

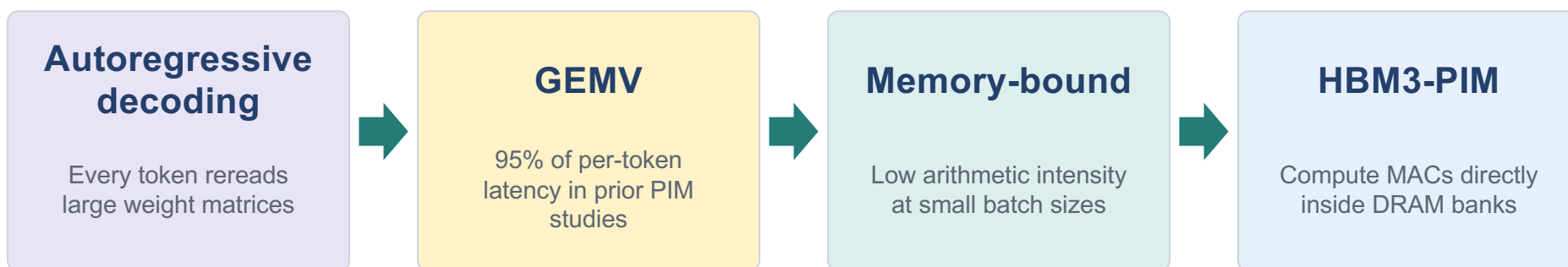
RH+: Row-Hit-Optimized Scheduling for PIM-based LLM Inference

Yongchan Jung · Shafayat Mowla Anik · Byeong Kil Lee · Jeeho Ryoo
Fairleigh Dickinson University · University of Colorado Colorado Springs

ASAP 2026

LLM decoding is memory-bound

Autoregressive inference is dominated by GEMV weight reads, making PIM an attractive fit.



**The opportunity: eliminate weight movement —
but only if the DRAM access schedule preserves locality.**

Stride-64 makes every MAC pay the full nRC

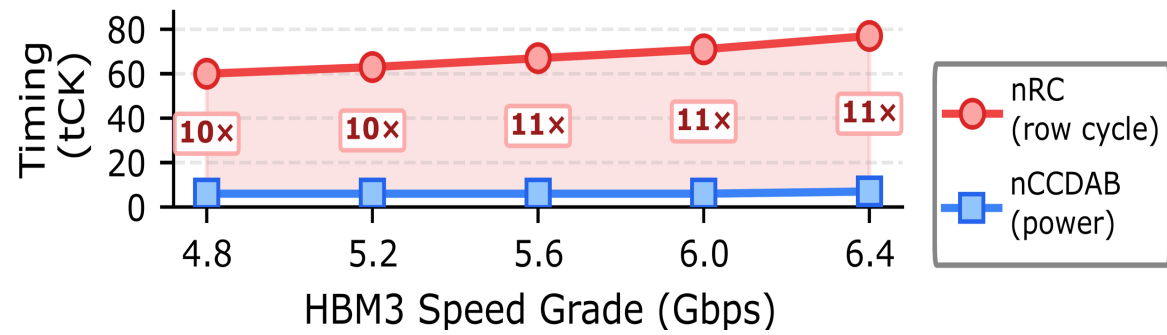
Each MAC lands in a new row, so 60–77 tCK nRC—not 6–7 tCK nCCDAB—sets latency.

