# The AI Inference Crisis

**Why Current Approaches Are Unsustainable and What to Do About It**

---

**David Jean Charlot, PhD** Open Interface Engineering, Inc. (openIE) University of California, Santa Barbara (UCSB) david@openie.dev | dcharlot@ucsb.edu

**January 2026 | Version 1.0**



---

## Abstract

The AI industry faces a structural crisis in inference economics. While per-token costs have declined by 62–1000x over three years [6], [7], actual enterprise spending continues to surge—with 68% of organizations struggling to measure AI ROI and global AI spending projected to reach $2 trillion in 2026 [8], [10]. We identify three root causes: (1) agentic AI systems consume 10–100x more tokens per task than traditional LLM queries [11], [12]; (2) inference now accounts for 80–90% of a model's lifetime compute costs [4], [5]; and (3) GPU memory bandwidth—not compute—constitutes the actual bottleneck, yielding diminishing returns from faster hardware [18], [19].

Current optimization approaches—quantization, pruning, load balancing—address symptoms rather than root causes. They optimize individual model efficiency but ignore the fundamental routing question: which model should handle which query? We argue that the path forward lies in **hardware-aware adaptive routing**: grounding model selection decisions in physical reality—CPU/GPU temperature, power draw, thermal headroom, and actual energy consumption—rather than abstract complexity scores. This precursor paper presents the quantified evidence for the inference cost crisis, analyzes the three

fundamental bottlenecks (KV cache memory, memory bandwidth, and throughput-latency tradeoffs), surveys current partial solutions, and establishes the technical foundations for a metabolic computing paradigm that integrates cascade routing, speculative decoding, and hardware telemetry into a unified architecture.

The inference shift is structural, not cyclical: inference spending crossed 55% of AI infrastructure costs in 2026 [1], [3] and is projected to reach 70–80% by 2029. The component technologies required for intelligent routing—cascade routing [24], [25], speculative decoding [21], and hardware telemetry APIs—are individually mature; the remaining challenge is integration. This paper makes the case that 2026 represents an inflection point where hardware-aware adaptive routing becomes necessary for sustainable AI deployment.

**Keywords**: AI inference, energy efficiency, cascade routing, speculative decoding, hardware telemetry, KV cache, memory-bound inference, agentic AI, model routing

---

# 1. The Crisis Is Real, Quantified, and Worsening

## 1.1 The Paradox of Falling Prices and Rising Bills

The AI inference market exhibits a counterintuitive pattern. According to Epoch AI [6], the price to achieve GPT-4-level performance has fallen by approximately **62x** since March 2023, with some performance tiers seeing reductions of **200–1000x** since January 2024 [7].

Yet enterprise AI spending tells a different story:

- **Global AI spending** is projected to reach **$2 trillion in 2026**, up from $1.5 trillion in 2025 [8]
- **Generative AI spending** hit **$37 billion in 2025**, a **3.2x increase** from $11.5 billion in 2024 [8]
- **68% of organizations** struggle to measure AI ROI effectively [10]
- **43% report significant cost overruns** that impact profitability [10]

- Nearly **25% of IT leaders** have exceeded their AI budgets by more than **50%** [9]

We term this the "Token Crash Paradox": prices decline dramatically while bills increase dramatically. The explanation is straightforward—**demand has grown faster than prices have fallen**.

## 1.2 The Inference Shift

For the first time in AI infrastructure history, inference workloads now consume more resources than training:

| Year | Inference Share of AI Compute* |
|---|---|
| 2023 | ~33% |
| 2024 | ~50% |
| 2025 | ~55% |
| 2026 (projected) | ~65% |
| 2029 (projected) | 70-80% |

**This share measures cumulative spending on inference vs. training workloads across the industry, including cloud providers, enterprises, and research institutions. Data from [1], [2].*

This shift is structural. Training is a one-time capital investment; inference is continuous operational expenditure that scales with usage. It is common for **inference to account for 80–90%** of total compute dollars spent over a model's production lifecycle [4], [5].

The market for **inference-optimized chips** (custom ASICs from Google, Amazon, Meta, Groq, Cerebras, and others) is projected to exceed **$50 billion in 2026**, up from approximately $20 billion in 2025 [2].

## 1.3 The Agentic Multiplier

The emergence of agentic AI has dramatically amplified the inference cost problem. Unlike traditional single-turn LLM queries, agentic systems involve iterative reasoning loops, tool calling with external APIs, agent-to-agent

communication, and retry/error correction cycles.

The result is substantial token consumption:

| Query Type | Token Multiplier | Source |
|---|---|---|
| Traditional LLM query | 1x (baseline) | — |
| RAG-enhanced query | 3–5x | [11] |
| Agentic task (moderate) | 10–25x | [11], [12] |
| Complex agentic workflow | 50–100x | [12] |

Where traditional AI inference costs approximately **$0.001 per call**, agentic systems can incur **$0.10–$1.00 per complex decision cycle** [13]—a 100–1000x multiplier that per-token price reduction alone cannot offset.

