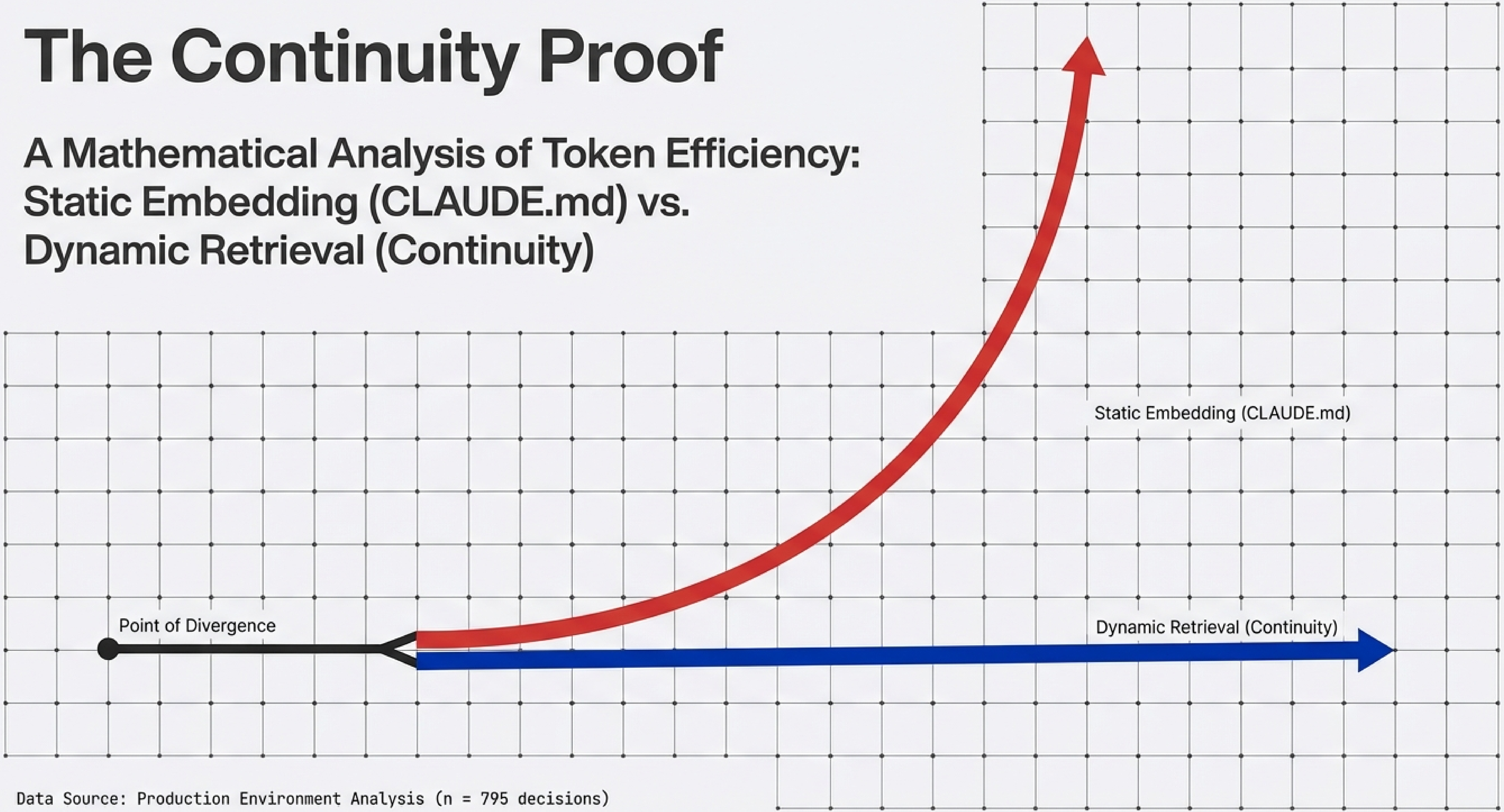
The Continuity Proof

A Mathematical Analysis of Token Efficiency: Static Embedding (CLAUDE.md) vs. Dynamic Retrieval (Continuity)



Data Source: Production Environment Analysis (n = 795 decisions)

Executive Summary: The Q.E.D.

Continuity is not merely an alternative; it is the mathematically superior architecture for scale.

11.66x

Efficiency Multiplier

Observed at current scale (n=795)

91.4%

Token Savings

Reduction in operational load per session

O(1)

Constant Complexity

Infinite scalability without context limits

Defining the Problem Space

To evaluate the efficiency of memory systems, we analyze the following variables based on real-world project data.

