NEURIPS 2026

Beyond the Trial-and-Error Loop

Hybrid Projection and Automated Tuning for Distributed Training

Anshu Raina, Peyman Razaghi, Yuankai Chen, Cheng Yao, Yao Fu,
Vidushi Goyal, Devang Patel, Wen Xie, Zhenyu Gu, Emad Barsoum

Advanced Micro Devices, Inc.

AMD
together we advance_

THE PROBLEM

Every tuning knob moves several bottlenecks at once

ONE RUN = ONE COUPLED CONFIGURATION

TP · PP · VPP · EP · CP · DP

micro-batch · sequence length · grad accumulation · recompute · FSDP · overlap

Each choice shifts the same four constraints

MEMORY	COMPUTE	NETWORK	PIPELINE
fit ↔ OOM	**GPU utilization**	**collective overhead**	**bubble fraction**

- **Irregular:** throughput is not smooth across knobs, so local tweaks and interpolation mislead.
- **Near-optimal sits at the cliff edge:** one small change flips fit → OOM, or fast → slow.
- **Every bad trial is expensive:** queue time and cluster hours before useful feedback.

MIXTRAL 8×22B · 64 GPUs

~160

TP × PP × EP × CP layouts on 64 GPUs

- **× 8** micro-batch sizes (1–8)
- **× 3** recompute: none / selective / full
- **× 2** MoE backends: All-to-All / DeepEP
- **× 5** pipeline schedules (1F1B … ILP)

~10^4

topologically valid configurations, before checking memory or throughput

AMD
together we advance_

MOTIVATION

Why existing tools don't close the loop

Approach	Examples	Limitation for planning at MoE scale
Analytical / roofline	Calculon, Galvatron	Misses clock throttling, kernel/framework-stack drift, grouped-GEMM & MoE routing effects
Parallelism auto-search	FlexFlow, Alpa	Requires profiling each candidate on target hardware
Trace / simulation	ASTRA-sim, Echo, Lumos	Needs target-hardware traces or heavy calibration; don't scale to sweeps
Config autotuning	DeepSpeed Autotune	Grid-search ZeRO/MBS; requires full scale

Our insight: benchmark a small slice of the job on the real stack; model the rest analytically.

AMD
together we advance_

Measure a slice on real GPUs, model the rest

TARGET



Mixtral 8x22B

TP1 · PP4 · EP8 · 32 GPUs

DOWNSCALE

fit 8 GPUs:
PP 4→1

MEASURE

1 node · TP1 · PP1 · EP8

Dense layer

fwd + bwd timed

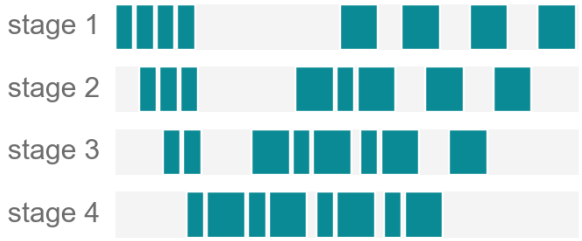
MoE layer

compute | in-node All-to-All

real kernels on the real stack

UPSCALE

RESTORE THE TARGET



replay the PP schedule,
add P2P and DP AllReduce

→ **iteration time, tok/s/GPU**

1 → 8 nodes: −1.4% vs. measured

DOWNSCALE HEURISTIC shrink to the GPUs you have, in order of how exactly each dimension can be restored · CP is never reduced

1. **PP → 1** — schedule replayed exactly
2. **then EP ↓** — experts per GPU kept
3. **then TP ↓** — compute rescaled + AllReduce

MIXTRAL TARGET (TP1 · PP4 · EP8), BY BUDGET

8 GPUs	→	**TP1 · PP1 · EP8 (this example)**
4 GPUs	→	TP1 · PP1 · EP4
1 GPU	→	TP1 · PP1 · EP1

AMD
together we advance_

FIDELITY

Within ~10% of measured, from 1 GPU to 1,024 GPUs

Sub-node sweep: 16 / 16 cells within ~10%

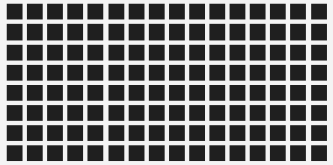
Benchmark on 1–8 GPUs, project the full multi-node target