**60 – 77
tCK**

**nRC
(row cycle)**

**6 – 7
tCK**

**nCCDAB
(PC)**



10 – 11x gap

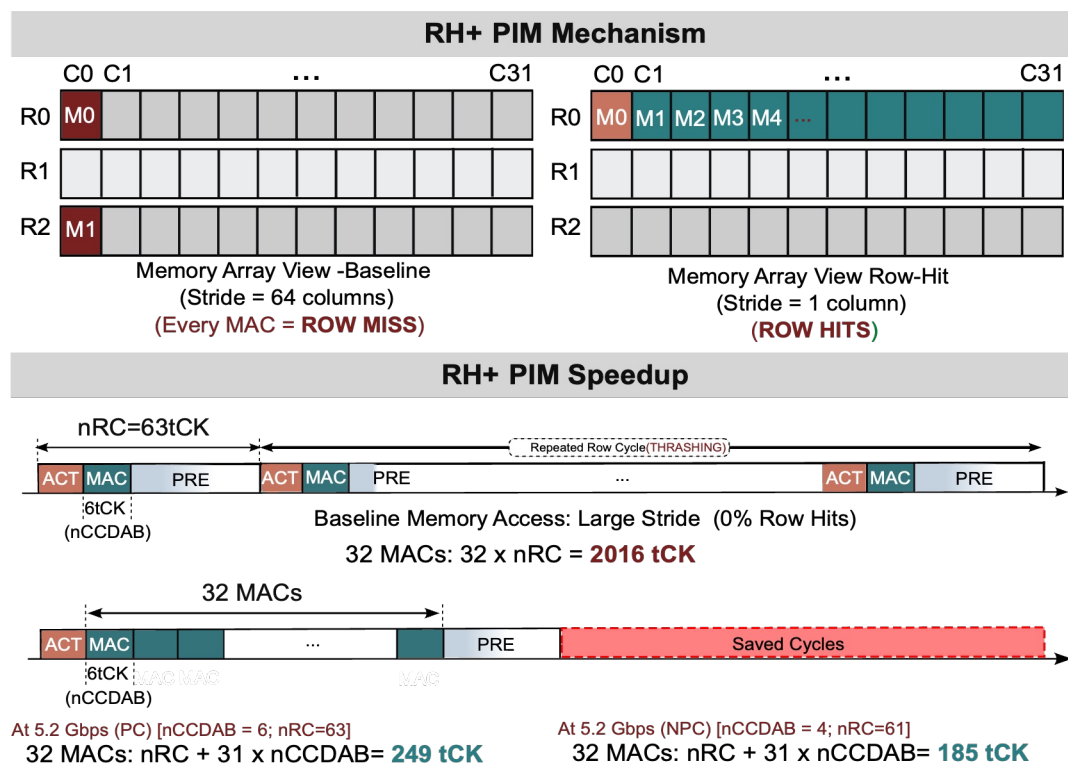
RH+ restores row locality with one stride change

Stride 64 → 1 keeps 32 consecutive MAC_AB commands in the same row.