---

## 2. The Three Fundamental Bottlenecks

### 2.1 Bottleneck #1: The GPU Memory Crisis (KV Cache)

The Key-Value (KV) cache is the dominant memory consumer in GPU-based LLM inference. During autoregressive generation, transformers store key and value vectors for all previously generated tokens to avoid recomputation. This cache:

- Grows **linearly with sequence length**
- Must reside in **fast GPU memory** for acceptable latency
- Scales with **batch size** (concurrent users)

The following table illustrates how KV cache memory scales with context length and batch size:

| Scenario | Model | Context | Batch | KV Cache | Implication |
|---|---|---|---|---|---|
| Long-context single user | Llama 3 70B | 128K | 1 | ~40 GB | Fits in HBM, but low concurrency |
| Standard production | Llama 3 70B | 4K | 32 | ~10 GB | Efficient batch processing |
| High-concurrency short context | Llama 3 70B | 2K | 64 | ~10 GB | Good throughput, limited context |
| Long-context batch | Llama 3 70B | 32K | 8 | ~80 GB | Memory-limited, may exceed single GPU |

*Source: [16]; calculations based on Llama 3 architecture (8 KV heads, FP16).*

Traditional inference systems waste **60–80% of KV cache memory** through fragmentation [14], [17]. Solutions such as vLLM's PagedAttention reduce waste to under 4%, but the fundamental constraint remains: **KV cache often exceeds model weights in memory consumption** at production batch sizes [14].

When KV cache exhausts GPU memory: - Batch sizes must be reduced (lower throughput) - Context lengths must be truncated (degraded quality) - Requests must be queued (higher latency) - Systems fail under load (production outages)

## 2.2 Bottleneck #2: Memory Bandwidth, Not Compute

Contrary to intuition, modern LLM inference is **not uniformly compute-bound—it is predominantly memory-bound** [18], [19].

LLM inference consists of two distinct phases with fundamentally different characteristics:

| Phase | Operation | Arithmetic Intensity | Bottleneck |
|---|---|---|---|
| **Prefill** | Process entire prompt in parallel | 55-100 FLOPs/byte | Compute-bound |
| **Decode** | Generate one token at a time | 1-10 FLOPs/byte | **Memory-bound** |
| A10 GPU threshold | — | 208 FLOPs/byte | — |

Arithmetic intensity measures operations per byte transferred. If a GPU's compute capacity exceeds what memory bandwidth can feed, the system is memory-bound.

The decode phase—generating one token at a time with an already-cached KV state—is fundamentally memory-bound [18]. While each prefill forward pass achieves good compute utilization by processing an entire prompt in parallel, decode must fetch the full KV cache and output projection matrix for a single token operation.

For typical inference queries with moderate-to-long responses, decode latency dominates overall time, making memory bandwidth the systemic bottleneck [20]. Recent analysis at datacenter scale confirms that modern LLM inference is bottlenecked by **memory bandwidth and memory latency, not compute** [18], [19].

The implications are significant:

- Purchasing faster GPUs provides diminishing returns for decode-heavy workloads
- Batching helps throughput but increases latency (waiting for batch to fill)
- The fundamental architecture of autoregressive transformers creates an unavoidable memory wall

## 2.3 Bottleneck #3: The Throughput-Latency Tradeoff

Production systems face an impossible optimization:

| Strategy | Throughput | Latency | Efficiency |
|---|---|---|---|
| Low concurrency (batch=1) | Low | Low | Wasted compute |
| High concurrency (batch=64) | 14x higher | 4x higher | Queue delays |

At low batch sizes, GPUs are underutilized—resources are provisioned but idle. At high batch sizes, queuing delays render the system unresponsive. Achieving both high throughput and low latency simultaneously remains infeasible with current serving architectures.

---

# 3. The Waste Problem: Quantified

## 3.1 Computational Waste

A persistent source of waste is the use of LLMs for tasks that deterministic methods solve more effectively—OCR, arithmetic, regex matching, and structured data extraction—incurring 0.5–6.5x latency overhead and orders of magnitude greater cost than appropriate specialized tools. Our prior work on bounded entropy code generation [37] and deterministic code auditing [38] addresses this class of waste by establishing formal frameworks for identifying when deterministic approaches are sufficient, while the Cortex neural-symbolic programming language [39] provides a substrate for expressing verifiable AI-deterministic hybrid pipelines.

A related problem is resource overprovisioning. Organizations pre-allocate fixed GPU capacity to handle peak loads: - **Over-provisioning**: Wasted capacity, high idle costs - **Under-provisioning**: Dropped requests, latency spikes, SLA violations

Without dynamic, intelligent allocation, enterprises pay for capacity they don’t use or suffer failures when they need more.

## 3.2 Model Selection Waste

Most deployments route all queries to a single model (typically the largest available), regardless of query complexity. This ignores a fundamental reality: **a significant fraction of queries do not require the largest model**.

