

## Problem and Motivation

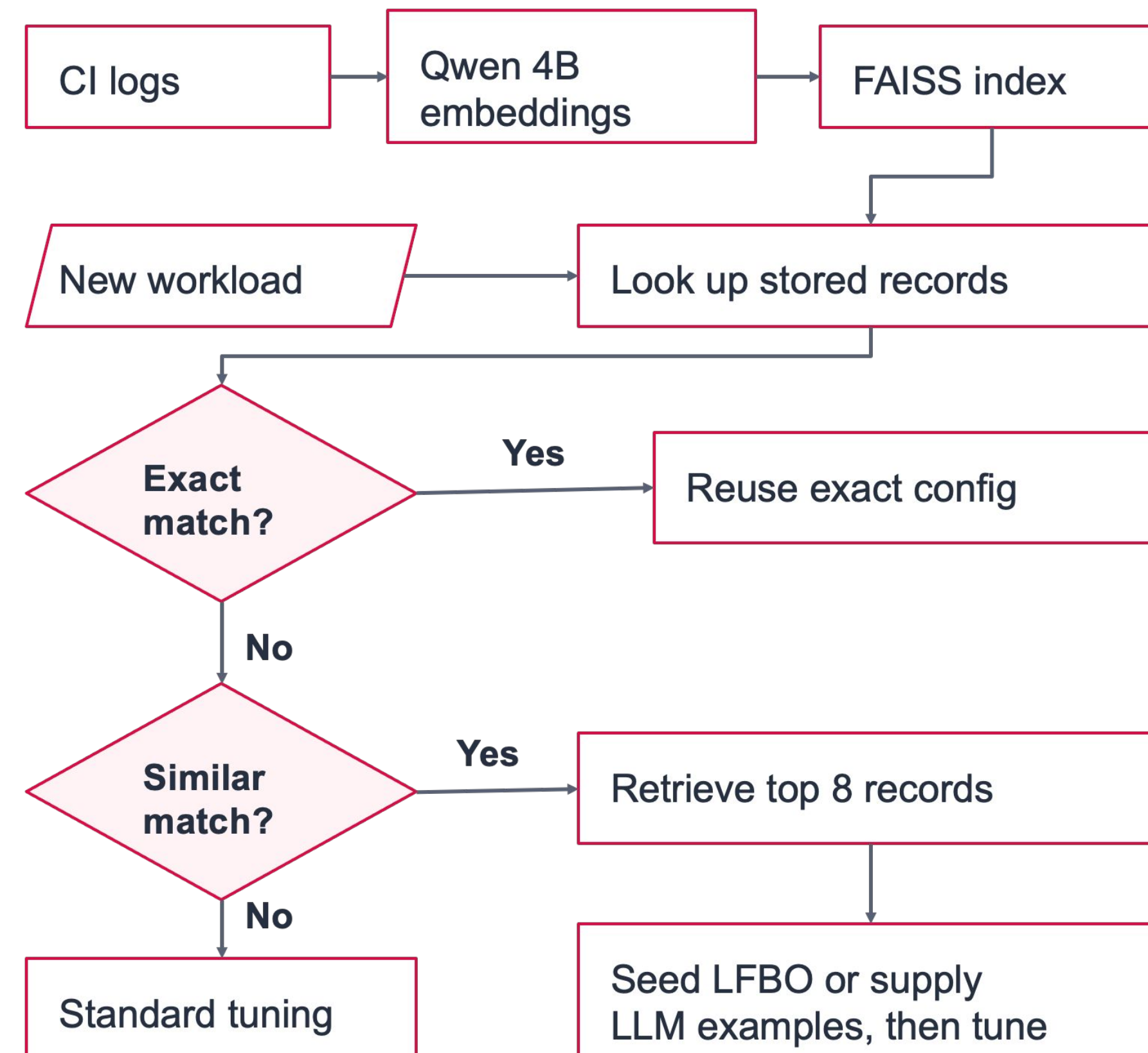
Helion autotuning searches kernel configurations at compile time. Historical results can reduce repeated search, but similar workloads may favor different configurations.

