BOARD BRIEFING — INFERENCE ECONOMICS

# The Determinism Tax Is Real. Paradatum Eliminates It.

A conservative financial model showing how deterministic inference creates a 2.5× cost penalty — and how Paradatum turns it into a cost advantage.

THE FUNDAMENTAL PROBLEM

## AI That Touches Money Must Be Reproducible

Same model. Same input. Different runs. Different answers. Fine for chat. Unacceptable for capital.

**THE REALITY**

GPU scheduling, floating-point non-associativity, and batch composition make every inference run a coin flip at the bit level. **Identical inputs do not produce identical outputs.**

**FINE FOR CHAT**

If a chatbot rephrases an answer, nobody notices. Nondeterminism is invisible when the stakes are low and outputs are consumed by humans in real time.

**UNACCEPTABLE FOR CAPITAL**

When AI makes pricing decisions, triggers trades, adjudicates claims, or drives autonomous agents — **you must be able to reproduce the result.** Regulators require it. Auditors demand it.

**Compliance**
Cannot audit what cannot be reproduced

**Trading**
Non-reproducible signals are not defensible

**Claims**
Disputes require identical decision outputs

**Agents**
Autonomous systems must be verifiable

**Insurance**
Pricing models must pass actuarial review

**THE CONSEQUENCE**

The industry knows that **nondeterministic AI cannot be trusted with capital decisions.** Billions are being spent to force determinism onto infrastructure never designed for it — freezing hardware, disabling optimized kernels, isolating workloads, re-running inference for verification. It works. But it kills utilization and blocks infrastructure.

**Determinism is being purchased at the cost of economics. That is not scalable.**

INDUSTRY CONTEXT

## Why Determinism Matters — And Why Everyone Gets the Cost Wrong

The industry has recently acknowledged this problem is solvable. Two landmark publications have demonstrated that deterministic inference is achievable through carefully engineered kernels. But both accept a critical tradeoff: **determinism costs performance.**

***The prevailing consensus is that enterprises must choose between reproducibility and economics. We believe this consensus is wrong.***

**Thinking Machines Lab** — SEP 2025

**"Defeating Nondeterminism in LLM Inference"**

**Eigen Labs**

**"EigenAI: Deterministic Inference, Verifiable Results"**

EIGEN LABS — THE TRUE COST THEY DON'T ADVERTISE

Eigen Labs reports a "~2% latency penalty" for deterministic GEMM kernels. This number measures only the isolated kernel-level overhead. It excludes the massive operational constraints their architecture imposes — constraints that, at enterprise scale, dwarf any kernel-level penalty.

**The real cost:** When you add dedicated GPU allocation (no shared batching), homogeneous fleet requirements, frozen software stacks, and verification overhead — the "2% latency" becomes a **40–60%+ total cost of ownership increase** versus standard shared inference. The kernel benchmark is accurate. The economic framing is not.

**PINNED HARDWARE**

**PINNED SOFTWARE STACK**

**NO DYNAMIC BATCHING**

**NO FLEET FLEXIBILITY**

THINKING MACHINES LAB — WHAT "MODEST SLOWDOWN" ACTUALLY MEANS

Thinking Machines deserves credit for proving the thesis: nondeterminism is an engineering bug, not an inevitable hardware limitation. Their batch-invariant kernel library is genuinely valuable. But their published performance numbers require careful reading.

**The framing problem:** Thinking Machines calls the slowdown "modest" and frames it as an acceptable tradeoff. But a 34–61% throughput reduction is not a percentage cost increase — it is a **51–160% cost increase** per token, because cost and throughput are inversely related. At hyperscaler volume, "modest" becomes millions per year. The paper proved determinism is possible. It also proved the industry needs a better path to get there.

**61.5%**
**Overhead WITHOUT CUDA Graphs**
This is the raw cost of their batch-invariant kernels in standard serving mode

**~34%**
**Overhead WITH CUDA Graphs**
Best case reported by SGLang/LMSYS with full optimization pipeline enabled

**2.5×**
**True Cost Multiplier (at 60% slowdown)**
Because cost scales inversely with throughput — not linearly

**KERNEL REPLACEMENT PENALTY**

Batch-invariant ops replace highly optimized vendor kernels (cuBLAS, FlashAttention) with custom implementations that enforce fixed reduction ordering. The overhead is not constant — it **varies by operation and scales with model size**. Matrix multiplications, the dominant cost in LLM inference, carry the heaviest penalty.

**FORWARD PASS ONLY**

The batch-invariant library covers the forward pass. The backward pass (FlashAttention backward) still uses non-deterministic atomic operations. For RL training pipelines that need end-to-end reproducibility, **additional engineering and further performance penalties** are required beyond what's published.

**CUDA GRAPHS DEPENDENCY**

The optimistic "34%" number requires CUDA Graphs, which pre-compiles kernel launch sequences. This **limits dynamic behavior** — variable sequence lengths, changing batch sizes, and model-switching all become constrained. Many production serving patterns are incompatible with static graph capture.

