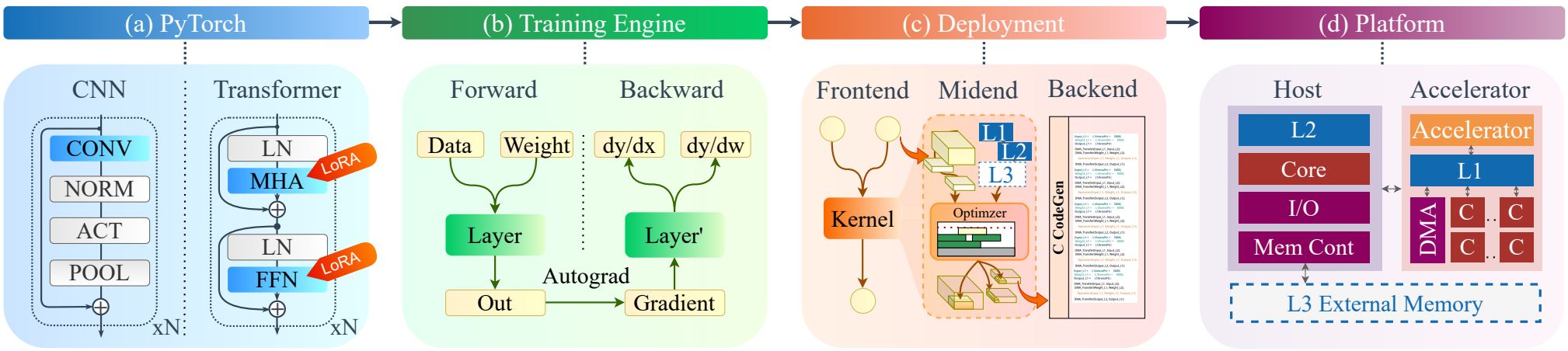
TrainDeeploy: Hardware-Accelerated Parameter-Efficient Fine-Tuning of Small Transformer Models at the Extreme Edge

Run Wang[1], Victor J.B. Jung[1], Philip Wiese[1], Francesco Conti[2], Alessio Burrello[3], Luca Benini[1,2]
[1]Integrated Systems Laboratory, ETH Zurich; [2]DEI, University of Bologna; [3]DAUIN, Politecnico di Torino



1 Motivation

How to enable **on-device Transformer training** on ultra-low-power edge SoCs? We present **TrainDeeploy**, a unified framework for **CNN + Transformer fine-tuning with LoRA** on heterogeneous RISC-V SoCs, achieving up to **13.4 FLOPs/cycle**. Using LoRA, we reduce trainable parameters by **15×** and off-chip transfers by **1.6×**. RedMulE acceleration achieves **2.3–3.5× speedup** over the 8-core RISC-V cluster.

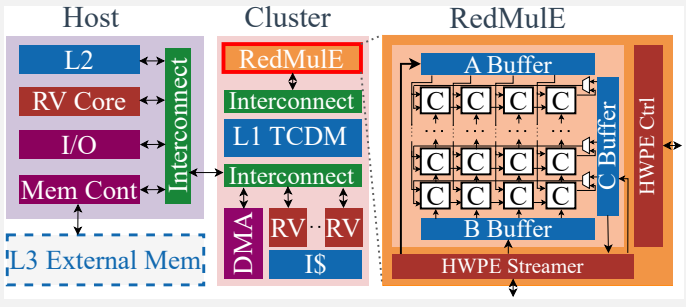
2 TrainDeeploy Framework

TrainDeeploy is the **first end-to-end CNN + Transformer training pipeline** for heterogeneous ultra-low-power SoCs with **LoRA[3] support**.

- Automatic differentiation builds full forward+backward training graph.
- Memory allocation: tiling + static allocation via 2D bin-packing.
- Offloads GEMM kernels to RedMulE accelerator.