## Retrieval Workflow

1. **Exact match:** reuse a config for the same computation, hardware, shape, and dtype.
2. **Similar match:** retrieve top-K records to seed LFBO or provide LLM examples.
3. **Miss:** tune normally and retain the result.

Retrieval-augmented generation (RAG) provides examples to the large language model (LLM) and is used as the seeds for LFBO autotuner.

## System architecture



Retain newly tuned results in the corpus.

## Retrieval and Evaluation

We retrieve similar past tuning results to suggest starting configurations or provide examples to an LLM. We evaluate whether this reduces total tuning time while preserving the resulting GPU kernel's performance.

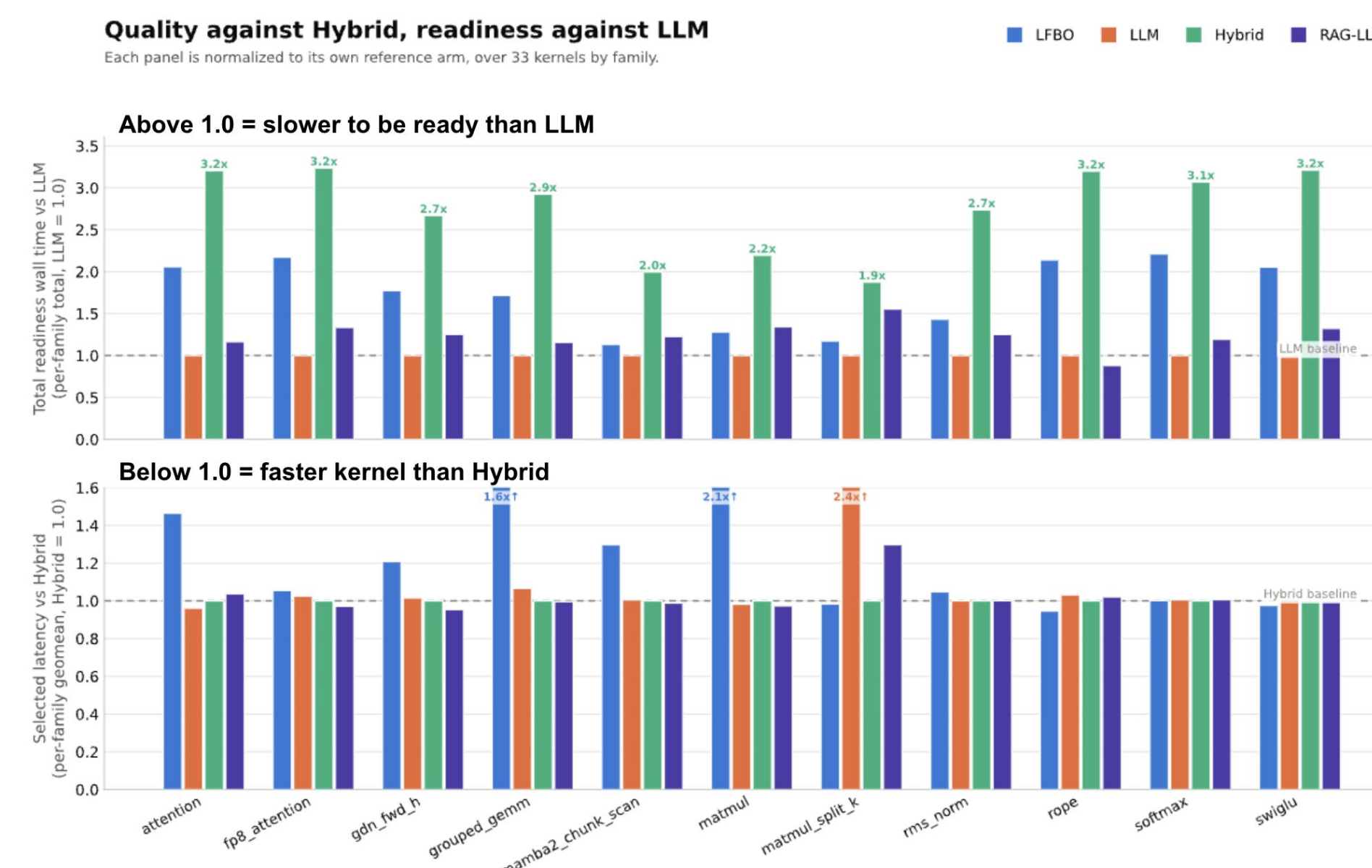
We performed 15 independent runs on H100, alternating whether baseline or retrieval-seeded tuning ran first (AB/BA order). We also collected B200 data. Total tuning time includes embedding, retrieval, validation, compilation, and any LLM calls.

