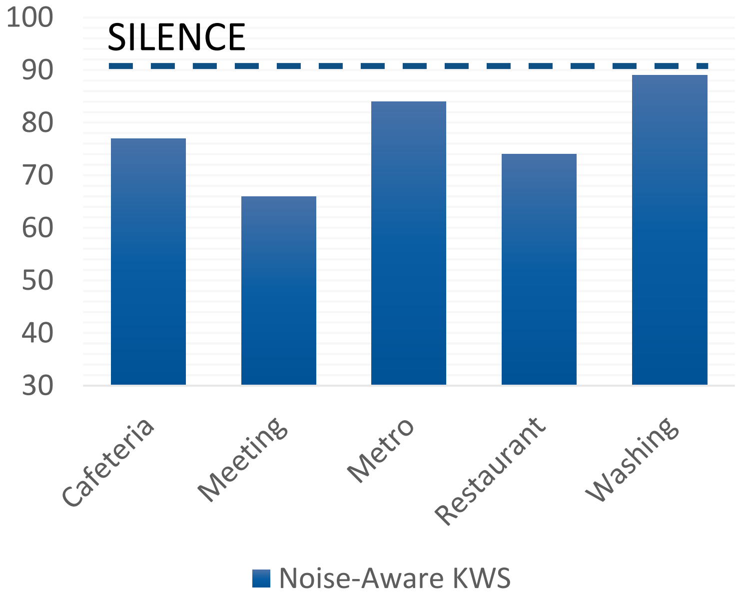
- **Unknown environments where pretraining (offline) ≠ target (online) data**
  - Domain shifts, differences in sensors, knowledge expansion
  - Accents, genders, background noises



WER variation by US region in Bing Speech (left) and YouTube Captions (right) [Tatman2017]



Feminine vs. masculine accuracy on function words for two speech translation model, in three languages [Savoldi2022]



Keyword spotting accuracy drops by 3%-26% compared to silent environments [Cioflan2024]

How to mitigate the performance degradation?

- **Server-side training on on-site data**
  - **Does not respect** privacy
  - **Communication reduces** device lifetime
  - **User-specific labeled data** **is scarce**

ETH zürich ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA HUAWEI

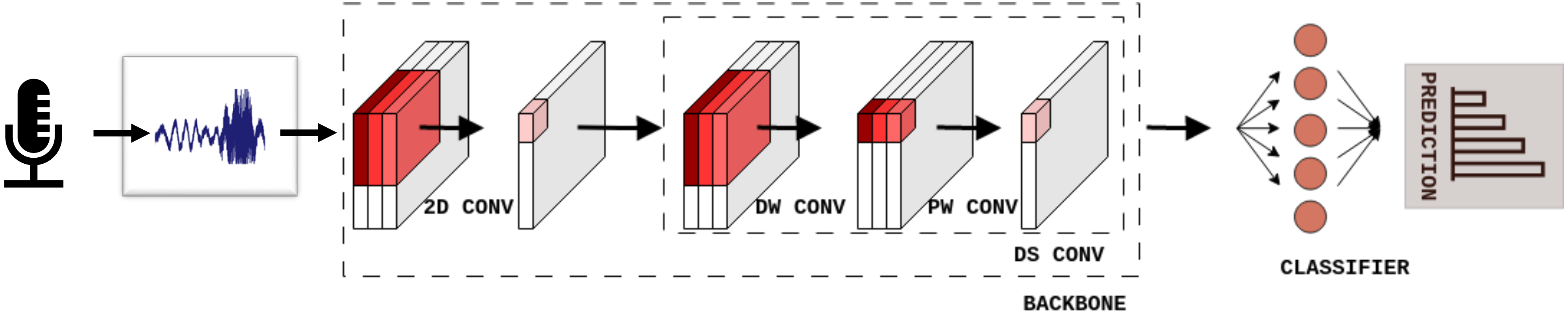
How to mitigate the performance degradation?

- **Server-side training on on-site data**
  - **Does not respect** privacy
  - **Communication reduces** device lifetime
  - **User-specific labeled data is scarce**

- **On-device training (by backpropagation)**
  - **Requires memory** beyond tinyML constraints
  - **Latency increases** with #layers, #samples, #epochs