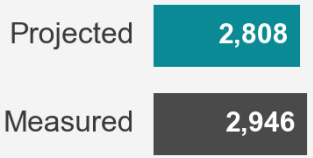
	1 GPU	2 GPUs	4 GPUs	8 GPUs
Llama 3.1 8B FP8 · DP · 64 GPUs	+8.9%	+8.9%	+8.8%	+8.8%
Llama 3.1 70B FP8 · FSDP2 · 64 GPUs	+8.2%	+8.2%	+7.2%	+5.7%
Llama 3.1 70B BF16 · FSDP2 · 64 GPUs	−7.4%	−7.1%	−7.4%	−7.5%
Mixtral 8x22B BF16 · PP4 · EP8 · 32 GPUs	+6.1%	+6.6%	+4.9%	+1.6%

Llama: MI325X. Mixtral: MI355X. Signed error vs. measured tok/s/GPU; darker = larger |error|.

DeepSeek V3 671B

1,024 GPUs · unseen in development · FP8





4.7% error

Qwen3-235B-A22B

256 GPUs · MoE · FP8





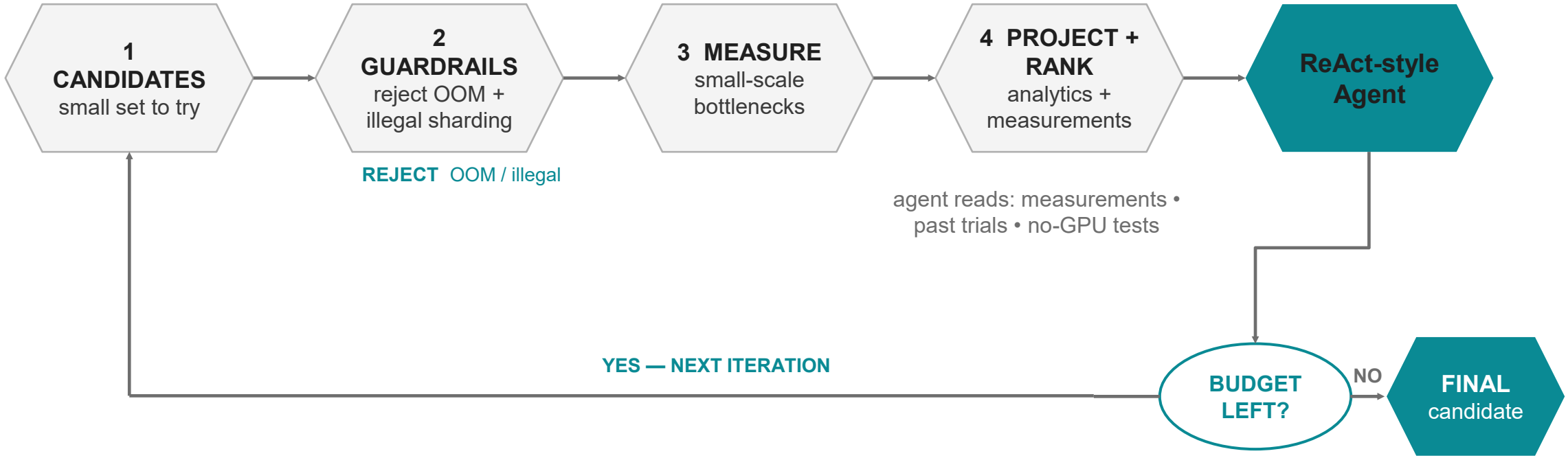
1.9% error

Collective model: 4.2% average error vs. preflight (MI325X, 2 GPUs to 4 nodes, ≥64 MB). tok/s/GPU. 1 square = 1 node of 8 GPUs.