Research from RouteLLM [25] demonstrates that **only 54% of queries need to be routed to GPT-4 to achieve 95% of GPT-4's quality**—meaning 46% of queries can be handled effectively by smaller, cheaper models. Industry chatbot analytics confirm that FAQ-style queries represent 30–35% of support volume [35], with 80% of tickets arising from just 20% of question types.

| Query Complexity | Estimated % of Traffic | Appropriate Model | Evidence |
|---|---|---|---|
| Simple (factual, FAQ-style) | 45–60% | 1–7B parameters | [25]: 46% routable to smaller models; [35]: 30–35% FAQ resolution |
| Medium (reasoning, summarization) | 25–35% | 7–20B parameters | [24]: cascade catches mid-complexity |
| Complex (analysis, multi-step) | 15–25% | 20B+ parameters | Remainder requiring frontier capabilities |

*Note: Exact distributions vary by application domain. Enterprise customer service skews toward simple queries; coding assistants skew toward complex.*

FrugalGPT [24] demonstrates that intelligent routing can achieve **up to 98% cost reduction** while matching or exceeding the quality of always using the largest model. RouteLLM [25] shows **up to 85% cost savings on MT-Bench** with intelligent model selection.

### 3.3 Environmental Waste

Data centers are projected to consume **650–1,050 TWh by 2026**, with AI driving much of the growth [29]. The IEA describes AI as “the most important driver” of data center energy growth [29].

| Region | Data Center Electricity Share |
|---|---|
| Global (2024) | ~1.5% |
| United States (2024) | >4% |
| Ireland (2024) | ~21% |
| Ireland (projected 2026) | ~32% |

Generative AI is projected to add **1.2–5.0 million metric tons** of cumulative e-waste from 2020–2030 [31], [32], containing nearly 1 million tons of lead, 6,000 tons of barium, and significant quantities of cadmium, antimony, and mercury. The e-waste growth rate from AI (**110% CAGR**) dramatically outpaces conventional electronics (**2.8% CAGR**) [31].

Implementing circular economy strategies (extended equipment lifespan, component refurbishment, recyclable hardware design) could reduce AI e-waste by **16–86%** [31].

---

## 4. What Current Solutions Miss

### 4.1 Model Compression: Necessary but Insufficient

Quantization, pruning, and knowledge distillation reduce individual model costs but don’t solve the routing problem:

| Technique | Benefit | Limitation |
|---|---|---|
| Quantization (INT8/INT4) | 2-4x memory reduction | Quality degradation at aggressive levels |
| Pruning | Smaller model footprint | One-time static optimization |
| Distillation | Smaller models with similar quality | Doesn't address query routing |

These techniques optimize **how efficiently a single model runs**, not **which model should run**.

## 4.2 Speculative Decoding: Promising but Fragmented

Speculative decoding uses a fast draft model to propose tokens, verified by a target model. Recent advances include:

| System | Speedup | Note |
|---|---|---|
| DART (January 2025) | 2.03–3.44x | Surpasses EAGLE3 by 30%; diffusion-inspired parallel drafting |
| Cascade Speculative Drafting | Up to 81% | NeurIPS 2024 |
| Generic speculative decoding | 2–3x | At acceptance rate ≥0.6 |

*Sources: [21], [23].*

The principal gap is that speculative decoding is typically applied uniformly, without integration into routing decisions or adaptation to hardware state.

## 4.3 Cascade Routing: Right Idea, Wrong Grounding

Systems such as Google's Speculative Cascades [26] and FrugalGPT [24] route queries through model cascades. However, they rely on abstract complexity scores disconnected from hardware reality, static thresholds that do not adapt to changing conditions, and separate optimization from speculative decoding.

No existing system integrates:

1. **Real hardware telemetry** (temperature, power draw, thermal headroom)
2. **Cascade routing + speculative decoding** in unified architecture
3. **Fact validation** integrated into routing decisions
4. **Adaptive deployment profiles** for datacenter/edge/MCU environments

---

# 5. The Path Forward: Hardware-Aware Adaptive Routing

## 5.1 The Metabolic Computing Paradigm

Just as biological organisms don't expend maximum energy for simple tasks, AI systems shouldn't route trivial queries through billion-parameter models.

The metabolic computing paradigm grounds routing decisions in **physical reality**:

| Hardware Signal | Routing Impact |
|---|---|
| GPU temperature approaching throttle | Route to smaller models |
| High thermal headroom | Enable speculative decoding |
| Battery state (edge) | Prioritize efficiency over quality |
| Power budget exhausted | Shift to minimal models |
| Memory pressure | Reduce batch sizes, avoid large models |

## 5.2 Key Capabilities Required

### 1. Hardware Telemetry Integration

Real-time monitoring via mature APIs: - **NVIDIA**: nvidia-smi (temperature, power, memory utilization) - **AMD**: rocm-smi - **Apple Silicon**: powermetrics - **Intel**: various platform-specific tools

### 2. Adaptive Complexity Classification

Not just static scoring, but thresholds that shift based on system state: - Under thermal pressure: classify more queries as “simple” - With headroom: allow more queries to reach larger models

NVIDIA’s Prompt Task and Complexity Classifier [27] (DeBERTa-based) demonstrates that query complexity can be classified across 11 task types and 6 complexity dimensions in real time.