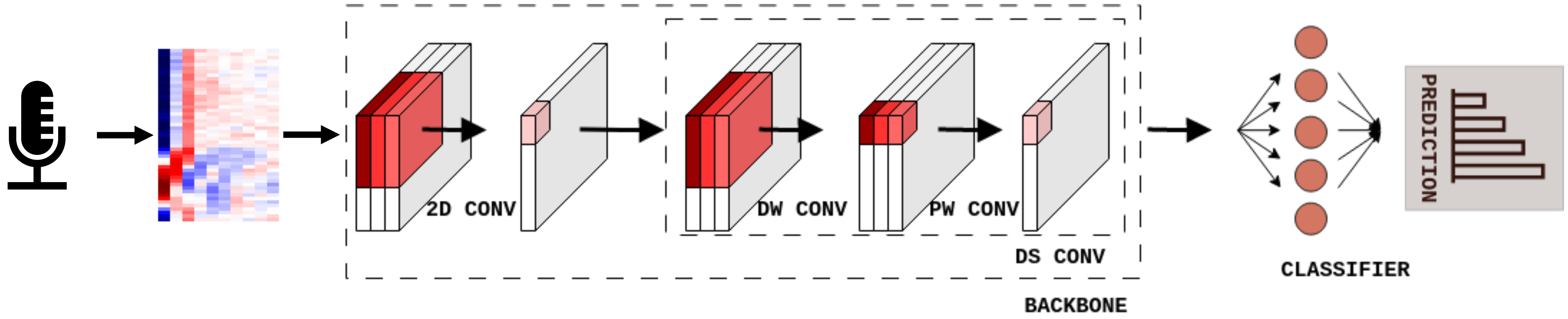
On-Device Learning of Speaker-Aware Embeddings

- **Lightweight backbone, suitable for ODL [Cioflan2024]**



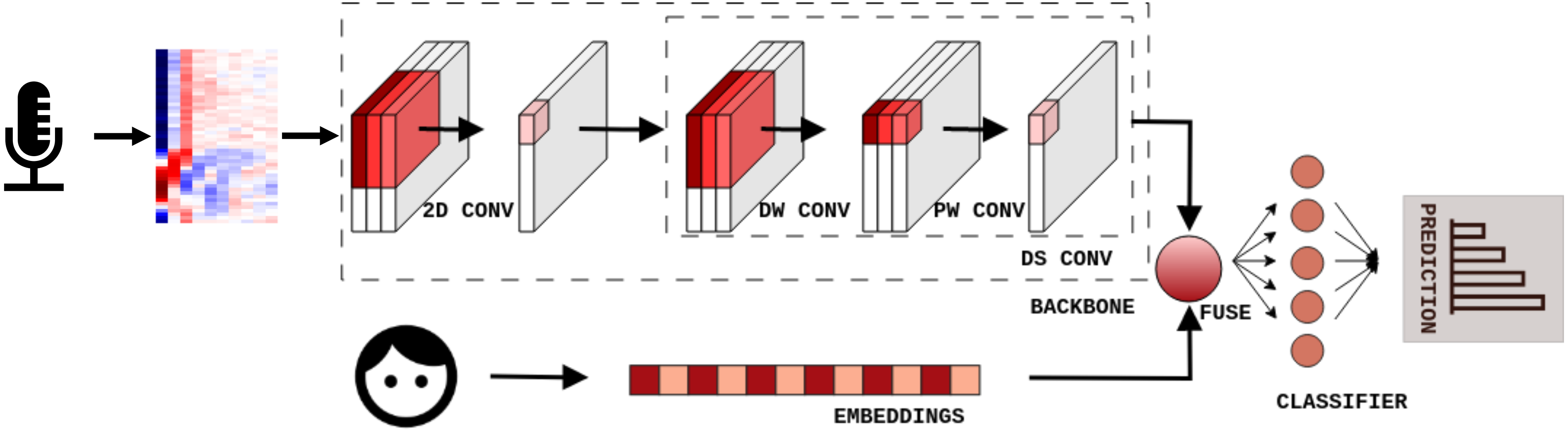
On-Device Learning of Speaker-Aware Embeddings