AMD
together we advance_

AGENTIC SEARCH LOOP

Closed loop search using cheap projections and agentic reasoning



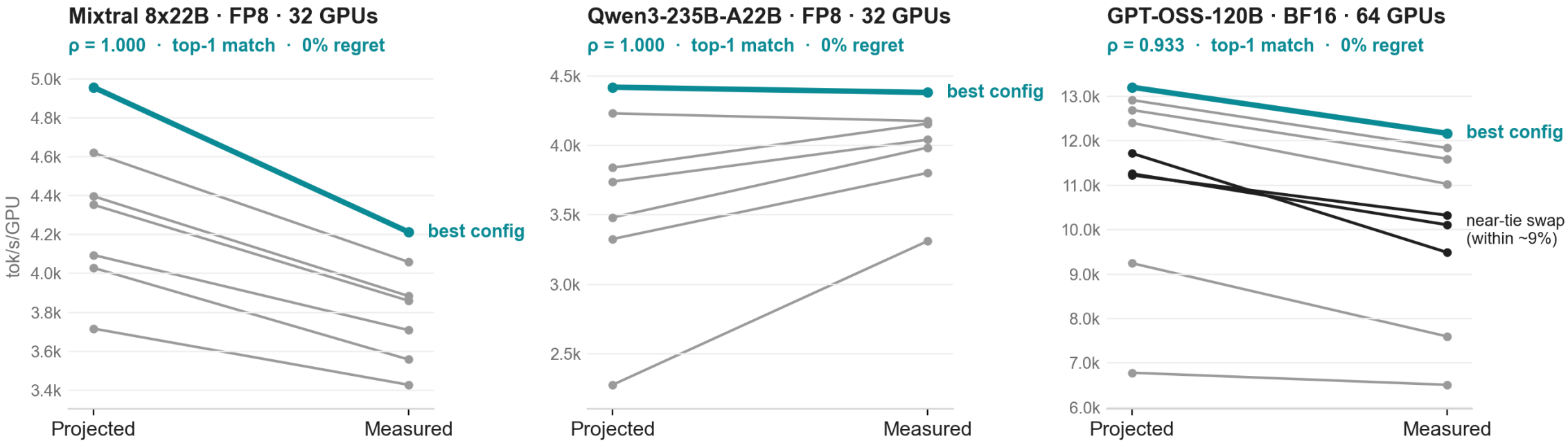
Each cycle learns; expensive cluster runs wait until search converges.

AMD
together we advance_

RANKING FIDELITY

Search needs the right ranking, not perfect absolute numbers

- **The agent's pick is only as good as the measured ordering.** Each line is one config, projected vs. measured.



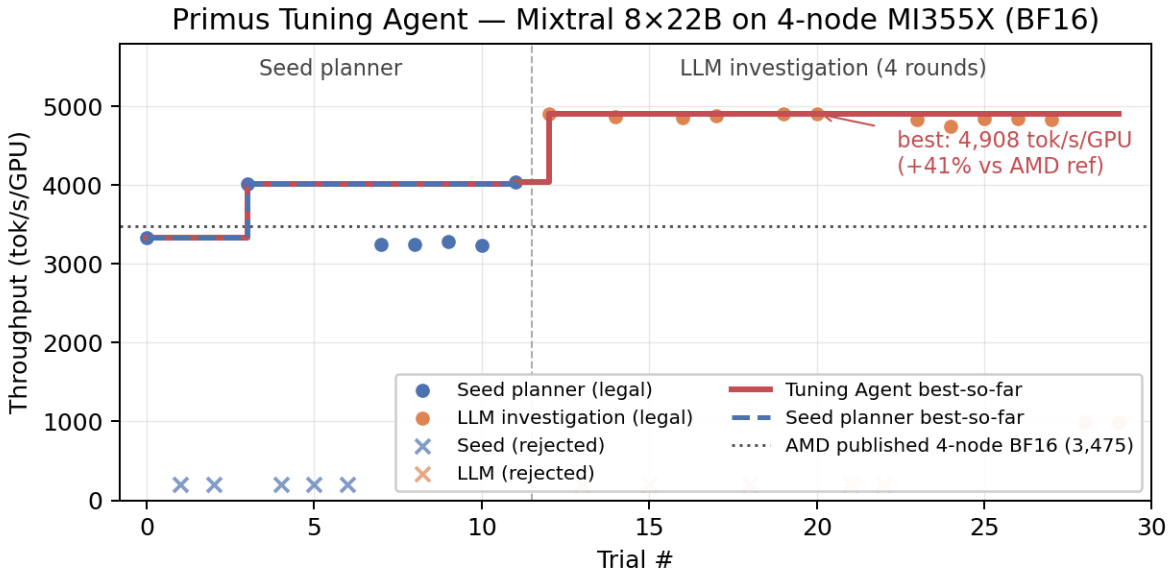
Offsets tilt the lines; they don't reorder them. The measured best is the projected best in all three sweeps.

