

Abstract

We restructure ExecuTorch’s quantized Vulkan GEMM dispatch using cooperative matrix tiles, workgroup staging, fusion and device-specific tile selection. An initial Radeon 780M study shows 3.90× faster int4 GEMM and 2.45–2.96× faster Llama 3.2 1B/3B inference versus conventional compute shaders. We compare the fully tuned Vulkan pipeline across eight GPUs with stock ExecuTorch and llama.cpp.

2.81–7.88×

Faster prefill than stock
ExecuTorch
Across 8 GPUs

32–46%

Lower Exynos GPU power
WMMA 4w vs stock ET 1.5
Prefill, all 3 models

24/24

Cases ahead of
llama.cpp Vulkan
Q4_0, 3 models

86–166%

Of llama.cpp vendor speed
CUDA / SYCL / ROCm
Per-GPU, 3-model geomeans

What is a cooperative matrix?

LLM linear layers multiply activation and weight matrices. GPU threads in a subgroup cooperate on small matrix tiles through the vendor-neutral VK_KHR_cooperative_matrix interface.

Matrix multiply-add: $D = A \times B + C$

Portable SPIR-V: OpCooperativeMatrixMulAddKHR

4w unpacks 4-bit weights to FP16 operands for hardware matrix instructions.
8da4w feeds dynamic INT8 activations and unpacked INT8 weights into hardware matrix instructions.

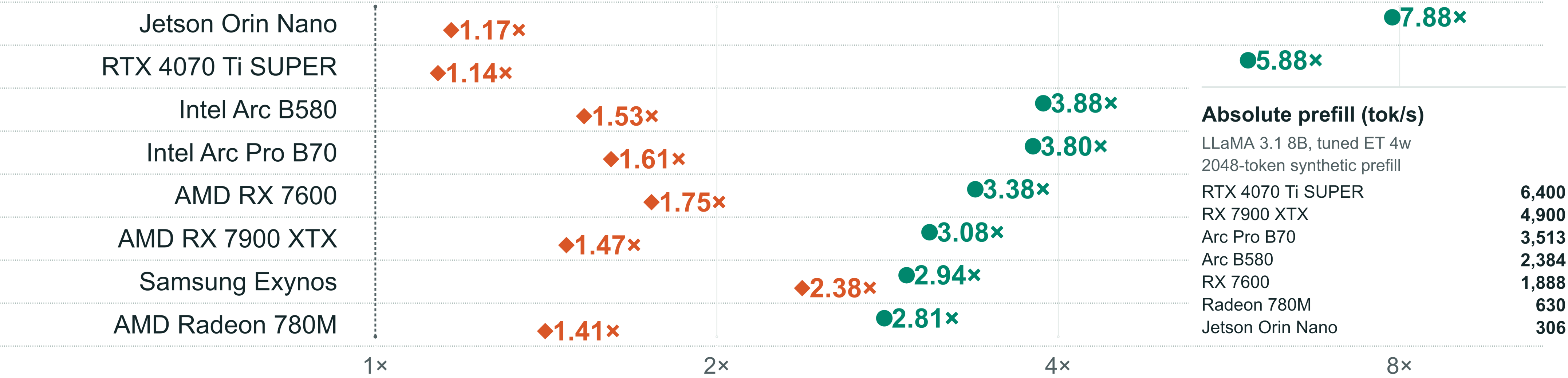
The driver compiles SPIR-V to GPU ISA instructions: WMMA on Radeon 780M and DPAS on Intel Arc. We query supported data types and tile sizes per GPU.

Prefill gains across the GPU fleet

Tuned cooperative-matrix ExecuTorch prefill / each named baseline

• vs stock ExecuTorch 1.5 (stable)

◆ vs llama.cpp Vulkan Q4_0



Speedup ratio on a logarithmic axis. 1× equals the named baseline.

Teal: geomean over 1B / 3B / 8B × 4w / 8da4w (6 cases). Orange: geomean of ET 4w / llama.cpp Q4_0 over the 3 model sizes.

Relative to llama.cpp vendor backends

ET 4w / llama.cpp vendor Q4_0

GPU / backend	1B	3B	8B
Orin / CUDA	0.98×	1.05×	1.07×
RTX / CUDA	0.88×	0.89×	0.86×
B580 / SYCL	0.97×	0.88×	0.75×
B70 / SYCL	0.96×	0.86×	0.78×
RX 7600 / ROCm	1.75×	1.68×	1.55×
RX 7900 XTX / ROCm	0.93×	1.19×	1.19×

1× = same prefill speed. ROCm = llama.cpp HIP. RTX = RTX 4070 Ti SUPER.
AMD rows use matched 12 Oct 2026 sessions. No vendor data for 780M or Exynos.

Test configuration

Models (3): LLaMA 3.2 1B, LLaMA 3.2 3B, LLaMA 3.1 8B.
Stock: unmodified ExecuTorch 1.5 stable release (Vulkan).
llama.cpp: b11430 / 8345f33 (5 Oct 2026).
2048-token aligned synthetic prefill. Median of 5 valid runs (7 for tuned RTX / Intel).
4w: 4-bit weights. 8da4w: dynamic 8-bit activations / 4-bit weights. Full Vulkan pipeline tuning. Best screened llama.cpp settings.

Code and upstream integration

Samsung branch



sarc-acl/executorch
branch: sarc/1.5

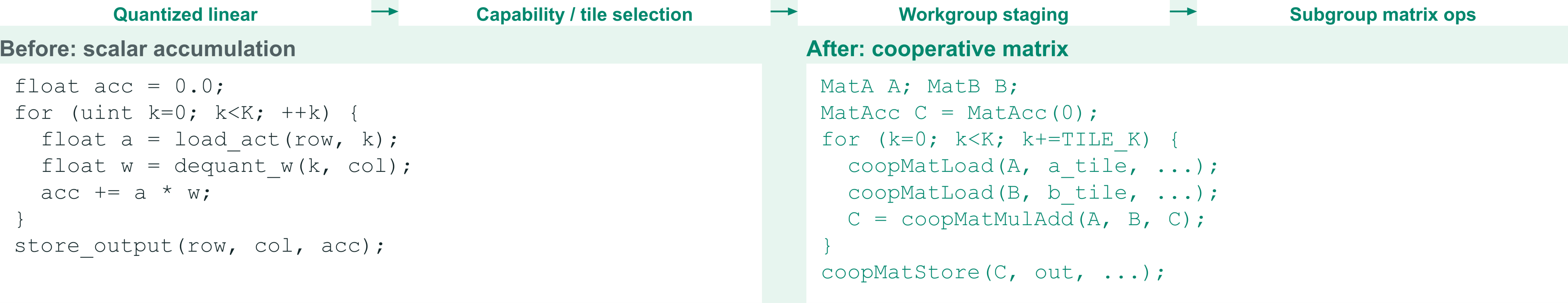
Upstream



pytorch/executorch

Upstream
contribution
in progress

Shader anatomy: before and after



Each invocation accumulates one output.
The shader manages tile bookkeeping.
Schematic GLSL. MatA / MatB / MatAcc are compatible subgroup matrix types. Addressing, staging, synchronization and quantization corrections are omitted.

Subgroups accumulate matrix tiles.
Workgroups stage and schedule tiles.