Work In Progress: Linear Transformers for TinyML

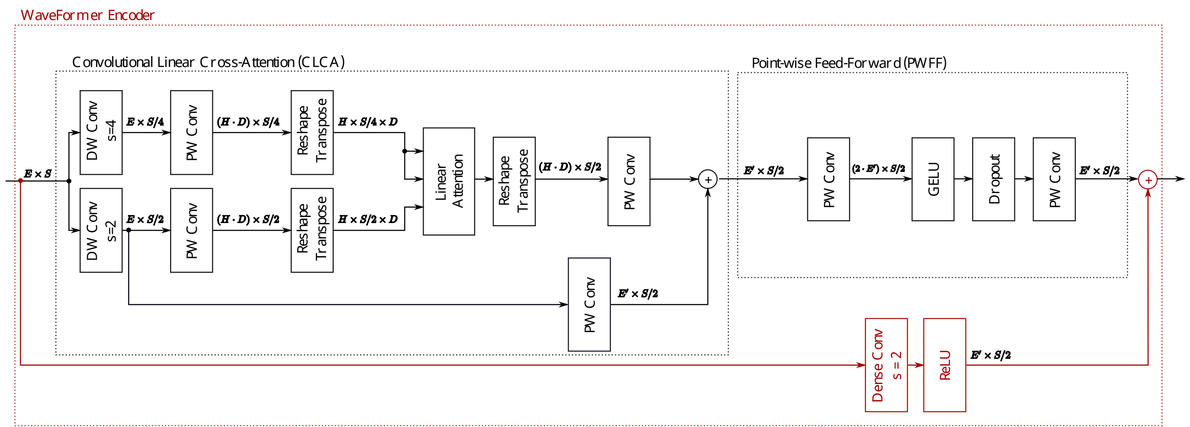
Moritz Scherer†, **Cristian Cioflan**†, Michele Magno‡, Luca Benini†§
†IIS, ETH Zürich, Switzerland, ‡PBL, ETH Zürich, Switzerland, §DEI, University of Bologna, Italy

1 Introduction

The main bottleneck preventing transformer's adoption in embedding time-series processing is the **attention layer**. When implementing the conventional attention mechanism, the **memory and computational requirements scale quadratically with the input length**, severely limiting the ability to process long data sequences. **TinyML platforms have limitations:** computational power, storage and memory, typically in the order of hundreds of kilobytes, and available energy in the context of always-on, battery-operated devices.

Can one use transformers for time-series at the extreme edge while ensuring low inference latency for real-time computation?

2 WaveFormer architecture



We introduce the **WaveFormer** architecture, comprised of three modules. The **Convolutional Linear Cross-Attention (CLCA)** implements Multi-Head Cross-attention [1] by using a convolutional projection for Q and a shared convolutional projection to compute the K and V matrices . Each projection consists of a depthwise (DW) convolution, with configurable stride to reduce the sequence length, followed by a pointwise (PW) convolution. A residual connection with a PW convolution, added between the downsampling DW convolution of the Q tensor and the output of the linear attention operator, increases training stability. The attention block is followed by a **Pointwise Feed-Forward (PWFF)**, while a **residual, strided dense convolution** is added to improve the intermediate representation.

3 Quantization

For all operators in the network except linear attention and GELU activations, we use the TQT algorithm [2]. For GELU activations, we use the algorithm proposed in I-BERT [3]. To quantize linear attention, we target 8-bit integer matrix multiplications and 32-bit division operations. To avoid division by zero, we add a numerically small constant μ, at least as large as the quantum of the denominator. To address asymmetric error sensitivity in the division operator during quantization, we apply percentile-based clipping to determine the initial clipping bound values of all quantized tensors.

3 Results on Google Speech Commands dataset

We evaluate WaveFormer on the Google Speech Commands v2[4] keyword spotting dataset, considering the **12-class and 35-class** problem. We resample the input of training, validation, and test inputs from 16.384 kHz to 8.192 kHz, shortening the sequence length to **8192 samples for a 1-second input**. We apply random data augmentation (i.e., polarity inversion, Gaussian noise, gain, and reverb) on the training dataset and we quantize the inputs.

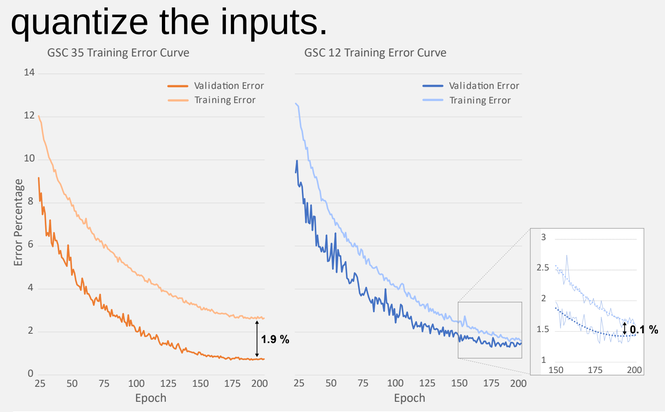


Fig. 1. **Plot of error rate curves for training and validation of the 35-class and 12-class model.** Notably, the training-validation margin is much larger for the 35-class model, indicating better generalization.

Model	Type	Size [Params]	Ops per inf.[M]	GSC12 Acc.[%]	GSC35 Acc.[%]
KWT-3 [5]	Transf.	5360k	526	98.6	97.7
LeTR-256 [6]	Transf.	1110k	81	-	98.2
Att-RNN [7]	Att. & LSTM	202k	-	96.9	93.9
BC-ResNet-8 [8]	CNN	321k	89	98.7	-
WaveFormer	**Lin. transf.**	**130k**	**19**	**98.8**	**99.1**

We **outperform the state-of-the-art in terms of accuracy by 0.1% on GSC12** and **0.9% on GSC35.** We reduce the number of parameters by **2.5×** and the number of operations by **4.7×**

We deploy WaveFormer on Ambiq Apollo 4 (ARM Cortex-M4F) running at 192 MHz and measure an **inference latency of 741 ms**, enabling real-time attention-based keyword spotting at the extreme edge.

4 Conclusion

WaveFormer employs Linear Attention blocks and optimized quantization strategies enabling transformers for TinyML systems.

We achieve state-of-the-art accuracies on GSC12 and GSC35 keyword spotting datasets, with 2.5× and 4.7× parameters and operations reductions, respectively. We show real-time inference on ARM Cortex-M4F microcontroller.

References

[1] Lin, H., Cheng, X., Wu, X., and Shen, D., "CAT: Cross Attention in Vision Transformer" in *IEEE International Conference on Multimedia and Expo (ICME)*, 2022.
[2] Jain, S., et al. "Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks" in *Proceedings of Machine Learning and Systems* 2, 2020.
[3] Kim, S., Gholami, A., Yao, Z., Mahoney, M.W. and Keutzer, K., I-BERT: Integer-only BERT Quantization" in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.
[4] Warden, P., "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition", 2018.
[5] Berg, A., O'Connor M., and Tairum Cruz, M., "Keyword Transformer: A Self-Attention Model for Keyword Spotting" in *Interspeech,* 2021.
[6] Ding, K., Zong, M., Li, J., and Li, B., "LETR: A Lightweight and Efficient Transformer for Keyword Spotting" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
[7] de Andrade, DC., Leo, S., Viana, MLDS., and Bernkopf, C., "A neural attention model for speech command recognition", 2018.
[8] Kim, B., Chang, S., Lee, J., Sung, D., "Broadcasted Residual Learning for Efficient Keyword Spotting", in *Interspeech*, 2021.