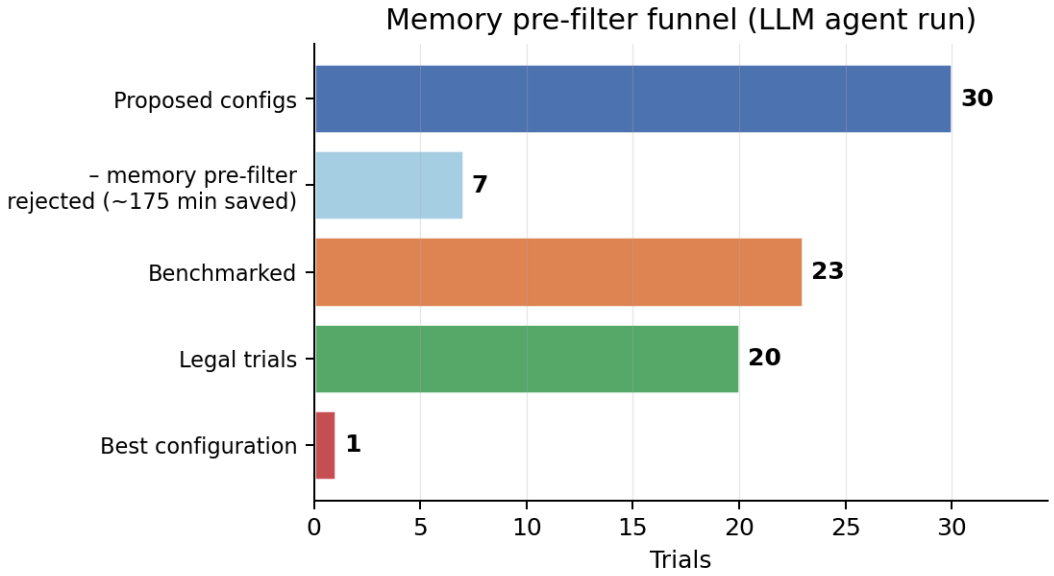
AMD
together we advance_

CASE STUDY

Mixtral 8x22B — +27% over the reference in ~30 minutes of GPU

- **Baseline:** AMD's 4-node BF16 recipe, 3,475 tok/s/GPU.
- **Seed planner:** 4,036 (+16%) from single-axis sweeps.
- **LLM:** MBS=1 + no recompute + DeepEP → 4,402 (+27%).
- **Cheap:** 7 of 30 trials rejected before GPU time; ~30 min.