**VALIDATED ON LIMITED ARCHITECTURES**

Published results focus on Qwen3-8B and Qwen3-235B-A22B (MoE). Performance characteristics **may differ significantly** on dense 70B+ models (Llama, Mistral) where matmul dominance is higher and the batch-invariant penalty compounds across more layers.

**Both papers validate that determinism is achievable and that the market demands it.** But both understate the true economic impact. Eigen Labs hides operational costs behind a kernel benchmark. Thinking Machines hides a 2.5× cost multiplier behind the word "modest." The industry consensus is that determinism requires either rigid infrastructure constraints or significant throughput penalties — **pick your poison.** Paradatum rejects both. Through lossless BF16 compression that accelerates computation while guaranteeing bit-exact reproducibility, Paradatum doesn't just eliminate the determinism tax. **It makes deterministic inference faster and cheaper than today's non-deterministic baseline.**

KEY FINDINGS

## Three Numbers That Matter

DETERMINISM COST PENALTY
**2.5×**
cost increase per token

GPU FLEET INCREASE
**2×**
more hardware required

PARADATUM ADVANTAGE
**5×**
cheaper than deterministic baseline

THE CORE MATH

## Why 60% Slower Is Actually 150% More Expensive

Cost scales inversely with throughput. A 60% speed reduction is a 150% cost increase.

**Shared Baseline** — 200 t/s — $4.08

**Deterministic (−60%)** — 80 t/s — $10.20

**Paradatum (2×)** — 400 t/s — $2.04

Cost per million tokens → longer bar = higher cost

**Key insight:** The industry frames the determinism tradeoff as a "60% slowdown." In reality, it's a **150% cost increase** because throughput and cost have an inverse relationship. Every token takes 2.5× longer to produce, which means 2.5× more GPU-hours billed.

SIDE-BY-SIDE COMPARISON

## Unit Economics at 650M Tokens/Month

Conservative production assumptions on H100 infrastructure.

| MODE | TOKENS / SEC | GPUS REQUIRED | COST / 1M TOKENS | MONTHLY COST | DELTA |
|---|---|---|---|---|---|
| ●Shared Baseline | 200 | 2 | $4.08 | $3,650 | Base |
| ●Deterministic (−60%) | 80 | 4 | $10.20 | $7,300 | +150% |
| ●Paradatum (2×) | 400 | 1 | $2.04 | $1,825 | −50% |

HARDWARE IMPACT

## GPU Fleet Requirement — Same Workload

At 650M tokens/month, the hardware delta is 4× between deterministic and Paradatum.

SHARED BASELINE
H100 H100
**$3,650**
per month

DETERMINISTIC (−60%)
H100 H100 H100 H100
**$7,300**
per month — 2× fleet

PARADATUM
H100
**$1,825**
per month — 1 GPU

AT SCALE

## Hyperscaler Economics — Single Enterprise Instance

One large enterprise customer running 10B tokens/day (300B tokens/month). This is one workload on one platform.

**The math:** 300B tokens/month ÷ effective tokens/hour per GPU ÷ 730 hours/month = GPU count. GPU count × $2.50/hr × 730 = monthly cost. At $4.08/M tokens baseline: 300B × $4.08/M = **~$1.2M/month**. At $10.20/M deterministic: 300B × $10.20/M = **~$3.0M/month**. At $2.04/M Paradatum: 300B × $2.04/M = **~$600K/month**.

SHARED BASELINE
**$1.2M**
per month / per instance

DETERMINISTIC (−60%)
**$3.0M**
per month / per instance

PARADATUM
**$600K**
per month / per instance

**Per-instance savings vs. deterministic baseline: ~$2.4M/month → ~$29M/year.** That is one customer, one workload. A hyperscaler serving 10–50 enterprise customers at this volume tier faces an aggregate determinism tax of $30–150M/year. Paradatum doesn't reduce that tax. It reverses it — turning a cost penalty into a cost advantage across every instance on the platform.

STRATEGIC POSITIONING

## What This Means

Paradatum doesn't compete on correctness alone. It competes on cost collapse.

### Industry Status Quo

Thinking Machines and others have proven determinism is possible — but it comes with a ~60% speed penalty, which translates to a 2.5× cost increase. Enterprises must choose between reproducibility and economics.

### The Paradatum Thesis

Determinism can be cheaper than non-determinism. Lossless BF16 compression with 2× acceleration means enterprises get bit-exact reproducibility and lower costs simultaneously. No tradeoff required.

### Conservative Floor

Even if full-stack performance lands at 1.5× (not 2×), Paradatum still beats the deterministic penalty by a wide margin. At parity, it eliminates the tax entirely. Even parity is a win.

### Market Implication

This is not niche infrastructure tooling. This is hyperscaler economics — where cost-per-token determines which platforms can profitably serve regulated industries at scale.

**Determinism is the market.**
**The execution layer is the moat.**

**Paradatum eliminates the tradeoff between reproducibility and economics, turning a 2.5× cost penalty into a 50% cost reduction.**

✓ Conservative assumptions
✓ Production-grade numbers
✓ Validated kernel-level performance