

INTRODUCTION

Helion meets RISC-V

Helion writes tiled kernels in PyTorch syntax and autotunes them, but its backends target GPUs and accelerators. RISC-V CPUs with the Vector extension (RVV) have no path.

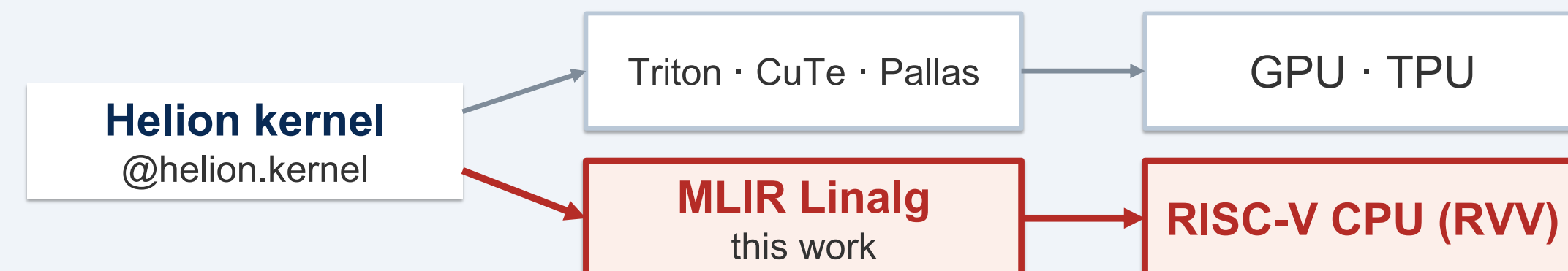


Fig. 1. Helion's existing back ends (grey) and the path added here (red).

What is MLIR?

MLIR (Multi-Level IR) is the LLVM project's compiler infrastructure. Programs are written in dialects at several abstraction levels and lowered step by step; every level can be inspected and transformed. IREE, Triton and torch-mlir build on it.

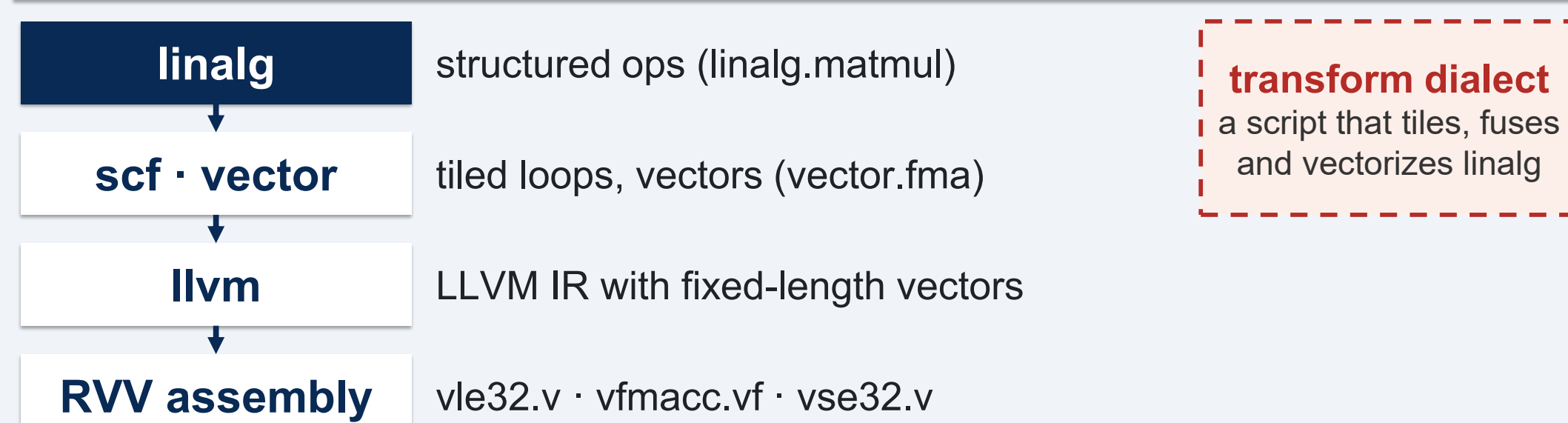


Fig. 2. Progressive lowering; tiling and vectorization act on Linalg.