- **Lightweight backbone, suitable for ODL [Cioflan2024]**



On-Device Learning of Speaker-Aware Embeddings

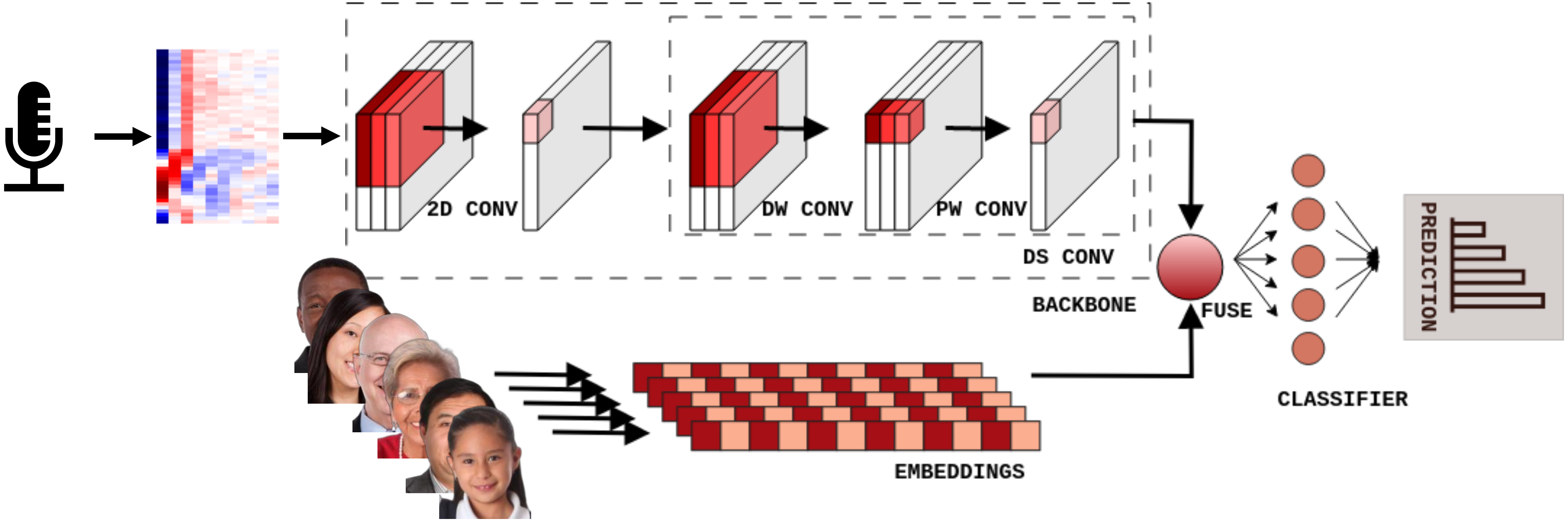
- **We introduced embeddings and fuse them with the KWS backbone**



- **Speaker identity ↦ *n*-dimensional vector**

On-Device Learning of Speaker-Aware Embeddings

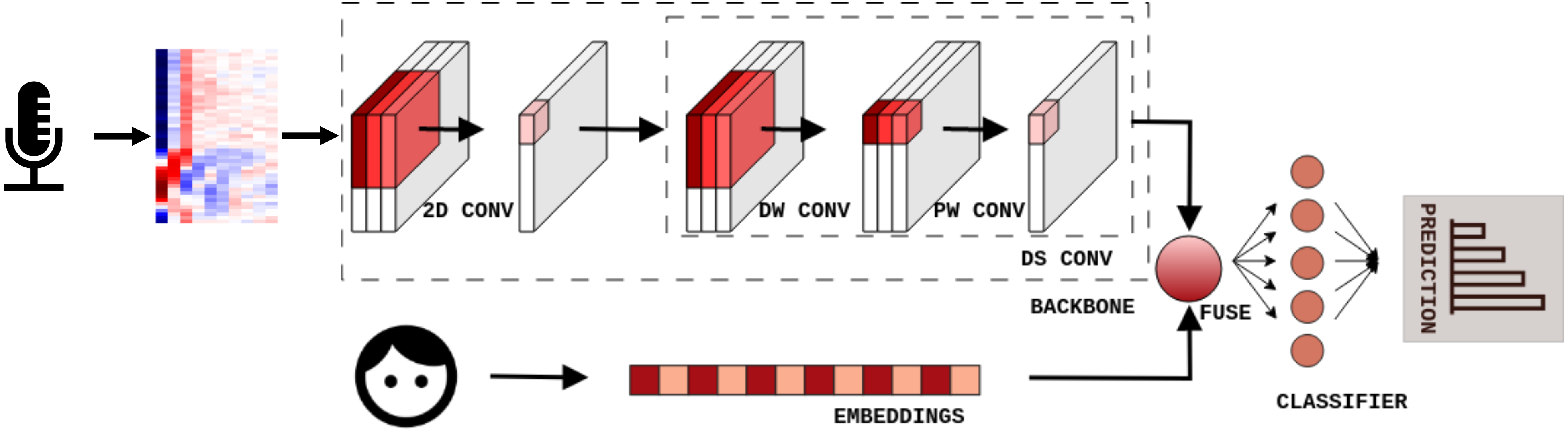
- **We introduced embeddings and fuse them with the KWS backbone**



- **Speaker identity ↦ *n*-dimensional vector** (jointly learned)