Variable Legend		
Symbol	**Definition**	**Value**
n	Total number of decisions	795
d	Average tokens per decision	286
b	Base instructions	7,053 tokens
q	Queries per session	3
r	Results per query	15
C	Context window limit	200,000 tokens

Current Project Scale

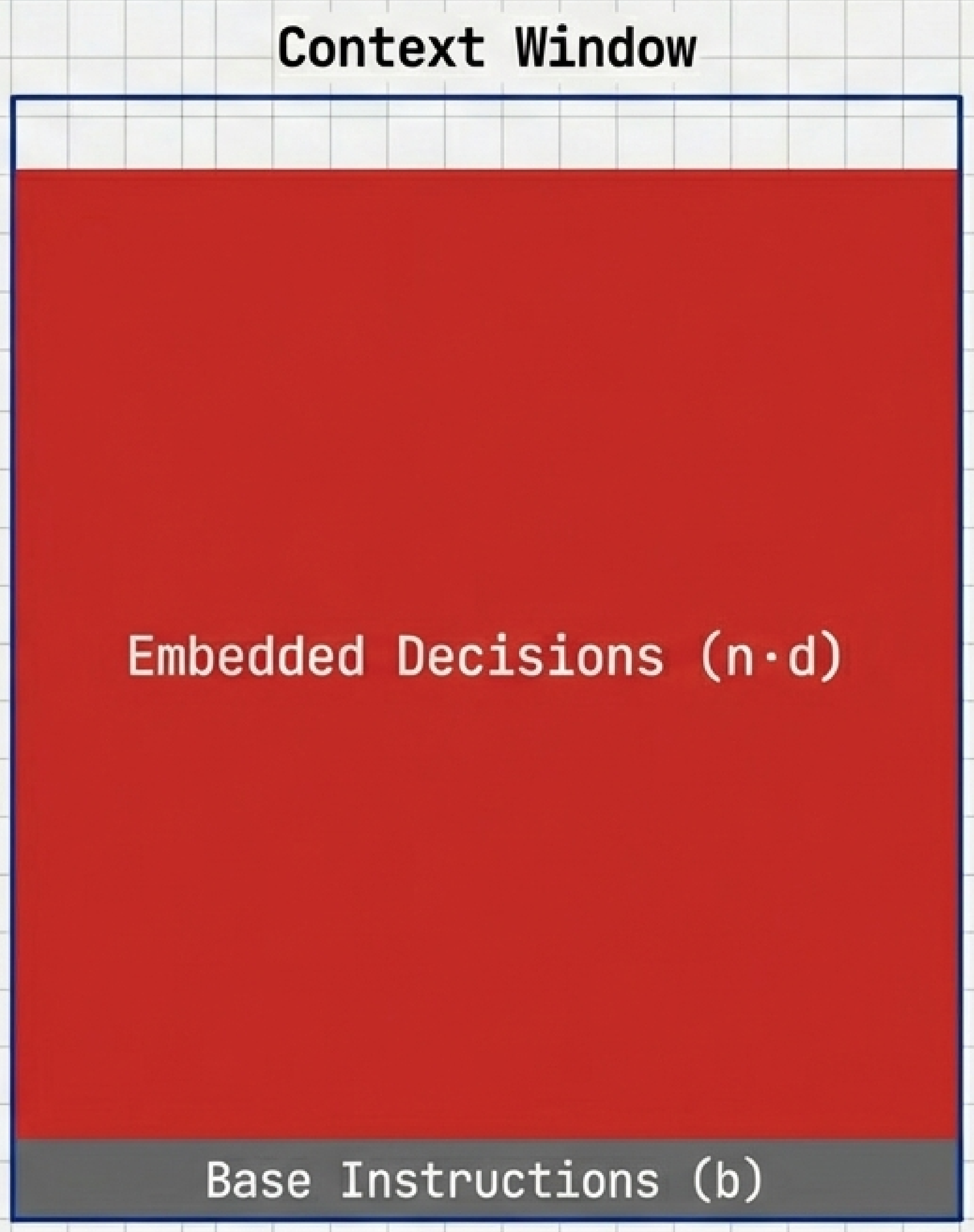
Method A: The Static Burden

Linear Growth (CLAUDE.md)

$$T(n) = b + n \cdot d$$

Characteristics:

- Linear Growth O(n)
- Indiscriminate Loading: 100% of history loaded regardless of relevance.
- Weight increases with every new decision.



