How to reduce AI costs by 50-70% with data curation

Don’t change your models — just what you send them

Enterprise AI projects are expensive. But most teams are paying for inefficient data.

Often the first instinct when facing high AI costs is to optimize model choice, infrastructure or prompts. Those things matter. But in building Deasy and working with enterprise teams across industries, we kept seeing the same problem upstream: organizations were sending the wrong data to their models, and paying for it at scale.

Every token sent to a model costs money. If your retrieval pipeline is pulling irrelevant documents, outdated files or entire contracts when you only needed one clause, you’re paying for noise, and at enterprise scale.

You’re not overspending on AI. You’re overspending on what you send to it.

The hidden cost model of enterprise AI

To understand where the spend actually comes from, consider a realistic deployment:

Baseline example: customer service AI assistant

- Corpus size: 100,000 documents
- Daily query volume: ~2,000 queries/day
- Avg tokens per query: ~10,000 (input + output)
- Model cost (blended): $3–$10 per 1M tokens
- Estimated annual cost: $20K–$70K per use case

Twenty to seventy thousand dollars per use case, per year. That seems manageable, until you account for what actually happens in production.

The multiplier most team miss: Retries

Enterprise AI systems rarely return the right answer the first time. Users rephrase queries. Agents fall back to secondary chains. Retrieval pulls incomplete or irrelevant context, triggering automatic retries.

In practice, 30–50% of queries trigger at least one retry. This creates a compounding cost effect that most budgets don't account for:

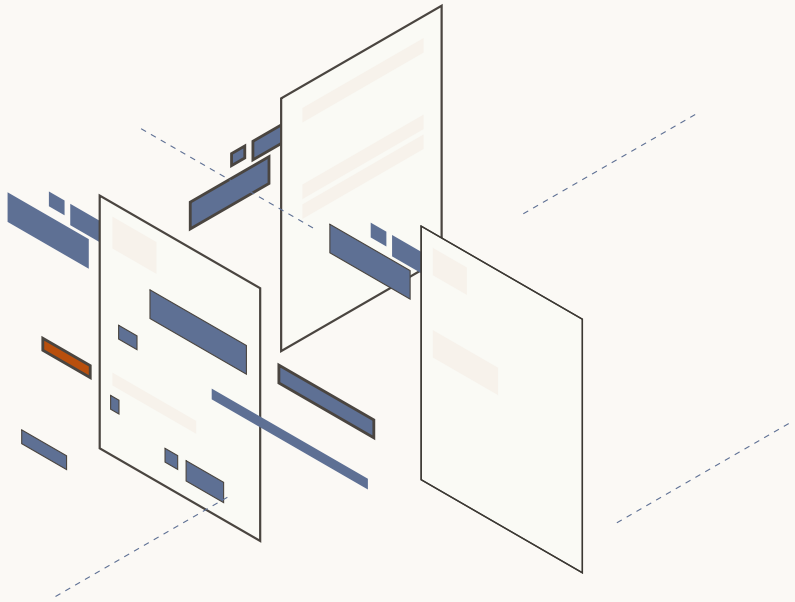
Scenario	Cost multiplier	Effective annual cost
No retries	1.0×	$20K–$70K
Moderate retries (30%)	1.5×	$30K–$105K
Heavy retries (50%)	2.0–3.0×	$40K–$150K+

For enterprises running dozens of AI applications, this quickly compounds into a multi-million dollar inefficiency — without anyone noticing the leak.

The conventional response is to look at scale: too many documents, too large a vector database. But that's the wrong diagnosis.

Storage is cheap. Retrieval mistakes are expensive.

Vector DB storage typically runs $1K–$10K per year. LLM usage runs 10–100× higher. Reducing your vector store saves thousands. Improving retrieval *saves hundreds of thousands*.



The three levers that actually reduce cost

Across every enterprise deployment we've seen, cost reduction comes from the same three places. Each works independently. Together, they compound.

Lever 1 — Curate what enters the pipeline

The highest-leverage change most teams can make is stopping irrelevant files from entering the retrieval pipeline at all.

Most RAG pipelines treat the data corpus as static: if it exists, it's retrievable. But enterprise data is never clean. It contains outdated policies, duplicate documents, sensitive files and content that has nothing to do with the AI use case at hand.