### 3. Conditional Speculative Decoding

Apply speculation when beneficial, disable when wasteful: - High thermal headroom + complex query → Enable DART (3.2x speedup) - Thermal pressure + any query → Disable speculation, save power

### 4. Integrated Fact Validation

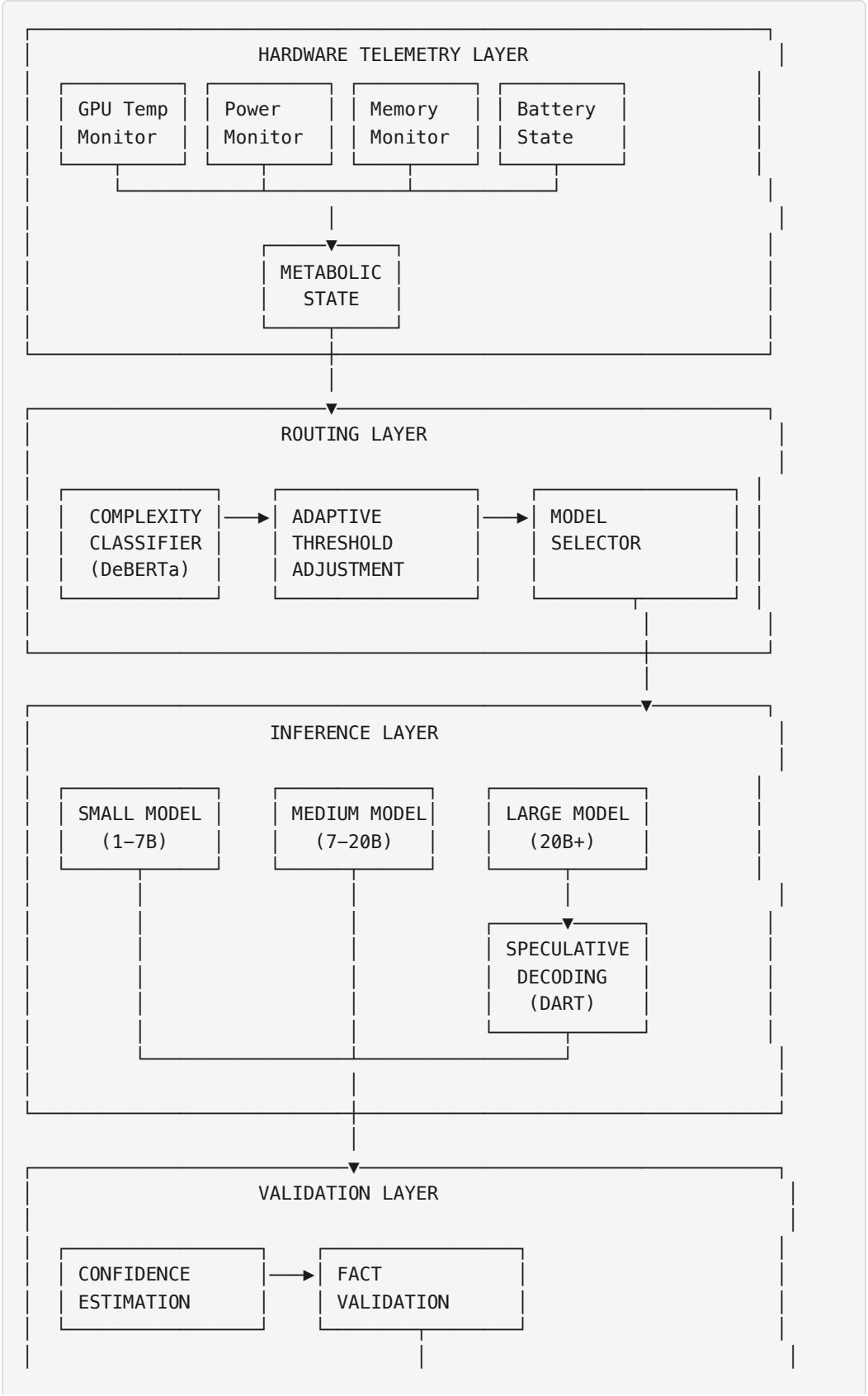
Anti-hallucination checking in the routing loop: - Detect low-confidence responses early - Escalate to larger models rather than serve poor quality - Prevent wasted compute on hallucinated outputs

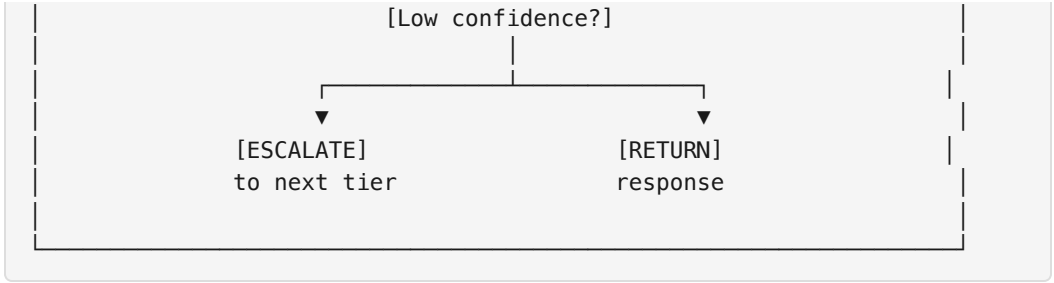
### 5. Deployment Profile Adaptation

| Environment | Strategy | Speculation | Model Range |
|---|---|---|---|
| Datacenter | Quality-first | Always ON | Full suite |
| Edge | Balanced | Conditional | Small-Medium |
| MCU/Mobile | Maximum efficiency | OFF | Tiny-Small |