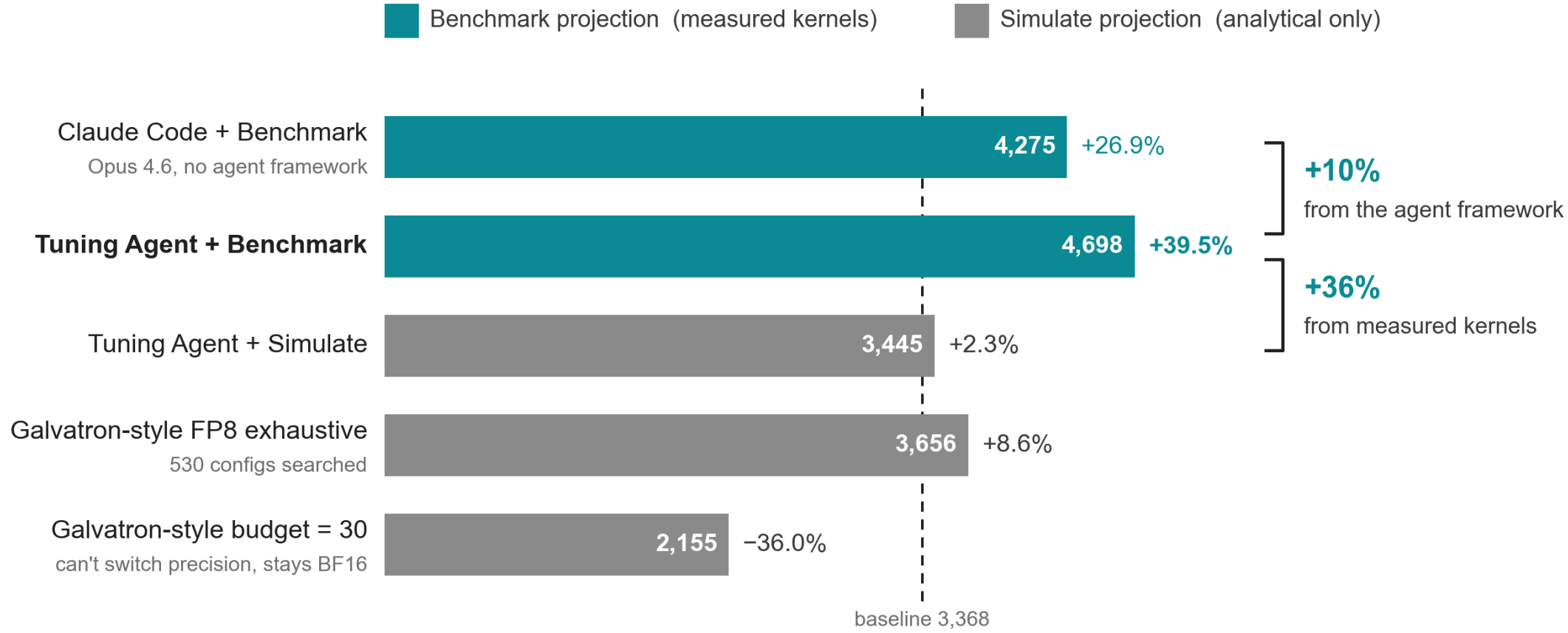




AMD
together we advance_

COMPARISON

What drives the win: measured kernels + agent



Mixtral 8x22B, 4-node MI355X. Projected tokens/s/GPU of the best config each method finds.

AMD
together we advance_

KEY RESULTS

Plan small, train at scale

PROJECTION

~10%

error vs. measured

Dense and MoE, FP8 and BF16, up to 1,024 GPUs, from benchmarks on 1 GPU up to one node.

RANKING

0%

selection regret

Projected #1 config was the measured #1 on all three MoE sweeps.

TUNING AGENT

+27%

over AMD's Mixtral 8x22B recipe

Found in 29 min on one node, and +10% over Claude Code given the same projection.

GPU COST

~40×

fewer GPU-hours to tune

~4 GPU-hours on one node vs. ~160 for trial and error on the 4-node target.

Benchmark a slice → Project the cluster → Let the agent search

AMD
together we advance_

Thank you



PAPER	arxiv.org/abs/2610.19423
CODE	github.com/AMD-AGI/Primus
CONTACT	anshu.raina@amd.com

Scan for the code