On-Device Learning of Speaker-Aware Embeddings

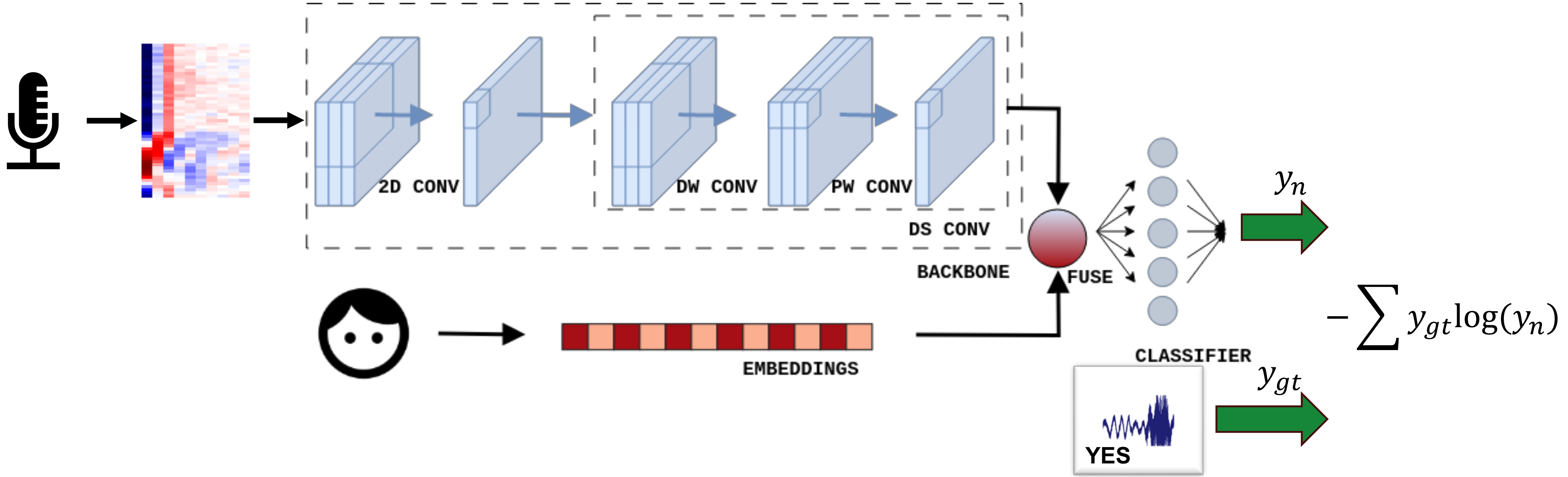
- **We introduced embeddings and fuse them with the KWS backbone**



- **Speaker identity ↦ *n*-dimensional vector** (jointly learned)
- **Feature-level fusion along channel dimension** (projection ∝ speech char.)

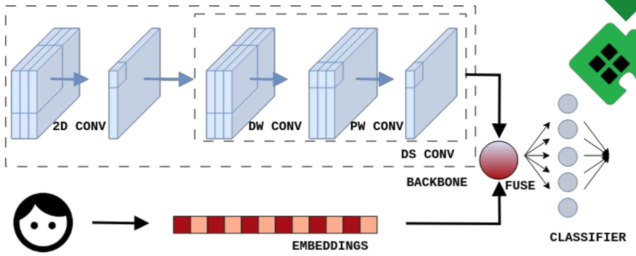
On-Device Learning of Speaker-Aware Embeddings

- **We introduced embeddings and fuse them with the KWS backbone**

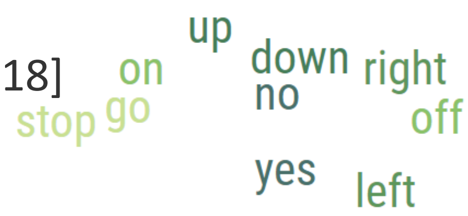


- **Speaker identity ↦ *n*-dimensional vector** (jointly learned)
- **Feature-level fusion along channel dimension** (projection ∝ speech char.)
- **Late fusion minimizing on-device learning costs**

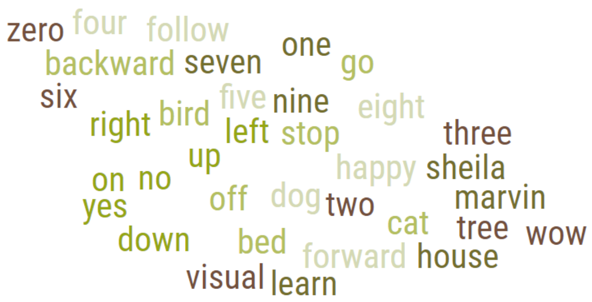
Google Speech Commands dataset



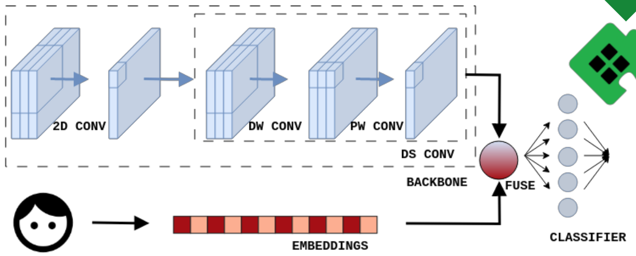
GSC–10
[Warden2018]