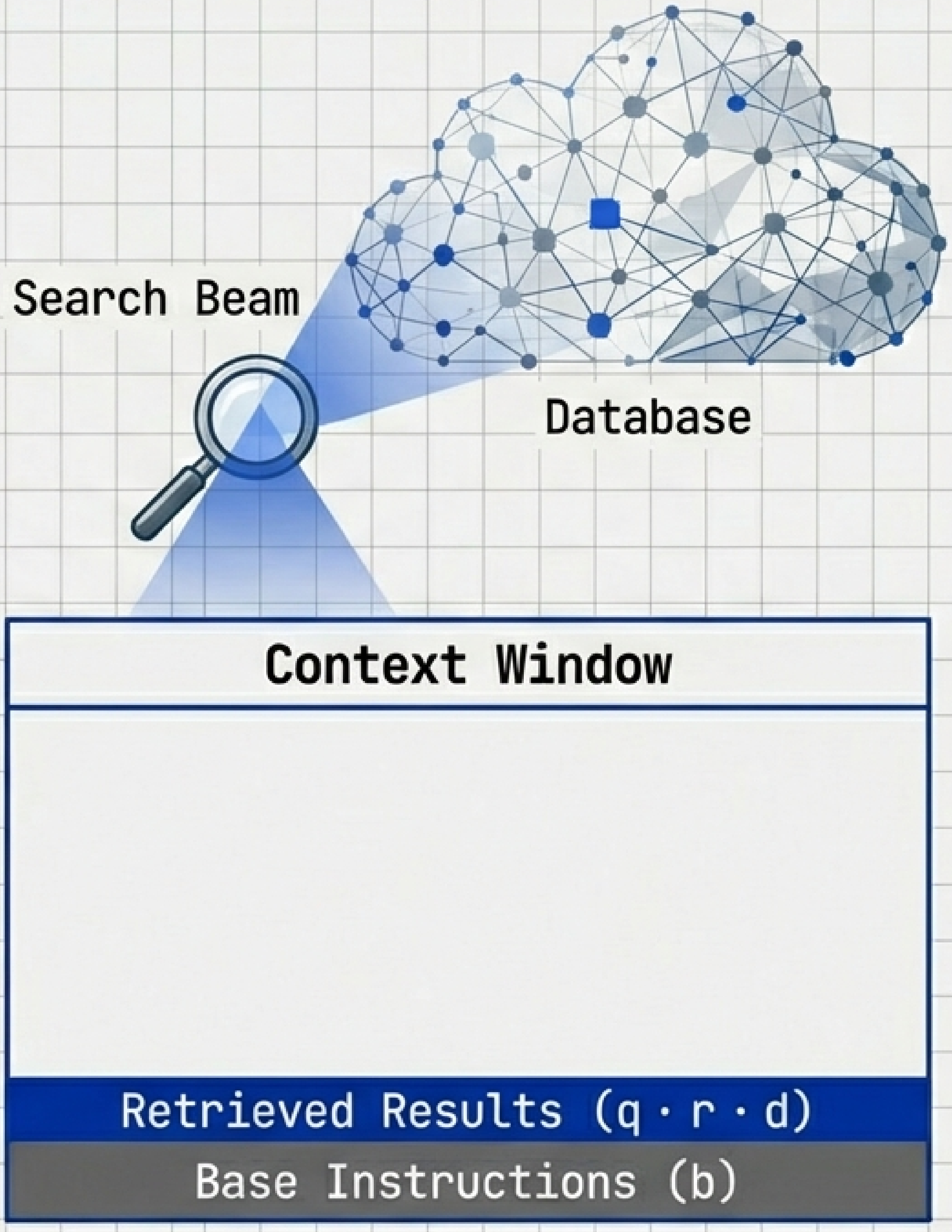
Method B: The Dynamic Search

Constant Efficiency (Continuity)

$$T(q) = b + q \cdot r \cdot d$$

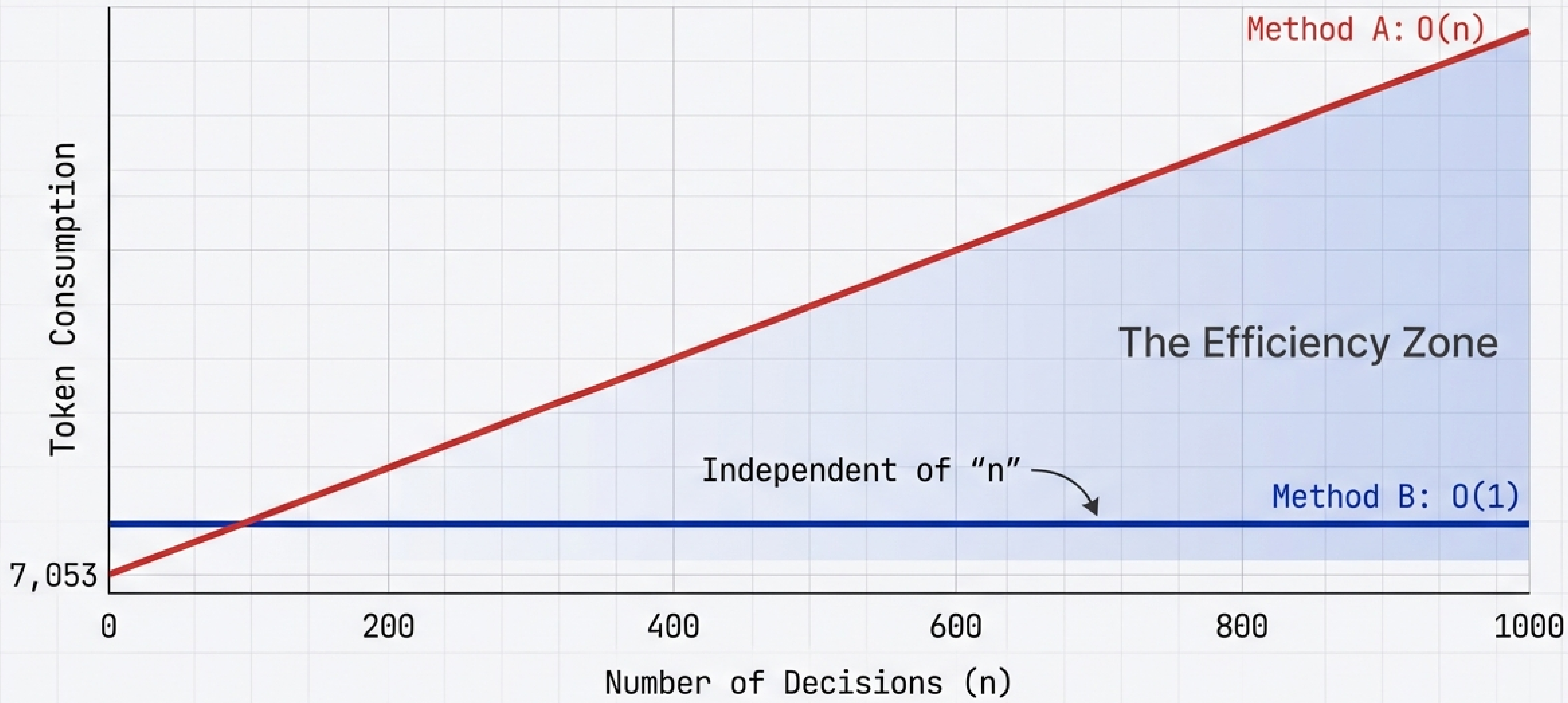
Characteristics:

- Constant Complexity O(1)
- Targeted Retrieval: Only relevant results loaded.
- Token consumption remains flat as 'n' grows.



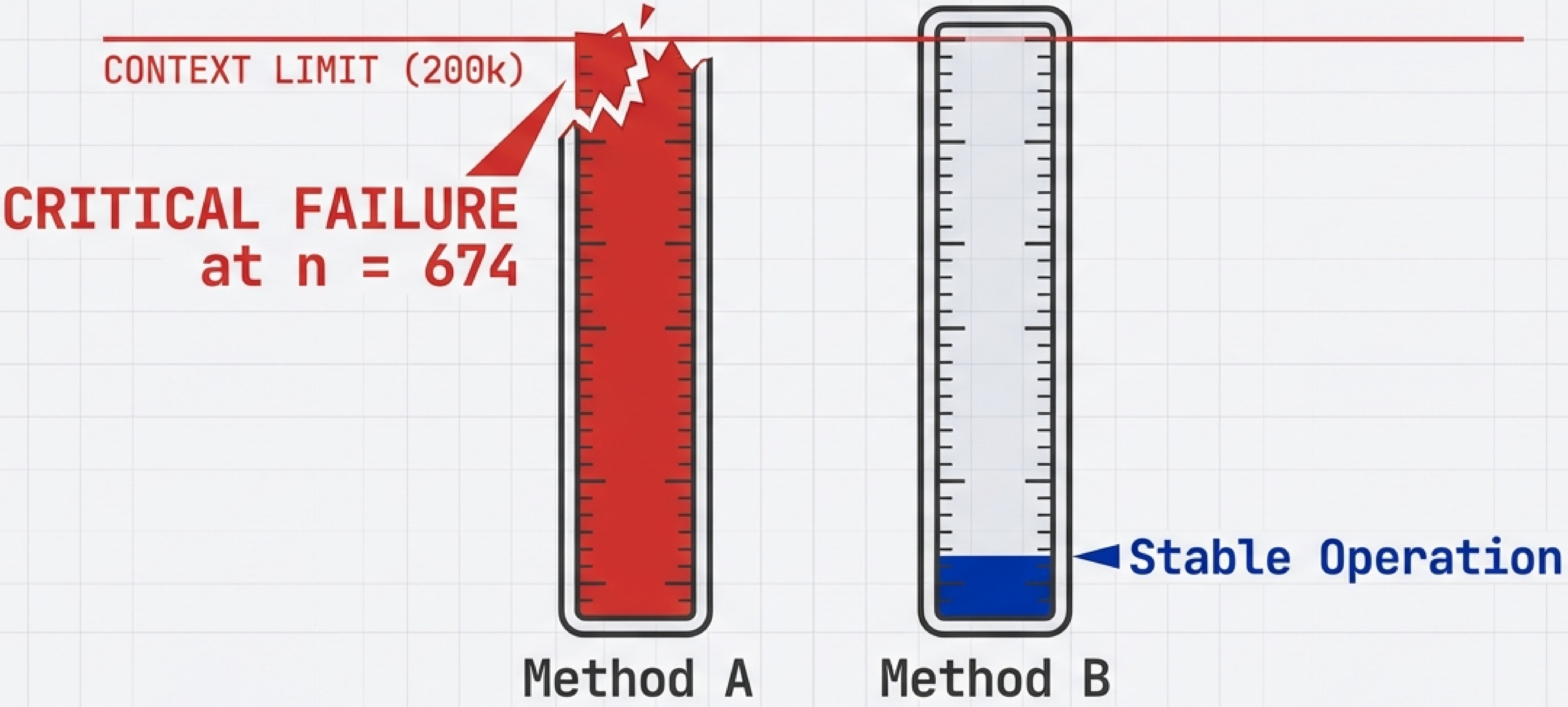
The Divergence: Complexity Analysis

Linear vs. Constant Growth