Hardware Target

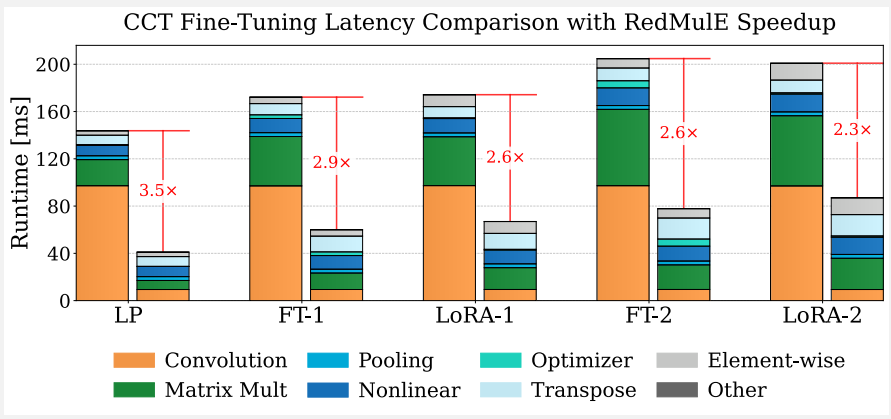
- PULP-based SoC: **8 RISC-V cores** + **RedMulE[4] FP GEMM engine** at 360 MHz.
- Memory hierarchy: **128 KB L1** TCDM + **2 MB L2** SRAM + **32 MB L3** HyperRAM.



3 Fine-Tuning Strategies

- Target: **CCT-2/3×2** (0.28M params, 67 MFLOPs inference).
- Five strategies: **LP** (linear probing), **FT-1/FT-2** (full fine-tuning of last 1/2 attention blocks), **LoRA-1/LoRA-2** (rank r=4 on last 1/2 blocks).
- Conv tokenizer frozen; adapt **attention blocks** + classifier selectively. LoRA-2 achieves **96.0% on CIFAR->MNIST** with only **0.05 MB** trainable params (15× fewer than FT-2).

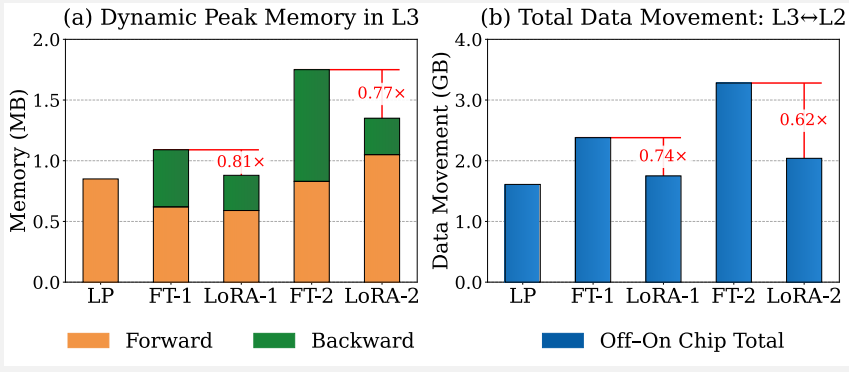
4 End-to-End Latency



- RedMulE acceleration yields **2.3–3.5× speedup**, runtimes drop to **41–87 ms**.
- Peak throughput: up to **11 gradient updates/sec** for end-to-end CCT fine-tuning.

5 Memory & Data Movement

- LoRA reduces dynamic peak memory by **19–23%** compared to full fine-tuning.
- Off-chip data transfer reduced by **1.6×** with LoRA (0.62× of full FT).
- Trainable parameters cut by **15×** (0.05 MB vs 0.76 MB).



References

1. M. Scherer et al., “Deeploy: Enabling Energy-Efficient Deployment of Small Language Models on Heterogeneous Microcontrollers,” IEEE TCAD, 2024.
2. D. Nadalini et al., “PULP-TrainLib: Enabling On-Device Training for RISC-V Multi-core MCUs,” SAMOS 2022.
3. E. J. Hu et al., “LoRA: Low-Rank Adaptation of Large Language Models,” arXiv:2106.09685, 2021.
4. Y. Tortorella et al., “RedMule: A Mixed-Precision Matrix-Matrix Operation Engine for Flexible and Energy-Efficient On-Chip Linear Algebra and TinyML Training Acceleration”, FGCS, 2023

github.com/pulp-platform/Deeploy