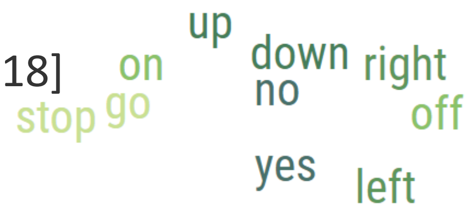
GSC–35



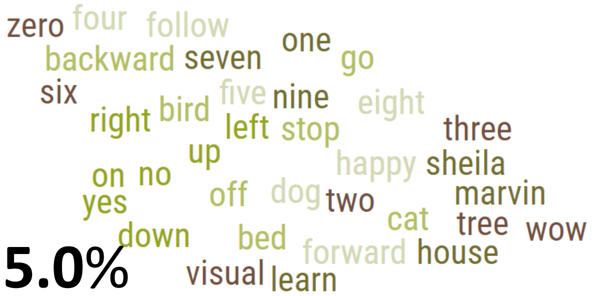
Lessons learned during pretraining



GSC–10
[Warden2018]

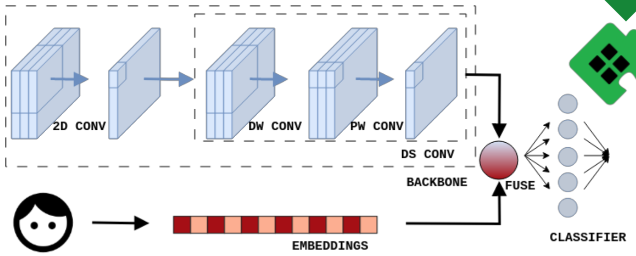


GSC–35



Error rate	**6.9%**	**4.3%**	**5.0%**
Train	1	4	7
Validation	1	1	1
Test	1	1	1
Pretraining	2256	2612	2617
Evaluation	362	6	1

Lessons learned during pretraining



GSC–10
[Warden2018]

up on down right no stop go off yes left

GSC–35

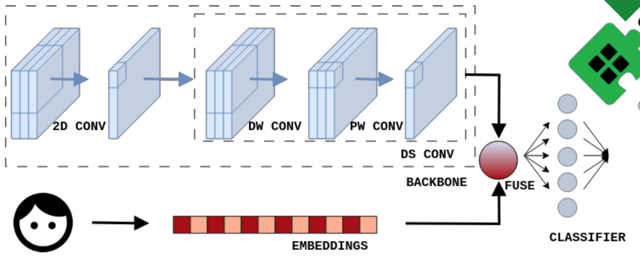
zero four follow one backward seven go six five nine eight right bird left stop three up happy sheila on no off dog two marvin yes cat tree wow down bed forward house visual learn

	GSC–10		GSC–35
Error rate	**6.9%**	**4.3%**	**5.0%**
Train	1	4	7
Validation	1	1	1
Test	1	1	1

On-Device Learning for 6 speakers (between 4 and 22 training samples per class)

2617
1

Lessons learned during pretraining



GSC–10
[Warden2018]

up on down right no stop go off yes left

GSC–35

zero four follow one backward seven go six five nine right bird eight left stop three up happy sheila on no marvin yes off dog two cat tree wow down bed forward house visual learn

Error rate **6.9%** **4.3%** **5.0%**

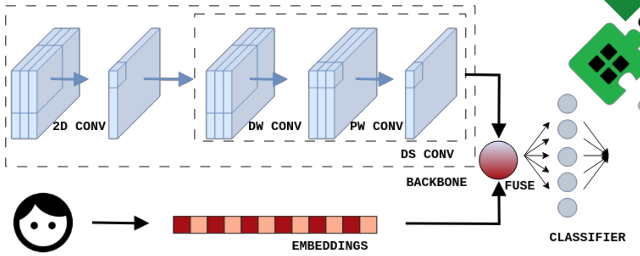
	GSC–10		GSC–35
Train	1	4	7
Validation	1	1	1
Test	1	1	1

On-Device Learning for 6 speakers (between 4 and 22 training samples per class)

2617
1

Embedding	**Error rate [%]**
-	5.39
Addition	5.38
Multiplication	5.28
Backbone-compatible concatenation	5.32
Classifier-compatible concatenation	5.75

Lessons learned during pretraining



GSC–10
[Warden2018]

up on down right no stop go off yes left

GSC–35

zero four follow one backward seven go six five nine eight right bird left stop three up happy sheila on no off dog two marvin yes cat tree wow down bed forward house visual learn