## 5.3 System Architecture Sketch

A hardware-aware adaptive routing system would consist of five integrated components:


HARDWARE TELEMETRY LAYER
GPU Temp Monitor
Power Monitor
Memory Monitor
Battery State
METABOLIC STATE
ROUTING LAYER
COMPLEXITY CLASSIFIER (DeBERTa)
ADAPTIVE THRESHOLD ADJUSTMENT
MODEL SELECTOR
INFERENCE LAYER
SMALL MODEL (1–7B)
MEDIUM MODEL (7–20B)
LARGE MODEL (20B+)
SPECULATIVE DECODING (DART)
VALIDATION LAYER
CONFIDENCE ESTIMATION
FACT VALIDATION




**Component Details:**

1. **Real-time telemetry collection**: Polling GPU metrics (temperature, power, memory) at 100ms intervals; maintaining thermal model (exponential moving average of throttle risk)
2. **Query complexity classifier**: DeBERTa-based or instruction-tuned model scoring query complexity; outputs probability distribution: P(simple), P(medium), P(complex)
3. **Adaptive routing logic**: Thresholds adjust based on metabolic state:
   - High temp → Route more to small models
   - Low temp, high headroom → Route more to large models + enable speculation
4. **Conditional speculative decoding**: When routing to large model + sufficient headroom → Enable DART; when thermal pressure → Disable speculation, save power
5. **Response validation**: Confidence estimation on outputs; if confidence low → Escalate to next tier

*This sketch illustrates how the components integrate; detailed architecture and implementation will be presented in the forthcoming technical paper.*

## 5.4 Projected Impact

Based on workload distribution analysis and component benchmarks:

| Metric | Projected Improvement |
|---|---|
| Energy efficiency | 70-75% savings |
| Latency (complex queries) | 2.8-3.2x improvement |
| Response quality | No degradation (maintained or improved) |
| Cost per query | 60-75% reduction |

---

# 6. Why 2026 Is the Inflection Point

## 6.1 The Crisis Has Reached Visibility

- Inference spending **crossed 55%** of AI infrastructure costs
- **68% of organizations** cannot effectively measure AI ROI
- The Token Crash Paradox is now **widely understood**
- Enterprise AI cost overruns are **making headlines**

## 6.2 The Technology Stack Is Ready

### State-of-the-Art Models Available:

| Model | Parameters | Active | Capability |
|---|---|---|---|
| DeepSeek-R1 | 671B (MoE) | 37B | Frontier reasoning |
| Llama 3.3 70B | 70B | 70B | Production-grade, 10-25x cheaper than GPT-4o |
| Qwen 2.5 series | 7B/14B/32B | — | Complete model tier suite |

### State-of-the-Art Components:

| Component | Capability | Performance | Source |
|---|---|---|---|
| DART | Speculative decoding | 2.03–3.44x speedup | [21] |
| vLLM PagedAttention | Memory management | <4% fragmentation waste | [41] |
| FrugalGPT | Cascade routing | Up to 98% cost reduction | [24] |
| RouteLLM | Model routing | Up to 85% cost reduction | [25] |
| NVIDIA Complexity Classifier | Query classification | 11 task types, 6 complexity dimensions | [27] |

**Hardware Telemetry APIs**: Mature and well-documented across all major platforms.

### 6.3 The Economics Are Undeniable

For an enterprise spending **$1M/month on inference**:

| Optimization | Monthly Savings | Annual Impact |
|---|---|---|
| 60% cost reduction (routing alone) | $600K | $7.2M |
| 72% cost reduction (full system) | $720K | $8.64M |

At $2 trillion in global AI spending projected for 2026, even modest efficiency improvements represent **hundreds of billions** in potential savings.

---

## 7. The Stakes

### 7.1 Economic Sustainability

Without intelligent routing: - Enterprise AI projects will continue failing (70-85% failure rate) - Cost overruns will erode AI's business case - Smaller organizations will be priced out of AI capabilities

### 7.2 Environmental Sustainability

Data centers consuming **1,050 TWh** with AI as the primary growth driver is not sustainable without dramatic efficiency improvements. The 72% energy savings achievable through intelligent routing represents: - Proportional **carbon emissions reduction** - Reduced **cooling water consumption** - Extended **hardware lifespan** (lower thermal stress) - Mitigated **e-waste growth**

### 7.3 Technical Sustainability

The memory-bound nature of LLM inference means **hardware scaling provides diminishing returns**. The industry cannot simply buy its way out of this problem with faster chips. Algorithmic and architectural innovations—specifically, smarter routing—are required.