NotebookLM

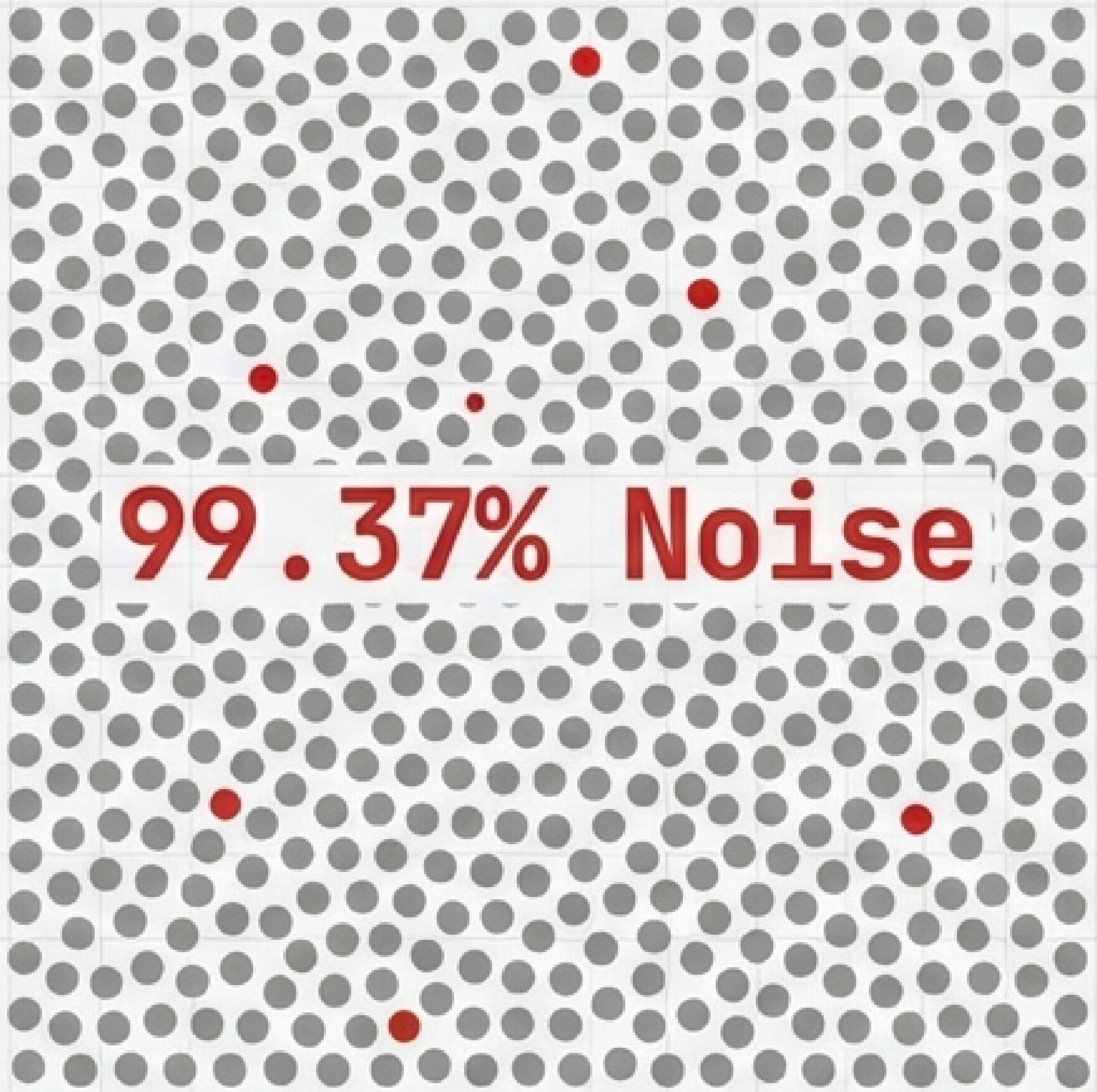
The Context Window Constraint



Beyond 674 decisions, the static approach becomes mathematically impossible. Continuity scales infinitely within the same limit.