Why emit Linalg?

- Parallel and reduction dims stay explicit, so tiling, packing and vectorization are transformations, not loop pattern matching.
- One IR feeds any MLIR compiler: IREE or upstream.

CONTRIBUTIONS

1	Linalg emitter	DeviceIR (torch.fx) → linalg on tensors
2	Two back ends	IREE, or upstream MLIR → LLVM
3	CPU search space	IREE configs; cache + register tiles
4	RVV-aware codegen	LMUL bound from the register tile
5	On-device tuning	candidates timed on the RISC-V board
6	Multi-core	scf.forall → OpenMP on 8 cores

METHODS

From a Helion kernel to RVV code

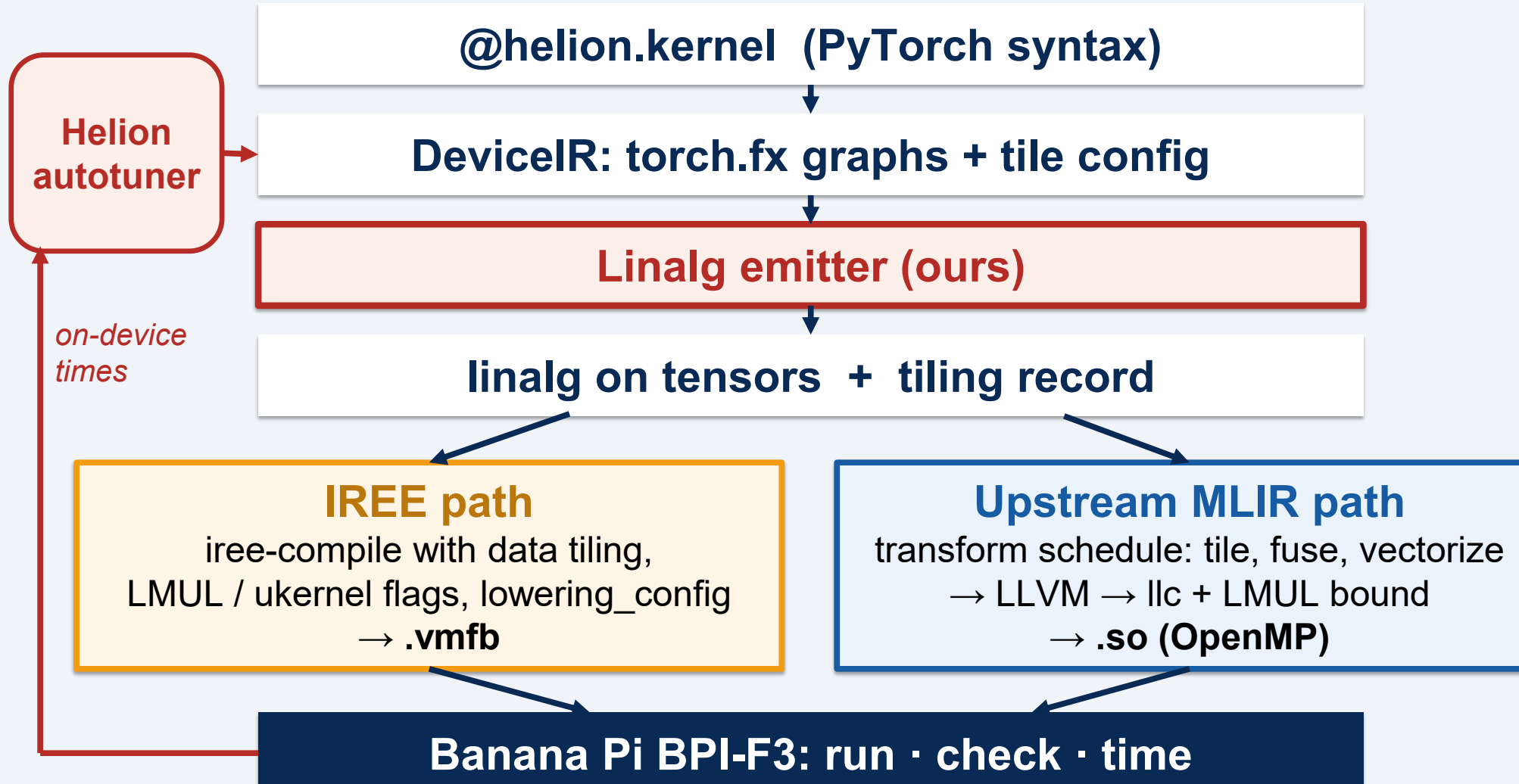


Fig. 3. One emitted Linalg, two compilers; on-device times drive the search.