## Tuning Records

We embed five blocks from CI logs:

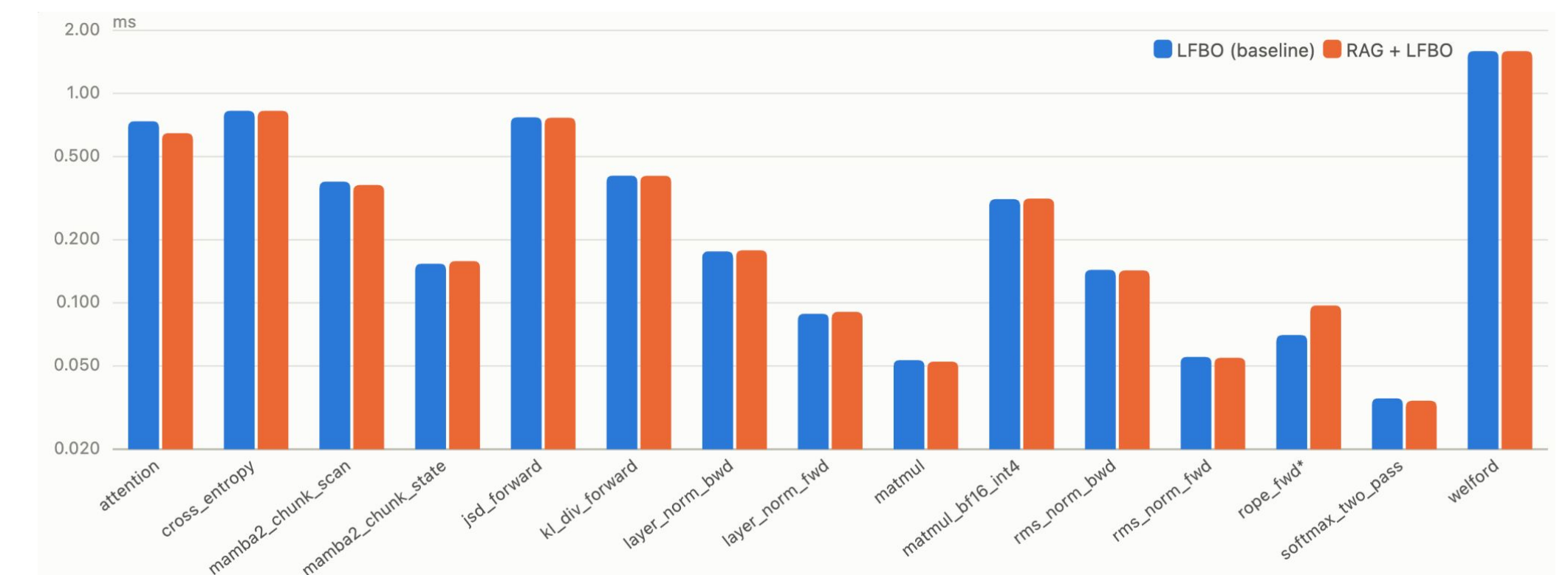
1. **Artifacts:** run ID, kernel/input source, dtypes.
2. **Hardware:** capability, versions, properties.
3. **Settings:** backend options.
4. **IR graph:** NetworkX graph, nodes, edges.
5. **Config/perf:** tried configs, Triton/Pallas code, metrics.