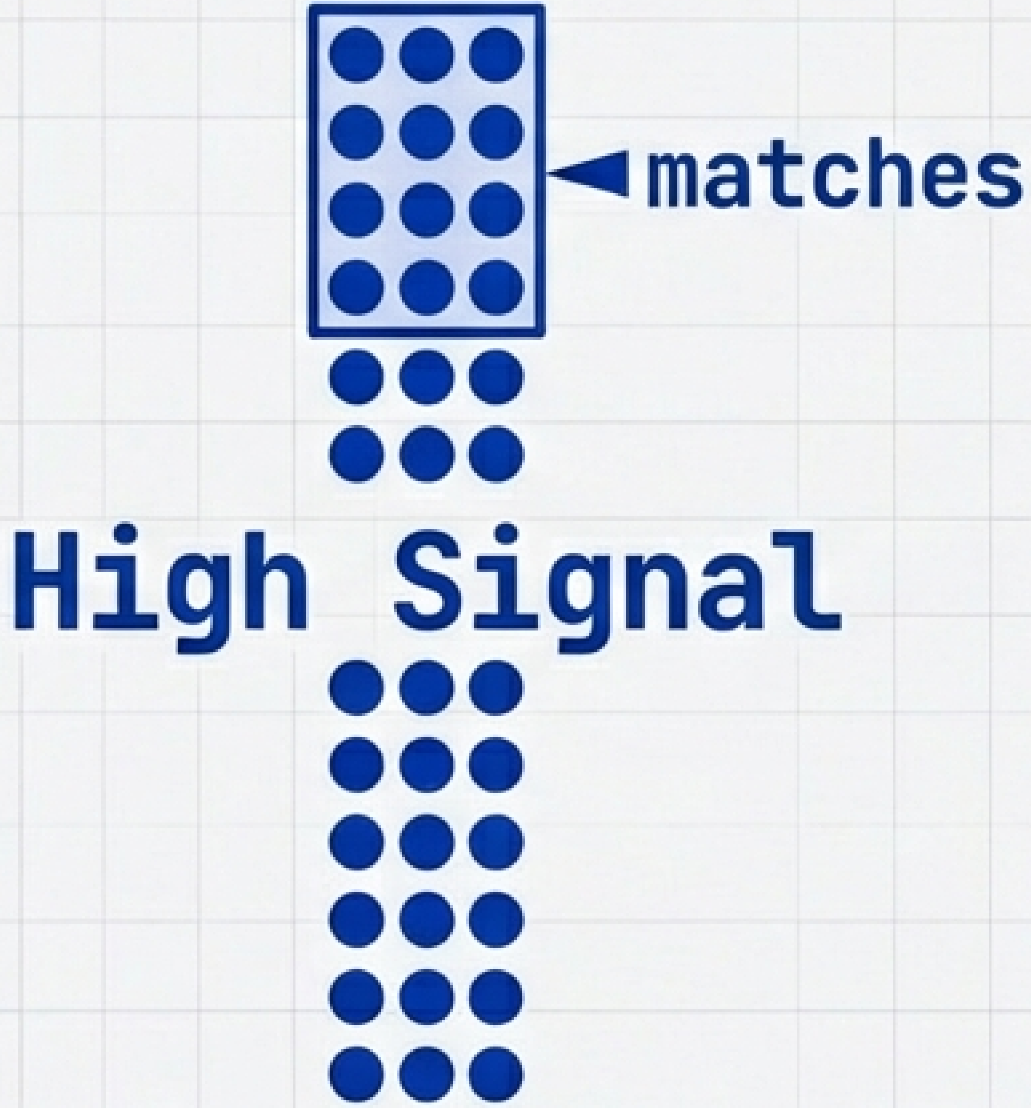
Signal-to-Noise Ratio

Method A



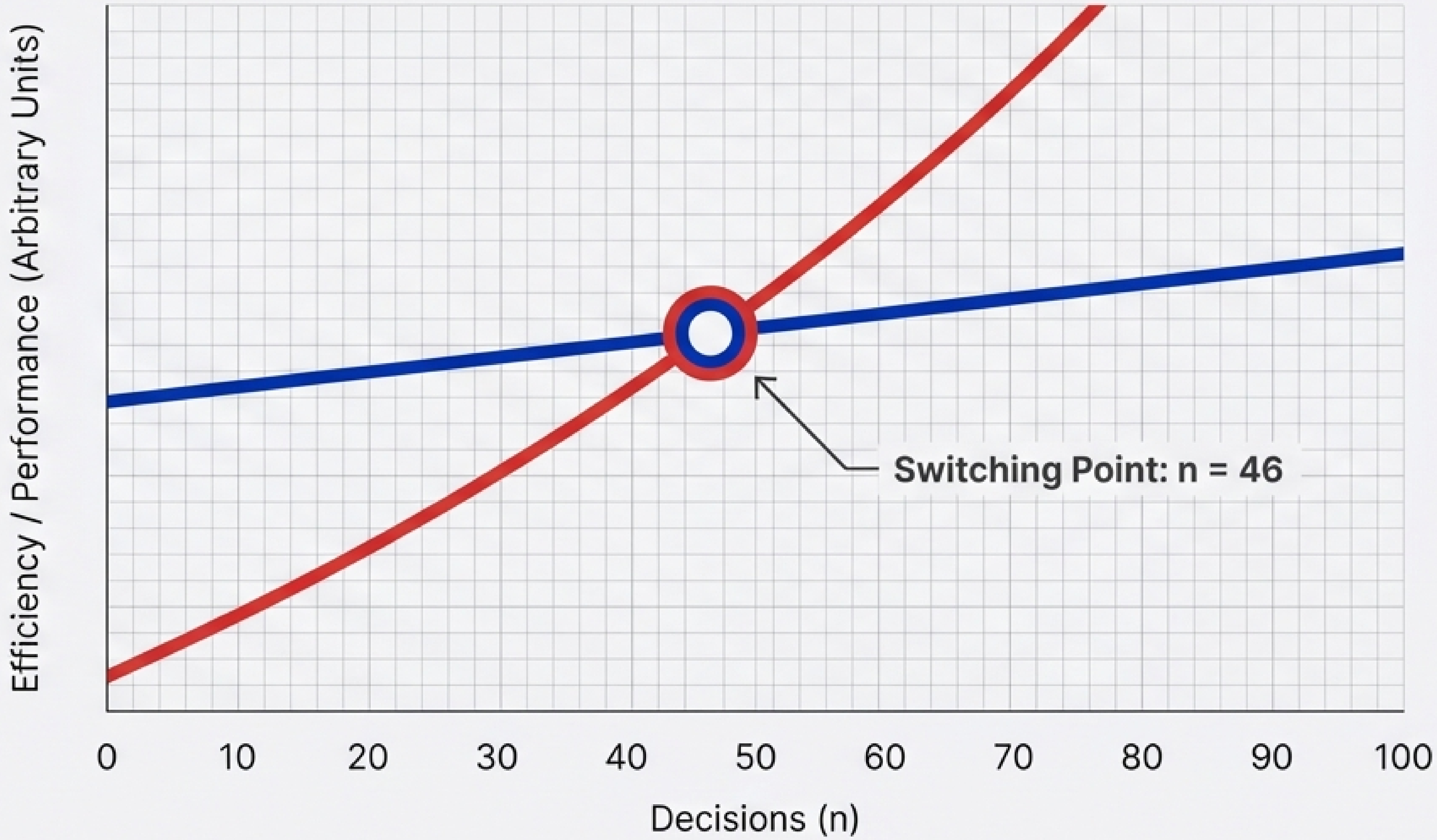
Static Embedding: Loading irrelevant history.

