ETH zürich

Distributed Inference with Minimal Off-Chip Traffic for Transformers on Low-Power MCUs

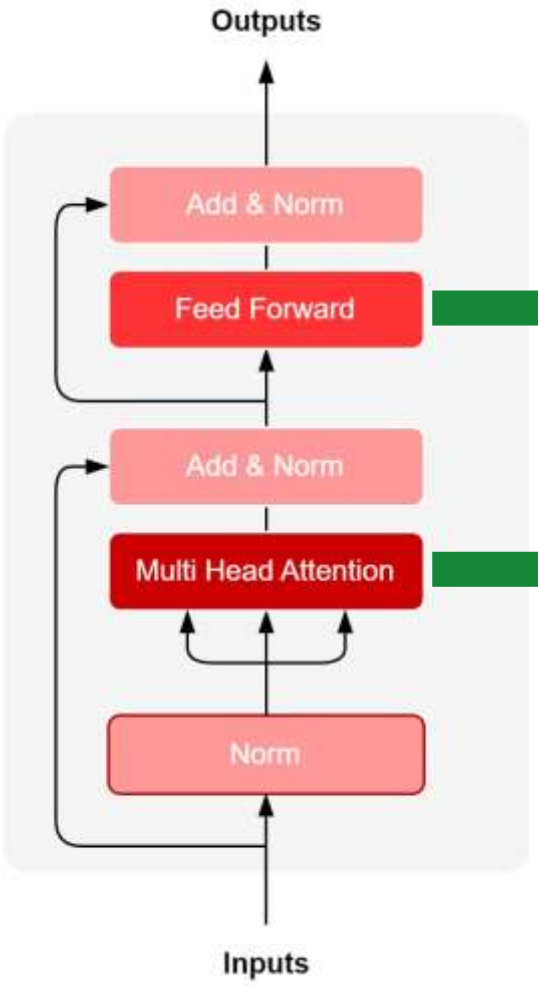
Severin Bochem*, Victor J.B. Jung†, Arpan Suravi Prasad†, Francesco Conti‡, Luca Benini†‡

**D-ITET, ETH Zurich; †Integrated Systems Laboratory, ETH Zurich; ‡DEI, University of Bologna*

PULP Platform

Open Source Hardware, the way it should be!

Transformer Model Architecture



- **Large weight matrices do not fit on chip**
- **Costly off-chip access**

$$\text{Output} = \text{GeLU}(I * W_{L1} * W_{L2}$$

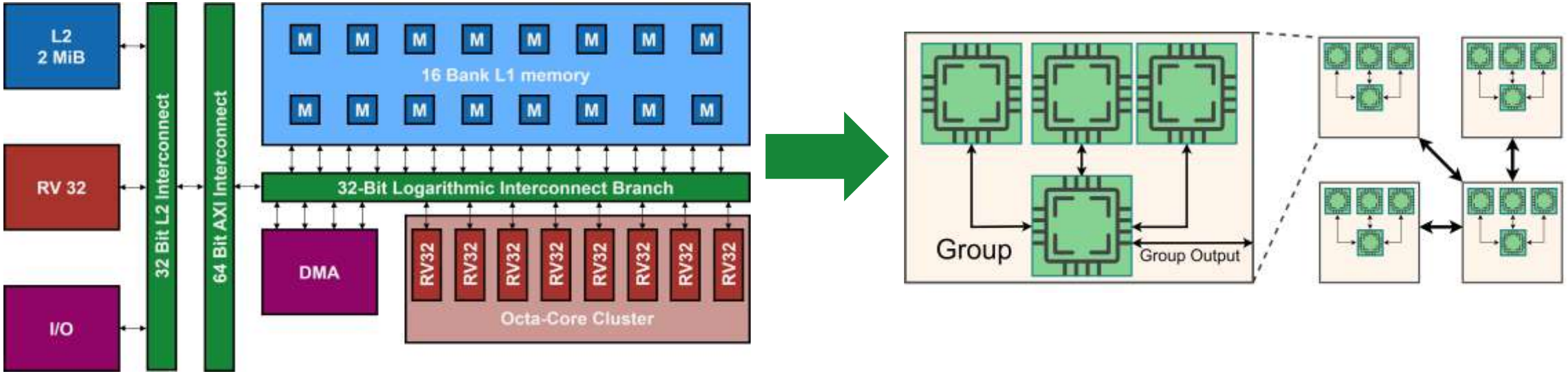
$$\mathbf{Q} = \mathbf{X}\mathbf{W}_{\text{query}}, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_{\text{key}}, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_{\text{value}}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}}\right)\mathbf{V}$$

$$\text{softmax}(\boldsymbol{x})_i = \frac{e^{x_i - \max(\boldsymbol{x})}}{\sum_{j=1}^{n} e^{x_j - \max(\boldsymbol{x})}}$$