Error rate **6.9%** **4.3%** **5.0%**

Train	1	4	7
Validation	1	1	1
Test	1	1	1

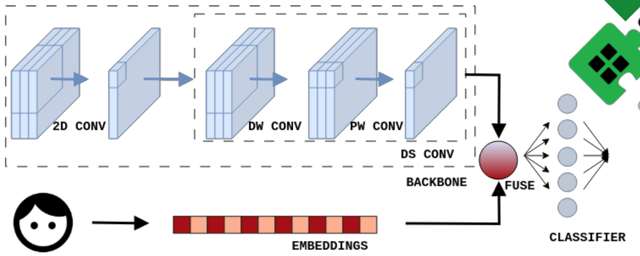
On-Device Learning for 6 speakers (between 4 and 22 training samples per class)

2617
1

Multiplicative fusion of backbone features and user embeddings

	r %]
-	5.39
Addition	5.38
Multiplication	**5.28**
Backbone-compatible concatenation	5.32
Classifier-compatible concatenation	5.75

Few-shot learning of speaker embeddings



- **Learning speech characteristics decreases KWS error rate**
  - Evaluated on six speakers (American, British, and Indian English accents)
  - Sufficient to have four training samples per user

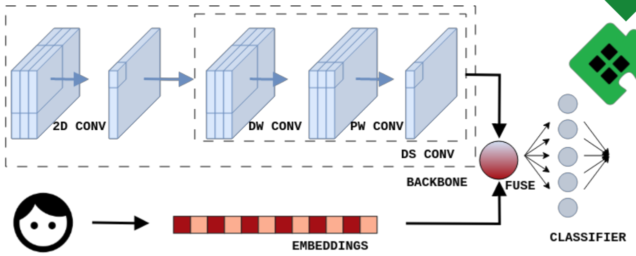
GSC-10	Error rate decrease [%] ↑	
# classes	#samples = 4	
8	0.46	
10	0.73	

Pretraining error rate: 5.33%

GSC-35	Error rate decrease [%] ↑	
# classes	#samples = 4	
20	3.08	
30	4.47	
35	5.74	

Pretraining error rate: 30.08%

Few-shot learning of speaker embeddings



- **Learning speech characteristics decreases KWS error rate**
  - Evaluated on six speakers (American, British, and Indian English accents)
  - Sufficient to have four training samples per user
  - More training samples → more user-specific information → better performance

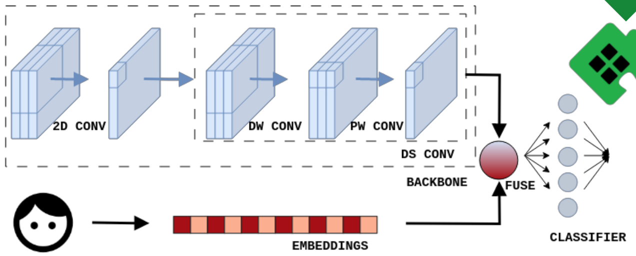
GSC-10	Error rate decrease [%] ↑	
# classes	#samples = 4	#samples ≥ 4
8	0.46	**0.92**
10	0.73	**0.99**

Pretraining error rate: 5.33%

GSC-35	Error rate decrease [%] ↑	
# classes	#samples = 4	#samples ≥ 4
20	3.08	**3.14**
30	4.47	**4.54**
35	**5.74**	5.27