Shared Linalg, two transform scripts

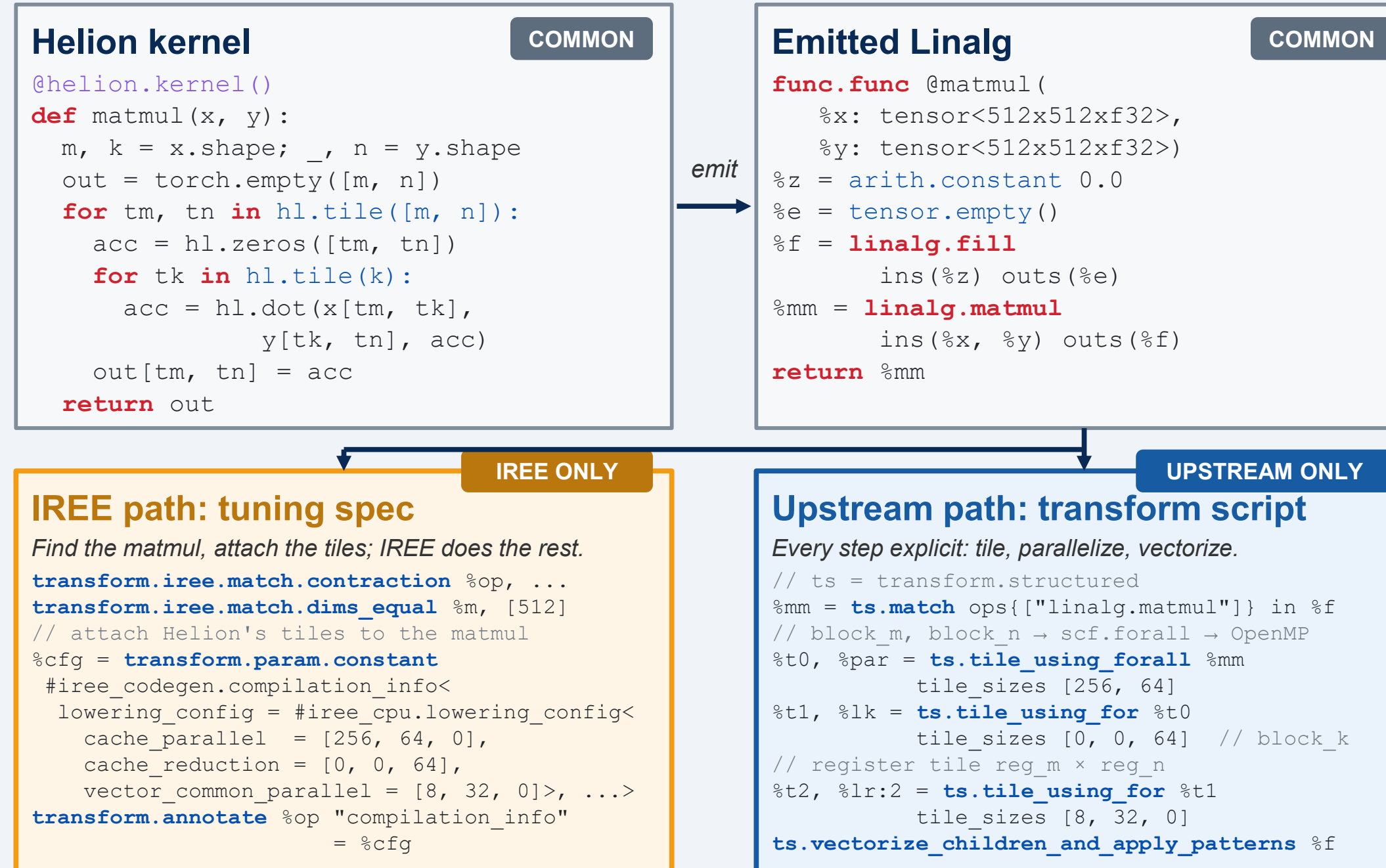


Fig. 4. Grey: common to both paths; orange: IREE only; blue: upstream only.