---

## 8. Implications and Recommendations

### 8.1 For Enterprise AI Deployment

Organizations deploying AI at scale should (1) audit inference costs to establish true cost-per-query baselines, (2) profile query distributions to quantify what fraction of traffic can be served by smaller models, and (3) evaluate routing solutions, which offer the potential for 60–98% cost reduction [24], [25].

### 8.2 For AI Infrastructure Engineering

We recommend that infrastructure teams instrument systems with hardware telemetry collection alongside performance metrics, experiment with cascade architectures beginning with simple routing rules and measuring impact, and contribute to open-source routing implementations to accelerate community progress.

### 8.3 For Research

The primary research gaps we identify are: (1) integrating speculative decoding with cascade routing in unified architectures, (2) grounding abstract complexity scores in hardware calibration, and (3) developing standardized benchmarks for evaluating routing systems. Our companion paper on Metabolic Cascade Inference [40] addresses these gaps directly.

### 8.4 For Policy

We recommend that policymakers consider requiring energy transparency in AI systems, incentivizing demonstrable efficiency improvements, and funding research into sustainable AI infrastructure.

---

## 9. Scope and Future Work

### 9.1 What This Whitepaper Establishes

This precursor whitepaper documents:

- **The problem**: Quantified evidence of the AI inference cost crisis
- **The root causes**: Technical analysis of memory, throughput, and efficiency bottlenecks
- **The gap**: Why existing solutions (compression, speculation, cascade routing) remain insufficient when used in isolation
- **The direction**: Hardware-aware adaptive routing as the integration paradigm

### 9.2 What This Whitepaper Does Not Claim

This document provides **diagnosis and direction**, not a complete solution. We do not claim:

- Production-ready implementation (forthcoming)
- Large-scale benchmark validation (forthcoming)

- Deployment case studies (forthcoming)

### 9.3 Companion Technical Paper

A companion paper (*Metabolic Cascade Inference* [40]) presents:

- **Complete system architecture** integrating cascade routing, speculative decoding, hardware telemetry, and fact validation
- **Comprehensive benchmarks** across 1,000+ queries on diverse workloads (MT-Bench, AlpacaEval, HumanEval, MATH, TruthfulQA)
- **Ablation studies** quantifying the contribution of each component
- **Production deployment case studies** with measured energy savings
- **Open-source implementation** (Apache 2.0 or MIT license)

The current paper provides the motivation and context; the companion paper provides the solution and validation.

---

## 10. Conclusion

We have presented a comprehensive analysis of the AI inference cost crisis, demonstrating that the problem is structural rather than cyclical. Token prices will continue to decline, but aggregate expenditure will continue to rise as agentic AI, extended context, and continuous deployment multiply consumption faster than unit costs fall.

The three fundamental bottlenecks we identify—KV cache memory pressure, memory bandwidth limitations, and throughput-latency tradeoffs—cannot be resolved through hardware scaling alone. The memory-bound nature of autoregressive decoding imposes a ceiling on what faster processors can achieve. Algorithmic and architectural innovations are required.

We argue that hardware-aware adaptive routing—grounding model selection decisions in physical reality through thermal monitoring, power measurement, and memory state tracking—represents the necessary next step in AI systems optimization. The component technologies (cascade routing [24], [25],

speculative decoding [21], [23], hardware telemetry APIs, and query complexity classification [27]) are individually mature; the remaining challenge is integration into a unified architecture.

By 2027, inference is projected to represent 70–80% of AI compute and 30–40% of total data center demand. Our companion paper on Metabolic Cascade Inference [40] presents a concrete architecture that addresses the integration challenge identified here, combining cascade routing, speculative decoding, hardware telemetry, and fact validation into a unified system achieving 70–75% energy savings while maintaining output quality.

---

## Appendix A: Fact-Checking Summary

The following table summarizes the verification status of key statistics cited in this paper, including corrected figures where original estimates required adjustment.

| Claim | Status | Verified Figure | Sources |
|---|---|---|---|
| Inference >55% of AI infrastructure costs (Q1 2026) | Verified | 55% in 2026, projected 65%+ by 2029 | [1], [3] |
| Inference = 80–90% of model lifecycle compute | Verified | 80–90% | [4], [5] |
| Token prices dropped ~280x | Adjusted | 62x–1000x depending on performance tier | [6], [7] |
| Agentic AI 10–50x more tokens | Verified | 10–50x (up to 100x in some cases) | [11], [12] |
| Data centers ~1,050 TWh by 2026 | Verified | 650–1,050 TWh range | [29], [30] |
| AI e-waste 1.2–5.0 million tons (2020–2030) | Verified | 1.2–5.0 Mt cumulative | [31], [32] |
| DART speedup 2.8–3.4x | Verified | 2.03x–3.44x (surpasses EAGLE3 by 30%) | [21] |
| Inference-optimized chip market >$50B (2026) | Verified | >$50 billion | [2] |
| FrugalGPT cost reduction | Adjusted | Up to 98% with FrugalGPT; 60% is RouteLLM figure | [24], [25] |

| Claim | Status | Verified Figure | Sources |
|---|---|---|---|
| DeepSeek-R1: 671B total, 37B active | Verified | 671B total, 37B active per forward pass | [33], [34] |
| KV cache dominates GPU memory | Verified | Can exceed model weights; 60–80% fragmentation waste | [14], [17] |
| LLM inference is memory-bound | Verified | Decode phase arithmetic intensity ~1–10 vs. A10 threshold of 208 | [18], [19] |