Baseline

stride = 64

0% row hits



RH+

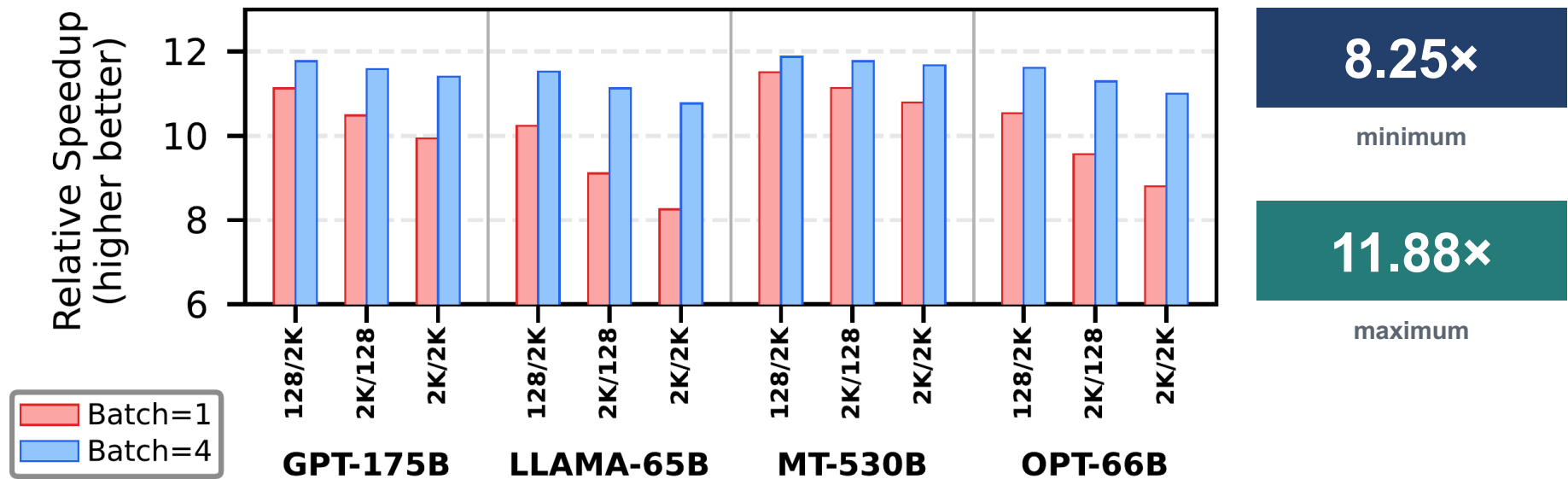
stride = 1

31 / 32 row hits

One-time offline weight reordering · zero runtime overhead

RH+ delivers 8.25–11.88× end-to-end speedup

Larger hidden dimensions and higher GEMV share produce the greatest gains.

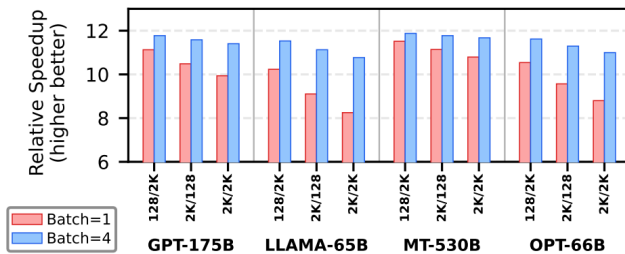


Batch 4 raises GEMV share; MT-530B benefits most from its large K.

Three views of RH+ gains across all configurations

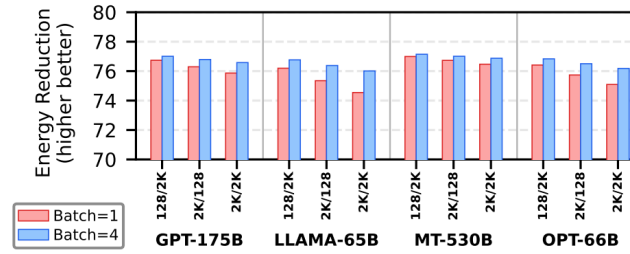
Performance, energy, and EDP improve consistently across four LLMs and 24 configurations.

Speedup



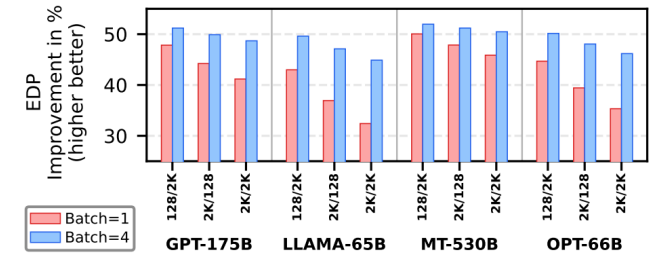
8.25 – 11.88×

Energy reduction



74.5 – 77.1%

EDP improvement



32.4 – 52.0×

RH+ removes row thrashing, reducing both execution time and time-proportional background energy.

Row locality comes before power optimization

RH+ unlocks PIM throughput by fixing the address schedule before tuning the power limit.

8.25 – 11.88×

speedup

74.5 – 77.1%

energy reduction

up to 52×

EDP improvement

nRC, not nCCDAB, is the true GEMV bottleneck under the inherited stride-64 mapping.

A one-column stride keeps 32 MACs in one row and restores the locality PIM needs.

Thank you