A compounding factor here: many enterprises now run a multimodal LLM to ingest everything upfront, processing documents, images, PDFs, and audio before retrieval even begins. Those models are expensive, and in practice only about 30% of that data is relevant to any given use case. That means the majority of ingestion spend goes toward content that will never surface a useful answer. Curating before ingestion is one of the most direct ways to cut that cost.

Deasy maps and filters your corpus before anything reaches a model — removing outdated and duplicate files, scoping to only the knowledge relevant to each use case, and excluding sensitive or restricted content.

Cost impact

For a typical enterprise RAG platform, filtering ~30% of irrelevant or duplicate data before multimodal ingestion translates to roughly **$75K–$150K/year** in direct savings, and up to **$500K+** when you account for reprocessing and infrastructure overhead.

Key assumptions:

- Scale: ~5M documents/year at ~3K tokens per document (enterprise-wide adoption across teams)
- Model cost: ~$0.006 per 1K tokens using multimodal models like GPT-4o
- Waste + reprocessing: ~30% of documents are irrelevant/duplicate, with 2–3× reprocessing over time (taxonomy changes, new use cases, model upgrades)

Lever 2 — Retrieve at the chunk level, not the document level

Even after curation, documents are far larger than what's relevant to any given query. A 40-page contract contains one relevant clause. A 200-page product manual has three paragraphs relevant to the error your team is debugging.

Sending whole documents bloats context windows, increases token usage and degrades model performance. Deasy embeds at the page and chunk level, then uses those embeddings to surface only the sections most relevant to each query — before anything gets sent to a model.

Metric	Without optimization	With chunk-level retrieval
Chunks retrieved per query	~30	~8
Tokens per query	~10,000	~4,000
Reduction in token usage	—	40–70%

You don't need less data. You need less irrelevant data per query.

Lever 3 — Skip the LLM entirely for structured tasks

This is the most underused insight in enterprise AI cost reduction. A large portion of classification tasks don't require a model call at all.

Extracting dates, classifying document types, tagging categories — these tasks are predictable, structured and repeatable. Deasy replaces many of these with contextual keyword matching, regex patterns, and lightweight ML classifiers that match or exceed LLM accuracy at a fraction of the cost.

The key is contextual matching: not just looking for the word 'termination,' but looking for it in proximity to date formats and clause structures that signal the right context. This prevents false positives while keeping the logic fast and cheap.

Cost impact

- 30–60% of classification tasks can bypass LLM calls entirely
- Cost per task reduced by 10–100×
- Model budget freed for reasoning, not pattern matching

What changes with metadata-driven retrieval

The biggest unlock comes from combining all three levers with strong metadata. Instead of searching across an entire corpus, metadata allows systems to filter by document type, restrict to relevant regions or products, exclude outdated or draft content, and narrow to specific timeframes.

Here is what that looks like against the same baseline deployment:

Metric	Without optimization	With Deasy
Chunks retrieved per query	~30	~8
Tokens per query	~10,000	~4,000
Retry rate	40%	15%
Effective tokens per query (incl. retries)	~14,000	~4,600
Cost per query	~$0.10	~$0.03
Annual cost (same use case)	~$70,000	$20,000–$30,000

Net result: 50–70% reduction in LLM spend — with higher answer accuracy.

Each lever improves cost individually. Together, they compound: smaller context means fewer tokens, better retrieval means fewer retries, structured task routing means fewer model calls. The result is not additive, it's exponential.

The big takeaway

The highest leverage is upstream, where you can control what your model sees. Better inputs mean lower costs, better accuracy and systems that scale.

The teams that get ahead will have better metadata, better retrieval discipline and better data curation.

See what this looks like against your actual data environment

Book a 30-minute demo at deasylabs.com

Key assumptions and methodology
Estimates are based on observed enterprise deployments and standard pricing models. Model cost: $3–$10/1M tokens (blended, GPT-4 class). Queries/day: 1,000–5,000 per use case. Tokens/query (baseline): 8K–20K. Retry rate (baseline): 30–50%. Chunk reduction with metadata: 60–80%. Retry reduction with better retrieval: 20–40%. Vector database storage costs are excluded from primary savings calculations due to their comparatively low impact. Cost reductions assume no degradation in answer quality — in practice, accuracy typically improves. Savings scale linearly with query volume and number of AI applications deployed.