# Appendix B: Terminology

| Term | Definition |
|---|---|
| **KV Cache** | Key-Value cache storing previously computed token embeddings during autoregressive generation; must reside in fast GPU memory for acceptable latency |
| **Memory-bound** | A workload limited by memory bandwidth rather than compute capacity; GPUs wait for data transfer rather than performing calculations |
| **Arithmetic intensity** | Ratio of compute operations (FLOPs) per byte transferred; indicates whether a workload is compute-bound or memory-bound |
| **Batch size** | Number of concurrent requests processed simultaneously; higher batch sizes improve throughput but increase latency |
| **Cascade routing** | Forwarding simple requests to small models and escalating complex requests to larger models based on query characteristics |
| **Speculative decoding** | Using a fast draft model to propose multiple tokens, verified in parallel by a larger target model; reduces latency without quality loss |
| **Thermal headroom** | Distance between current GPU/CPU temperature and the throttling threshold; indicates available capacity for compute-intensive operations |
| **Prefill phase** | Initial processing of the input prompt; compute-bound due to parallel GEMM operations |
| **Decode phase** | Token-by-token generation after prefill; memory-bound due to sequential KV cache access |
| **MoE (Mixture of Experts)** | Architecture where only a subset of model parameters activate per token, reducing compute while maintaining capacity |

## References


[1] Deloitte. “The AI Infrastructure Reckoning: Optimizing Compute Strategy in the Age of Inference Economics.” *Tech Trends 2026*, January 2026. https://www.deloitte.com/us/en/insights/topics/technology-management/tech-trends/2026/ai-infrastructure-compute-strategy.html

[2] Deloitte. “Why AI’s Next Phase Will Likely Demand More Computational Power, Not Less.” *TMT Predictions 2026*, January 2026. https://www.deloitte.com/us/en/insights/industry/technology/technology-media-and-telecom-predictions/2026/compute-power-ai.html

[3] ByteIota. “AI Inference Costs: 55% of Cloud Spending in 2026.” January 2026. https://byteiota.com/ai-inference-costs-55-of-cloud-spending-in-2026/

[4] SambaNova. “AI Is No Longer About Training Bigger Models — It’s About Inference at Scale.” 2025. https://sambanova.ai/blog/ai-is-no-longer-about-training-bigger-models-its-about-inference-at-scale

[5] NVIDIA. “How the Economics of Inference Can Maximize AI Value.” November 2025. https://blogs.nvidia.com/blog/ai-inference-economics/

[6] Epoch AI. “LLM Inference Prices Have Fallen Rapidly but Unequally Across Tasks.” December 2025. https://epoch.ai/data-insights/llm-inference-price-trends

[7] Andreessen Horowitz. “Welcome to LLMflation — LLM Inference Cost Is Going Down Fast.” 2024. https://a16z.com/llmflation-llm-inference-cost/

[8] TechCrunch. “VCs Predict Enterprises Will Spend More on AI in 2026 — Through Fewer Vendors.” December 2025. https://techcrunch.com/2025/12/30/vcs-predict-enterprises-will-spend-more-on-ai-in-2026-through-fewer-vendors/

[9] CIO.com. “AI Cost Overruns Are Adding Up — With Major Implications for CIOs.” 2025. https://www.cio.com/article/4064319/ai-cost-overruns-are-adding-up-with-major-implications-for-cios.html

[10] Panorad AI. “AI Spend Analysis: How Leading Companies Cut AI Costs by 40% in 2025.” 2025. https://panorad.ai/blog/ai-spend-analysis-optimization-2025/

[11] Galileo AI. “The Hidden Costs of Agentic AI: Why 40% of Projects Fail Before Production.” 2025. https://galileo.ai/blog/hidden-cost-of-agentic-ai

[12] OpenReview. “How Do Coding Agents Spend Your Money? Analyzing and Predicting Token Consumptions in Agentic Coding Tasks.” 2025. https://openreview.net/forum?id=1bUeVB3fov

[13] Adaline Labs. “Token Burnout: Why AI Costs Are Climbing and How Product Leaders Can Prototype Smarter.” 2025. https://labs.adaline.ai/p/token-burnout-why-ai-costs-are-climbing

[14] NVIDIA. “Accelerate Large-Scale LLM Inference and KV Cache Offload with CPU-GPU Memory Sharing.” January 2025. https://developer.nvidia.com/blog/accelerate-large-scale-llm-inference-and-kv-cache-offload-with-cpu-gpu-memory-sharing/

[15] NVIDIA. “How to Reduce KV Cache Bottlenecks with NVIDIA Dynamo.” January 2025. https://developer.nvidia.com/blog/how-to-reduce-kv-cache-bottlenecks-with-nvidia-dynamo/