Method B



Dynamic Search: 52.9x Relevance Improvement.

Break-Even Analysis



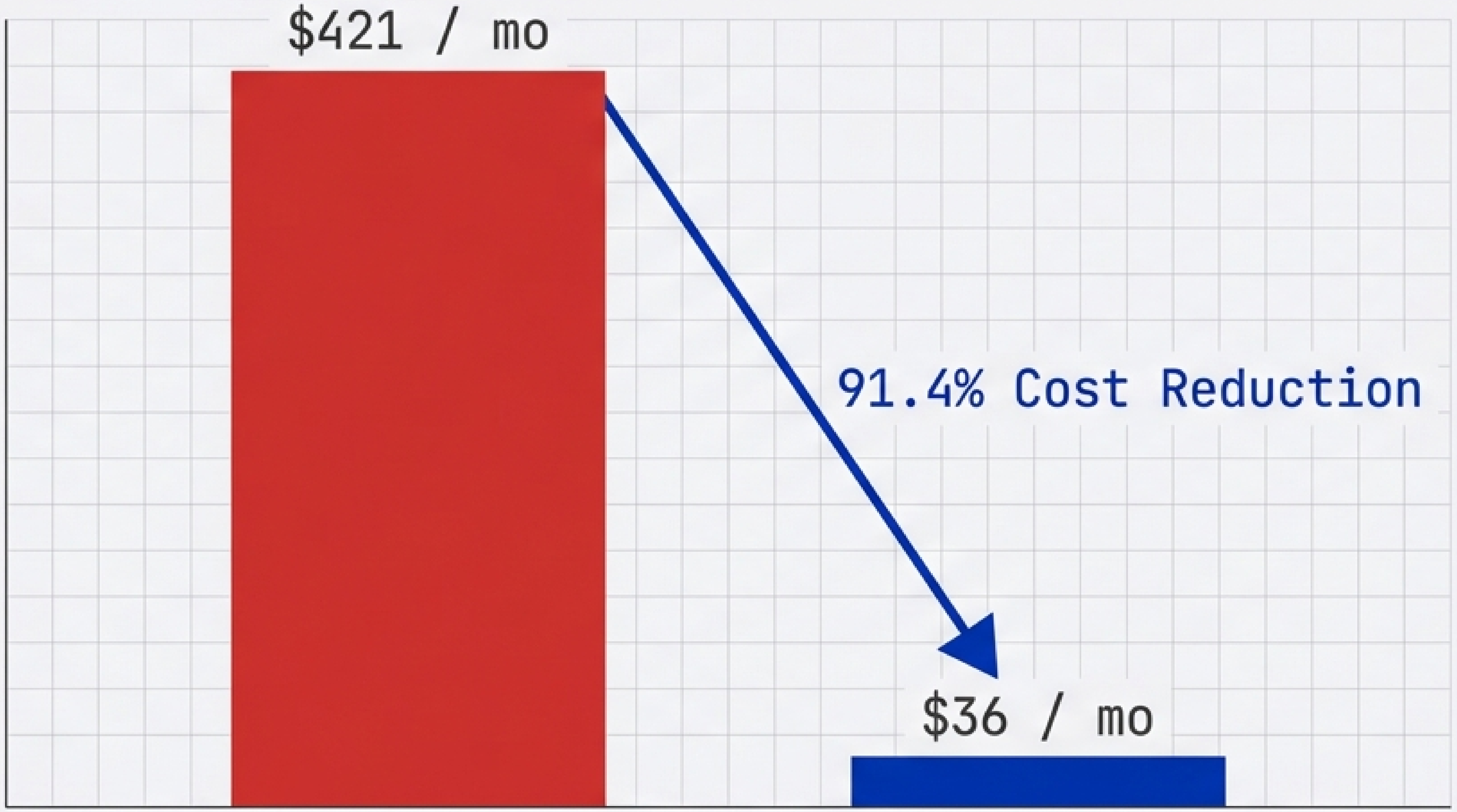
If n < 46:
Method A is sufficient.

