

We Gave the AI a 1M-Token Window. It Got **Worse**.

Response times jumped from 2s to 15s+, costs tripled, and the answers got worse, not better. Here's why bigger context isn't automatically better.

SWIPE →

What Actually Happens at Scale

01 The "lost in the middle" problem

02 Quadratic attention scaling

03 Hallucination risk under information overload

04 The fix: retrieval, not brute force

35%

Customer Satisfaction Gain From Extended Memory

Support operations that hold full interaction history and policy documents in one context report 35% satisfaction gains and 50% faster resolution.

Datapro.news, 2025

24%

Latency Increase From 6GB More Memory

Grow a model's memory footprint from 20GB to 26GB and throughput drops from ~50 to ~38 tokens per second. GPU memory bandwidth, not compute, becomes the bottleneck.

LocalLLaMA, 2025

47%

After switching to RAG with intelligent chunking, context caching, and routing simple queries to smaller models: response times back under 3 seconds, costs down 47% from baseline.

Our own redesign, post-1M-token experiment



Models reliably retrieve beginning and end-positioned tokens but demonstrate degraded performance for central content.

Liu et al., 2024

Transactions of the Association for Computational Linguistics

The 1M-Token Attempt vs. the Redesign

MAX CONTEXT, BRUTE FORCE

15+ second responses

Costs tripled

Answers got worse

RAG + ROUTING + CACHING

Under 3 second responses

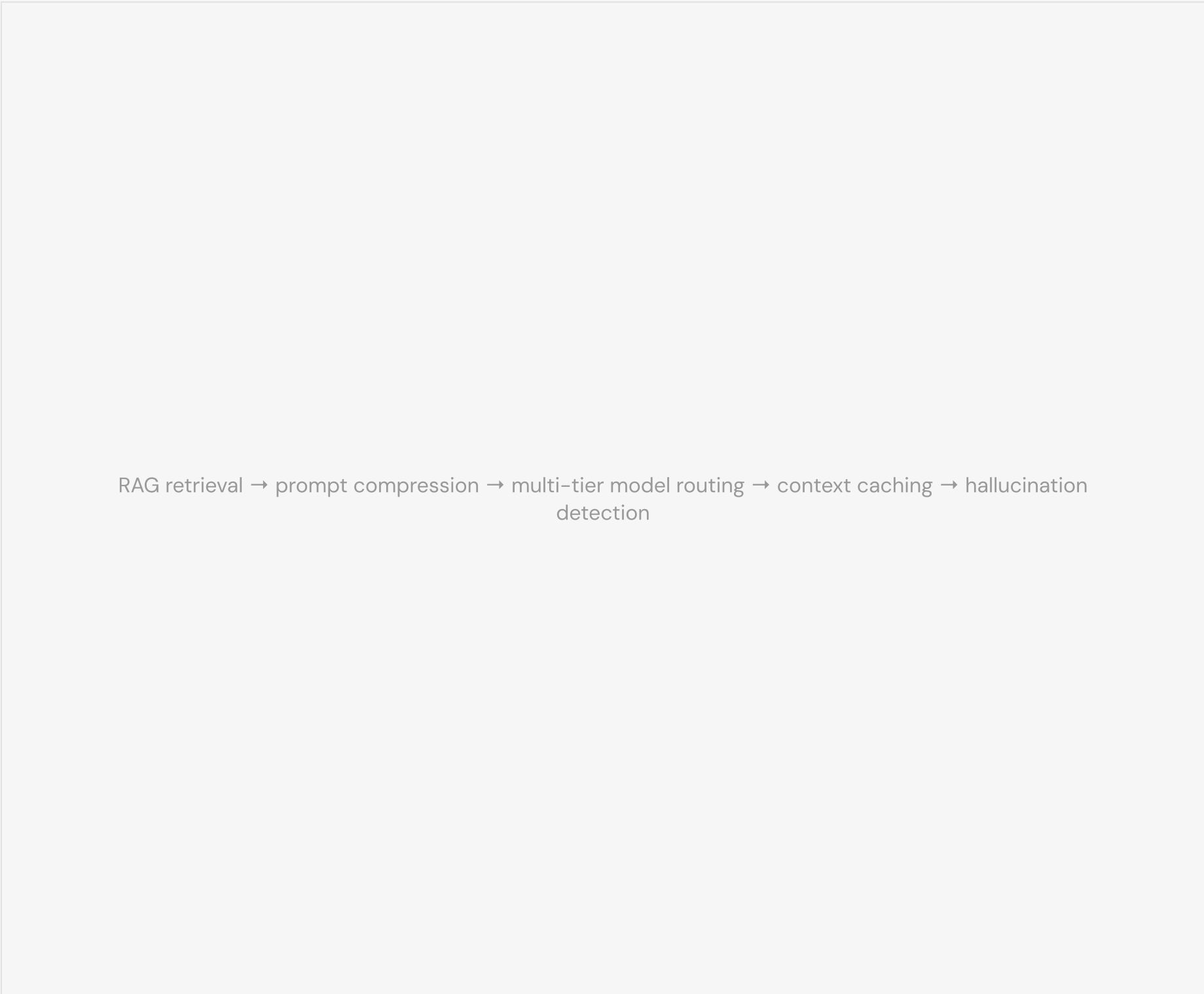
Costs down 47%

Grounded, fact-checked answers

A Bigger Desk Doesn't Make You **Faster**.

A context window is like desk space: a bigger desk lets you spread out more documents, but it doesn't make you think faster. Worse, the computational cost of comparing everything on that desk to everything else grows quadratically, not linearly. Double the context, and the work more than doubles.

Right-Size, Don't Maximise



Full breakdown, mitigation framework, and sources:
terencekok.com/blog/context-window-why-bigger-isnt-always-better

Give your AI exactly what it needs. **Nothing more.**

Context window engineering isn't glamorous, but it's the difference between a system that drains your budget and one that delivers.



Terence Kok · Enterprise AI Strategist

SAVE · COMMENT · FOLLOW