CPU Search Space

Path	Knob	From
IREE	data tiling · LMUL max · ukernels	compile flags
IREE	cache / vector tile sizes	lowering_config
Upstream	block_m · block_n · block_k	Helion block_sizes
Upstream	reg_m · reg_n	register tile
Upstream	LMUL bound	derived

EVALUATION

Experiment Setup

Board	Banana Pi BPI-F3 · SpacemiT X60 × 8 cores
ISA	RV64GCV · RVV 1.0 · VLEN 256
Data	FP32 · correctness-checked
ISA	oneDNN · OpenBLAS · MLAS · ggml

Decode in vLLM

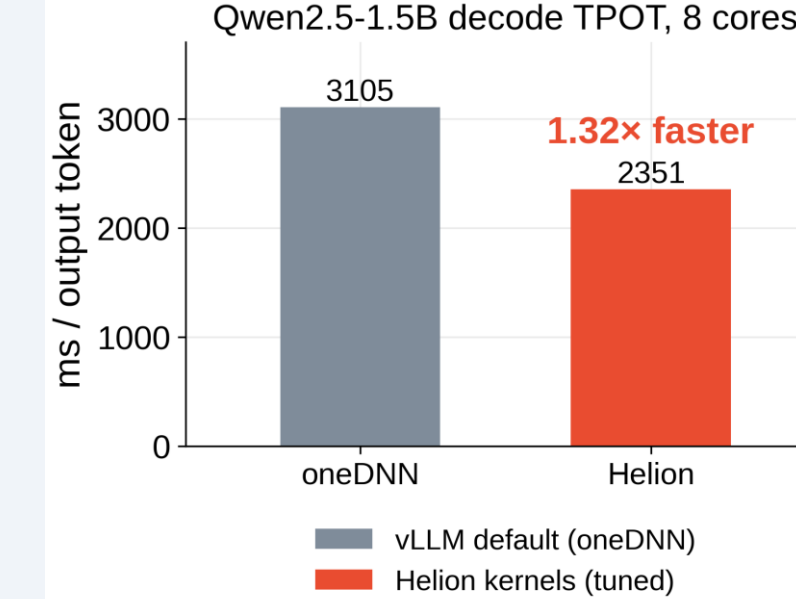


Fig. 5. Decode TPOT in vLLM.