Idea: Distribute Workload across mutliple chips

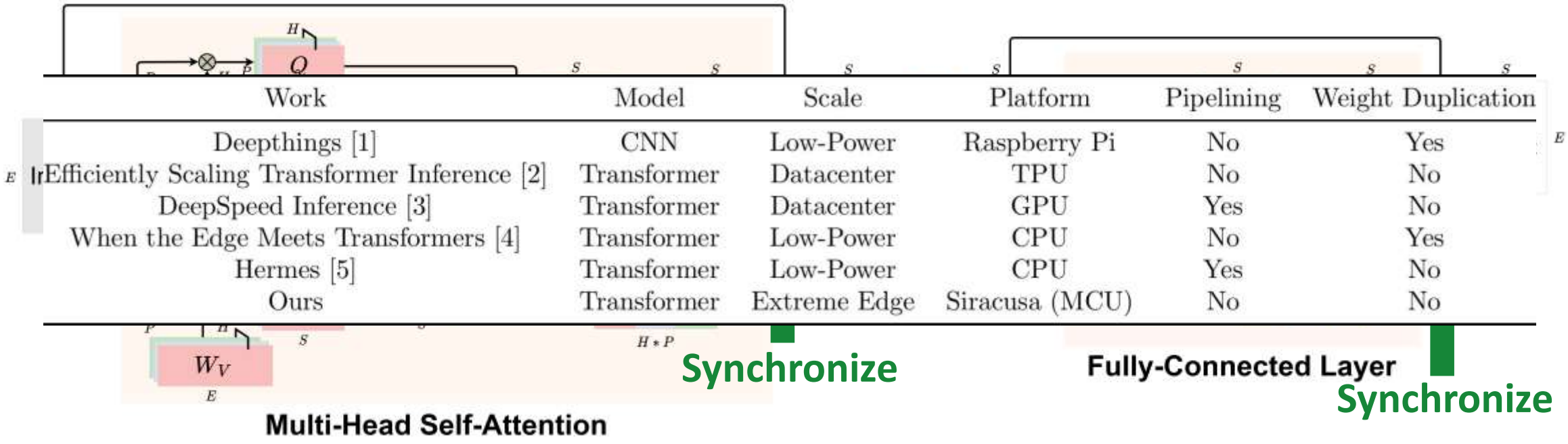
- One chip is too small to hold full model weights
- Distribute model across multi-chip system to run from on-chip memory



Partitioning Scheme

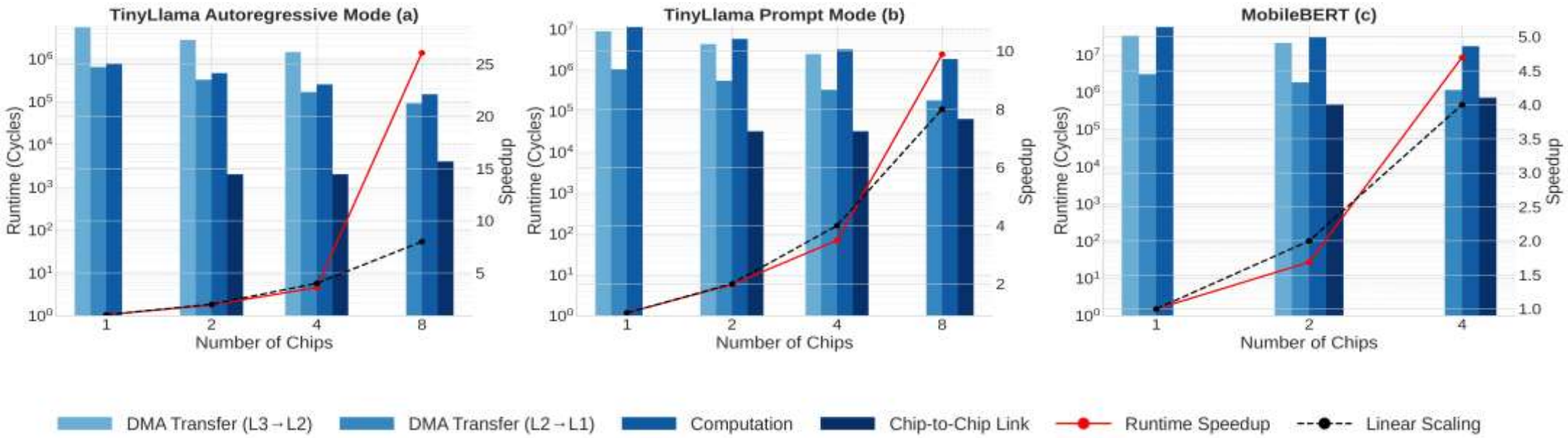
Previous work **duplicated weights** or uses **model pipelining**, both of which is infeasible for autoregressive Transformer inference!