Pretraining error rate: 30.08%

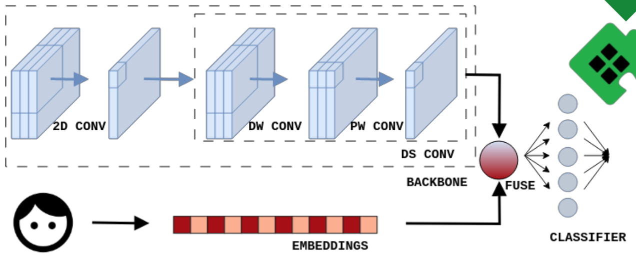
Few-shot learning of speaker embeddings



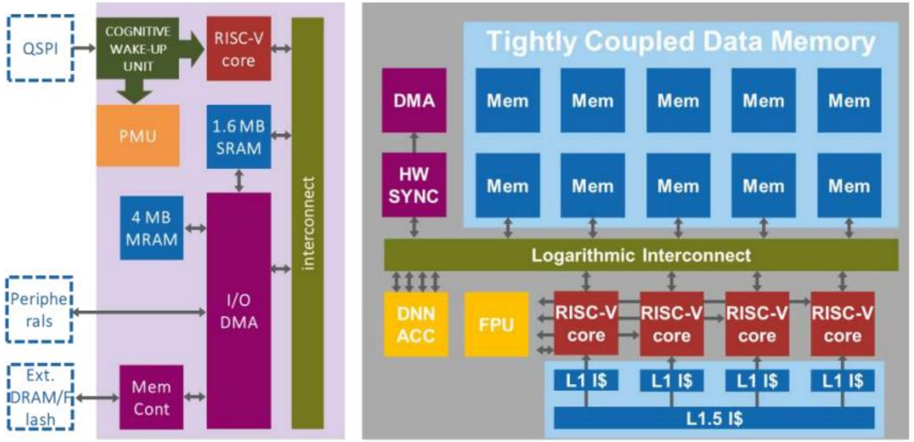
- **Learning speech characteristics decreases KWS error rate**
  - Evaluated on six speakers (American, British, and Indian English accents)
  - Sufficient to have four training samples per user
  - More training samples → more user-specific information → better performance
- **User embeddings mitigate overfitting in domain shifts**

	Error rate decrease [%] ↑	
# classes	Update only the backbone	Update the backbone and **the embeddings**
8/GSC-10	1.72	2.06
20/GSC-35	3.28	3.54
30/GSC-35	4.34	4.67

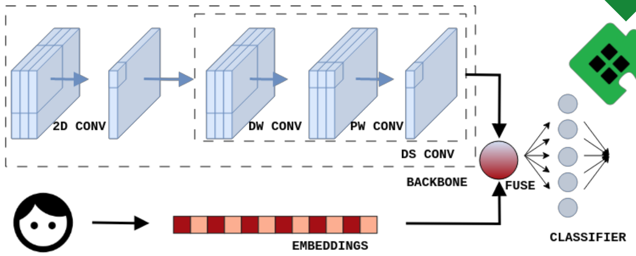
On-Device Learning and tinyML constraints



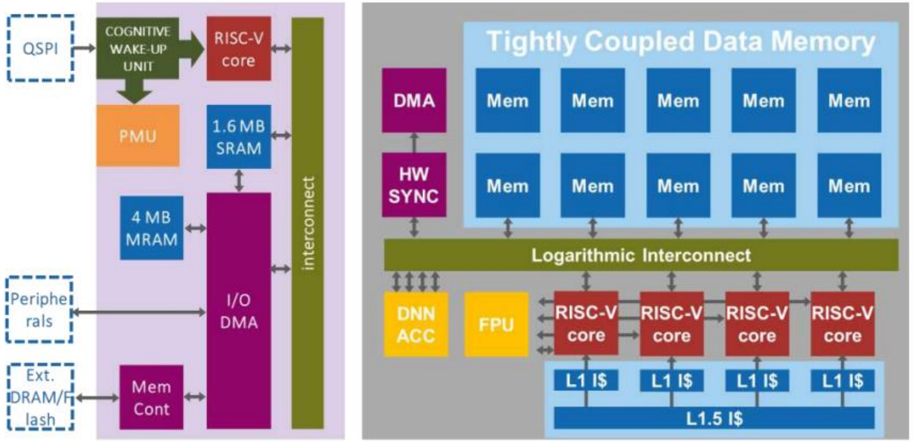
- **Deployment estimates on PULP Vega SoC [Rossi2022]**
  - 79 GFLOP/s/W @ 50 mW
  - 128 kB L1 TCDM, 1.6 MB L2 SRAM



On-Device Learning and tinyML constraints



- **Deployment estimates on PULP Vega SoC [Rossi2022]**
  - 79 GFLOP/s/W @ 50 mW
  - 128 kB L1 TCDM, 1.6 MB L2 SRAM



DS-CNN Model	S(mall)	M(edium)	L(arge)
Parameters [k]	23.7	138.1	416.7
Error decrease [%]	0.73	0.94	0.3
FLOPs [M]	1.04	2.8	4.5
Memory [kB]	3.6	9.7	15.5
Energy [µJ]	13.22	35.53	57.01