If n > 46:
Continuity is mathematically superior.

Current Scale: n = 795
(Far past break-even)

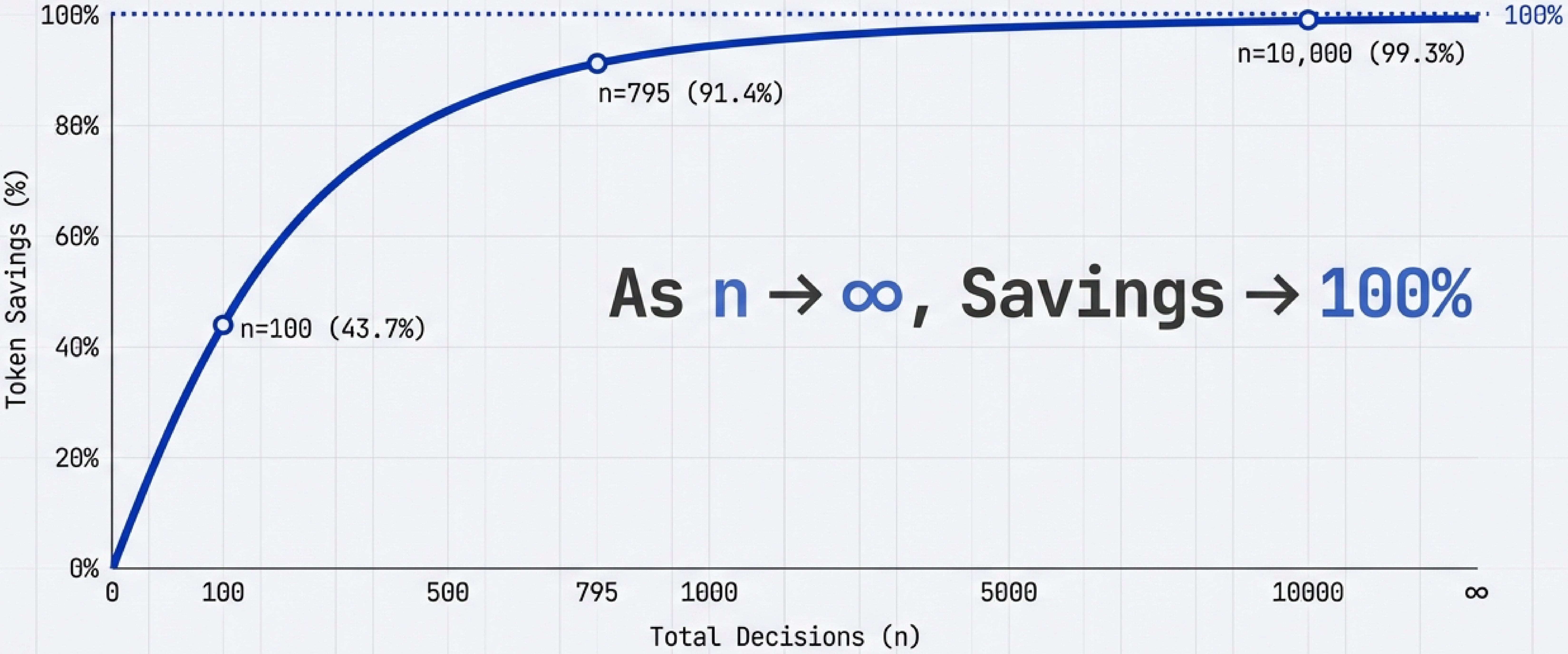
Financial Impact at Scale

Monthly Burn Rate (USD)



Yearly savings per project: $4,623 (Based on 600 sessions/mo)

Infinite Scalability



NotebookLM

Summary of Results

Metric	CLAUDE.md (Static)	Continuity (Dynamic)
Complexity	O(n) [Linear]	**O(1) [Constant]**
Max Decisions	674 (Crash Point)	**∞ (Unlimited)**
Tokens (n=795)	234,123	**20,073**
Efficiency	1x	**11.66x**
Monthly Cost	$421	**$36**

Theorem: Optimal Memory Architecture

Theorem: For any decision memory system with n decisions, search-based retrieval is asymptotically optimal.

Proof:

Let $T*(n)$ be optimal token consumption.

Upper bound embedding: $T\_A = O(n \cdot d)$

Upper bound search: $T\_B = O(1)$

Since $O(1) < O(n)$ for large n:

∴ Search-based retrieval approaches optimal efficiency as n increases. ■

Q.E.D.

The Fundamental Equation

As decisions grow, Continuity's relative cost approaches zero.

- ☑ O(1) Scalability
- ☑ >90% Savings
- ☑ Infinite Context