Work	Model	Scale	Platform	Pipelining	Weight Duplication
Deepthings [1]	CNN	Low-Power	Raspberry Pi	No	Yes
Efficiently Scaling Transformer Inference [2]	Transformer	Datacenter	TPU	No	No
DeepSpeed Inference [3]	Transformer	Datacenter	GPU	Yes	No
When the Edge Meets Transformers [4]	Transformer	Low-Power	CPU	No	Yes
Hermes [5]	Transformer	Low-Power	CPU	Yes	No
Ours	Transformer	Extreme Edge	Siracusa (MCU)	No	No



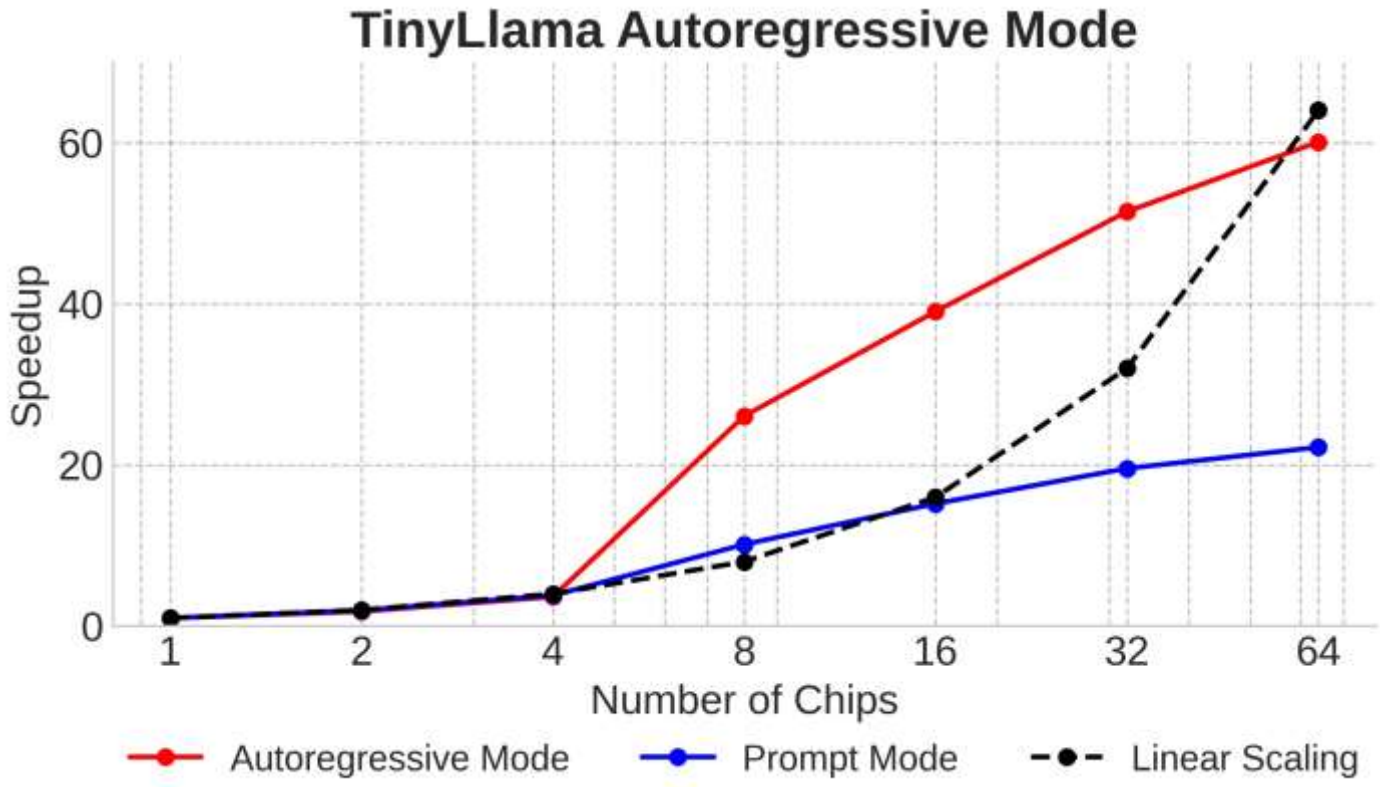
Speedup of Distributed Inference

Using more than two chips achieves **above linear speedup** by keeping **layer weights in on-chip** memory



Scalability Study for TinyLlama

Partitioning becomes **less efficient** as tensor sizes per chip scale down



Conclusion

- Proposed and benchmarked a **Transformer partitioning scheme**
- **No weight duplication** and small chip-to-chip traffic
- Achieves **super-linear speedup** for various models

Want to hear more?
See more results?
Stop by at the poster session!

Paper on Arxiv