## RAG-seeded LLM Autotuner

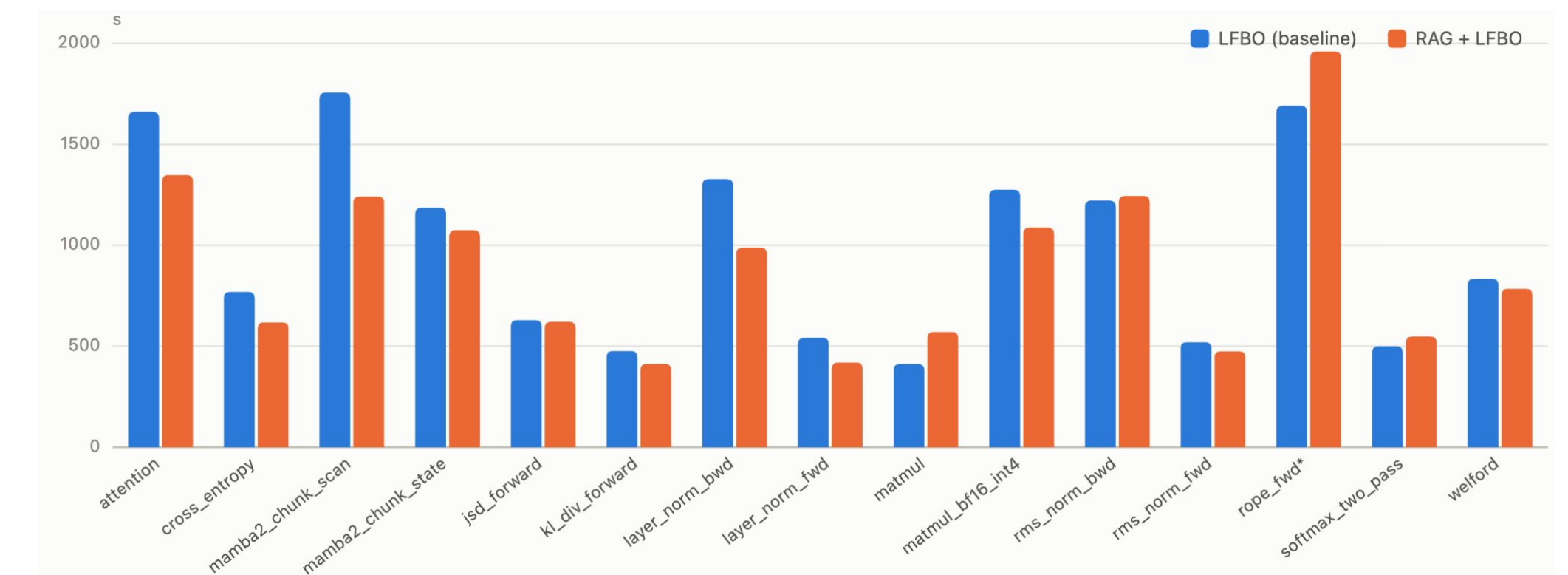


## RAG-seeded LFBO

Kernel latency (in ms, lower is better)



Autotune wall time (in s, lower is better)



Geomean ratios, RAG/LFBO (44 workloads): latency 0.985, tuning 0.930. Lower is better.

## Takeaway

Exact reuse skips autotuning with 1.1% higher geometric-mean kernel latency across 28 workloads. Across 44 workloads, RAG-seeded LFBO reduces geometric-mean tuning time by 7.0% versus LFBO. Benefits vary by workload.



Helion PR  
#3292



Helion PR  
#3304