[16] Introl. “KV Cache Optimization: Memory Efficiency for Production LLMs.” 2025. https://introl.com/blog/kv-cache-optimization-memory-efficiency-production-llms-guide

[17] arXiv. “Understanding Bottlenecks for Efficiently Serving LLM Inference With KV Offloading.” arXiv:2601.19910, January 2025. https://arxiv.org/abs/2601.19910

[18] arXiv. “Mind the Memory Gap: Unveiling GPU Bottlenecks in Large-Batch LLM Inference.” arXiv:2503.08311, 2025. https://arxiv.org/abs/2503.08311

[19] Baseten. “A Guide to LLM Inference and Performance.” 2024. https://www.baseten.co/blog/llm-transformer-inference-guide/

[20] arXiv. “A Systematic Characterization of LLM Inference on GPUs.” arXiv:2512.01644, December 2024. https://arxiv.org/abs/2512.01644

[21] arXiv. “DART: Diffusion-Inspired Speculative Decoding for Fast LLM Inference.” arXiv:2601.19278, January 2025. https://arxiv.org/abs/2601.19278

[22] BentoML. “Get 3x Faster LLM Inference with Speculative Decoding Using the Right Draft Model.” 2024. https://www.bentoml.com/blog/3x-faster-llm-inference-with-speculative-decoding

[23] Google Research. “Looking Back at Speculative Decoding.” 2024. https://research.google/blog/looking-back-at-speculative-decoding/

[24] Chen, L., Zaharia, M., Zou, J. “FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance.” *arXiv:2305.05176*, May 2023. https://arxiv.org/abs/2305.05176

[25] LMSYS Org. “RouteLLM: An Open-Source Framework for Cost-Effective LLM Routing.” July 2024. https://lmsys.org/blog/2024-07-01-routellm/

[26] Google Research. “Speculative Cascades — A Hybrid Approach for Smarter, Faster LLM Inference.” 2024. https://research.google/blog/speculative-cascades-a-hybrid-approach-for-smarter-faster-llm-inference/

[27] NVIDIA. “Prompt Task and Complexity Classifier.” *HuggingFace*, 2024. https://huggingface.co/nvidia/prompt-task-and-complexity-classifier

[28] NVIDIA. “Deploying the NVIDIA AI Blueprint for Cost-Efficient LLM Routing.” 2025. https://developer.nvidia.com/blog/deploying-the-nvidia-ai-blueprint-for-cost-efficient-llm-routing/

[29] IEA. “Energy Demand from AI.” *Energy and AI Report*, 2024. https://www.iea.org/reports/energy-and-ai/energy-demand-from-ai

[30] Carbon Brief. “AI: Five Charts That Put Data-Centre Energy Use and Emissions into Context.” 2024. https://www.carbonbrief.org/ai-five-charts-that-put-data-centre-energy-use-and-emissions-into-context/

[31] Wang, P. et al. “E-Waste Challenges of Generative Artificial Intelligence.” *Nature Computational Science*, October 2024. https://www.nature.com/articles/s43588-024-00712-6

[32] Scientific American. “Generative AI Could Generate Millions More Tons of E-Waste by 2030.” October 2024. https://www.scientificamerican.com/article/generative-ai-could-generate-millions-more-tons-of-e-waste-by-2030/

[33] DeepSeek. “DeepSeek-R1.” *GitHub*, January 2025. https://github.com/deepseek-ai/DeepSeek-R1

[34] BentoML. “The Complete Guide to DeepSeek Models: V3, R1, V3.1, V3.2 and Beyond.” January 2025. https://www.bentoml.com/blog/the-complete-guide-to-deepseek-models-from-v3-to-r1-and-beyond

[35] Botpress. “Ultimate Guide to FAQ Chatbots (2026).” 2026. https://botpress.com/blog/faq-chatbot

[36] Fullview. “200+ AI Statistics and Trends for 2025: The Ultimate Roundup.” 2025. https://www.fullview.io/blog/ai-statistics

[37] Charlot, D.J. “Bounded Entropy Code Generation: A Deterministic Framework for Reliable AI-Assisted Software Development.” OpenIE Technical Report, December 2025.

[38] Charlot, D.J. “Deterministic Code Auditing: Formal Verification of AI-Generated Software.” OpenIE Technical Report, December 2025.

[39] Charlot, D.J. “Cortex: A Neural-Symbolic Programming Language for Verifiable AI Integration.” OpenIE Technical Report, January 2026.

[40] Charlot, D.J. “Metabolic Cascade Inference: Hardware-Aware Adaptive Routing for Energy-Efficient AI.” OpenIE Technical Report, January 2026.

[41] Kwon, W. et al. “Efficient Memory Management for Large Language Model Serving with PagedAttention.” *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023. https://arxiv.org/abs/2309.06180

---

*This whitepaper describes independent academic research focused on sustainable AI inference and hardware-aware adaptive routing. Published freely without patent protection under CC BY 4.0 license.*

**Contact**: david@openie.dev | dcharlot@ucsb.edu **Latest Updates**: https://openie.dev/projects/precursor