Prefill and decode vs hand-written libraries

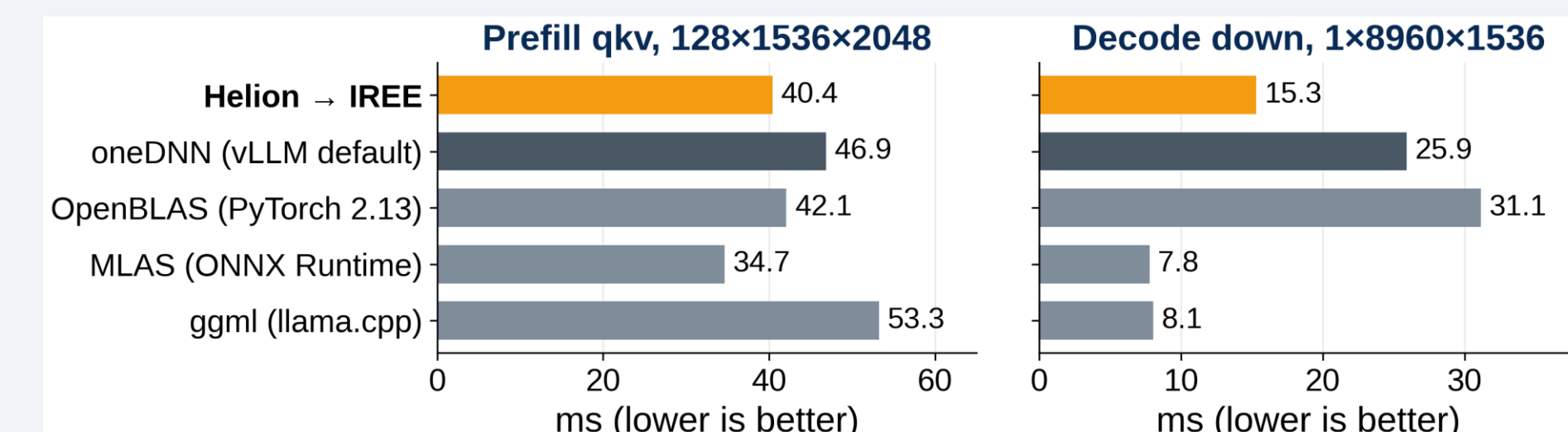


Fig. 6. Two Qwen2.5-1.5B linears; 8 cores, FP32, median of 3 rounds.

Qwen2.5-1.5B linears vs oneDNN, 8 cores

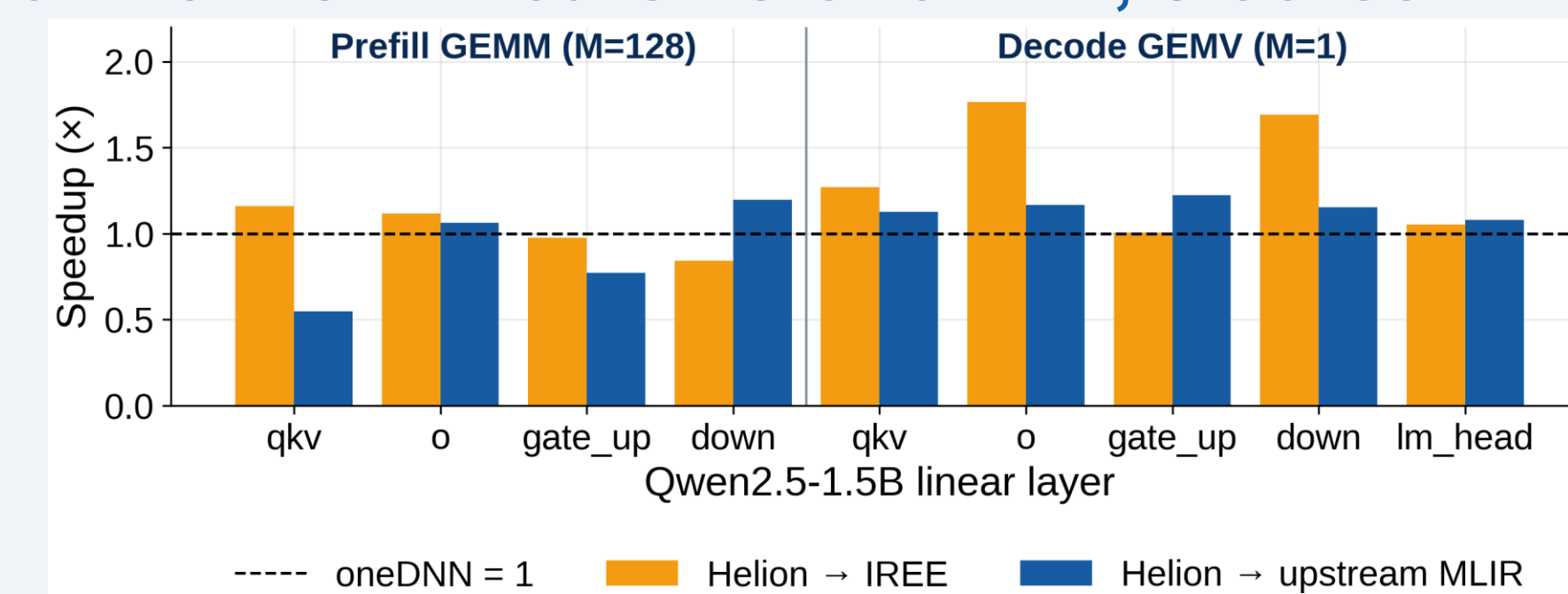


Fig. 7. Median of 3 rounds; oneDNN is vLLM's default; IREE data tiling per shape.

FUTURE WORK

- Faster decode GEMV:** pack weights once; MLAS and ggml are still ~2x ahead.
- More kernels, faster tuning:** attention and norms; a cost model to cut tuning time.

ACKNOWLEDGEMENT

This work is supported in part by Taiwan NSTC and Mediatek.

Give us feedback