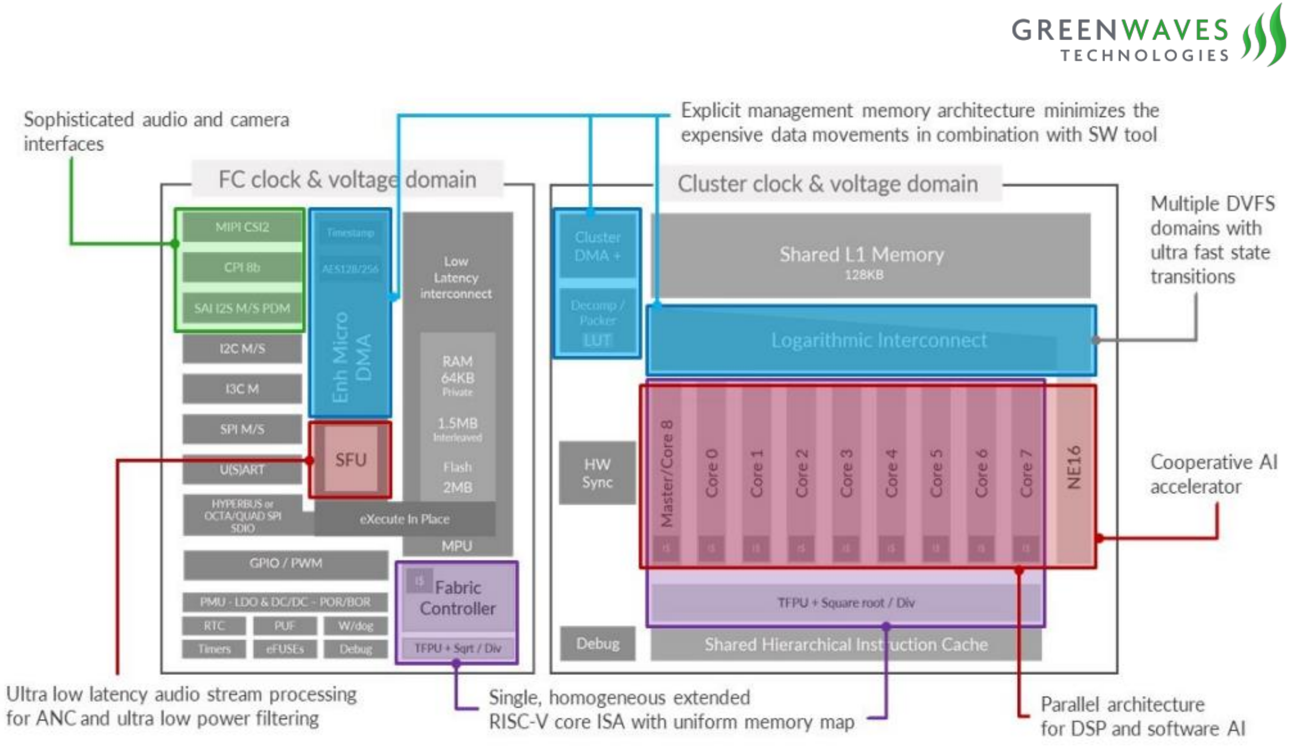
- ODL under 16 kB, feasible for tinyML
- 13 µJ/epoch prolongs device lifetime
- vs. full training: 78% of the error rate, 340× fewer FLOPs
- vs. classifier update: 13× less energy efficient, 55% more accurate

Conclusions

- **Learning speech characteristics increases keyword spotting accuracy**
  - Multiplicative **user embeddings** integrated through late-level fusion
  - Error rate decreases by **up to 5.74%** (relative improvement of **19%**)
    - Sufficient to have **only 4 training samples** per class
- **Speech embeddings are suitable for On-Device Learning on tinyML systems**
  - **16 kB of memory** for backpropagation learning, 128 kB of storage per class for samples
  - **0.73% error decrease with 13 µJ/epoch** (260 µJ per speaker with early stopping)

Conclusions

- **Learning speech characteristics increases keyword spotting accuracy**
- **Speech embeddings are suitable for On-Device Learning on tinyML systems**
- **What are we working on now?**
  - Pairing efficient on-device-learning with state-of-the-art (linear) attention-based backbones [Scherer2024]
  - On-device implementation on GAP9 – derivative of Vega
  - From user embeddings to environment embeddings – boosting performance by focusing on the domain shift
  - **Domain-Aware Keyword Spotting at the Extreme Edge**



References

[Tatman2017] R. Tatman, C. Kasten, "Effects of Talker Dialect, Gender & Race on Accuracy of Bing Speech and YouTube Automatic Captions," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, 2017.

[Savoldi2022] B. Savoldi et al, "Under the Morphosyntactic Lens: A Multifaceted Evaluation of Gender Bias in Speech Translation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

[Cioflan2024] C. Cioflan, L. Cavigelli, M. Rusci, M. De Prado and L. Benini, "On-Device Domain Learning for Keyword Spotting on Low-Power Extreme Edge Embedded Systems," *2024 IEEE 6th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2024.

[Warden2018] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” *arXiv preprint arXiv:1804.03209*, 2018.

[Rossi2022] D. Rossi et al., "Vega: A Ten-Core SoC for IoT Endnodes With DNN Acceleration and Cognitive Wake-Up From MRAM-Based State-Retentive Sleep Mode," in *IEEE Journal of Solid-State Circuits*, 2022.

[Scherer2024] M. Scherer, C. Cioflan, M. Magno, L. Benini, “Work In Progress: Linear Transformers for TinyML”, in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2024.