RAG, CAG AND OTHER DATA-SCIENCE TOOLS

RAG, CAG and long context: use the simplest thing that works

Retrieval pipelines were essential when models read 4,000 tokens. With windows of millions, many fund use cases no longer need them.

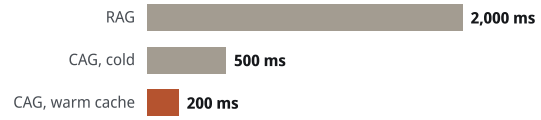
KEY MESSAGE

Use the simplest architecture that works: load and cache documents first, retrieval only for large archives.

3 things to know

1. **Load documents before building retrieval:** in published experiments, long context often outperformed RAG [1].
2. **Caching makes it cheap:** about 10% of the price on repeat calls [3].
3. **Keep RAG for large, changing archives**, and code for numbers.

EXHIBIT 1 **Indicative end-to-end latency, ms (one practitioner's production estimates; varies by setup)**



Production estimates: RAG 600 ms to 2 s; CAG 200–500 ms, under 200 ms warm. Source [3].

4K → 10M
tokens of context, GPT-3.5 to Llama 4 Scout [1]

~10%
price of cached tokens on repeat calls [3]

0.6–2 s
RAG latency in one practitioner's estimates [3]

80/20
common split: CAG for bounded docs, RAG for the long tail [3]

Three ways to give a model your documents

RAG (retrieval-augmented generation) cuts documents into pieces, indexes them and retrieves the most relevant pieces at query time. **Long context** loads the whole set into the prompt.

CAG (cache-augmented generation) loads it and caches it, so repeat questions are fast and cheap [1][3].

When RAG is useful

Very large or changing collections: every past IC memo, years of news, a library of expert-call transcripts, a CRM.

Cases where freshness matters or the corpus far exceeds any context window.

RAG fails when retrieval misses the right passage: the model then answers confidently from the wrong pieces [2].

When it is useless complexity

A data room for one deal, a CIM and its appendices, a sector library of a few hundred pages: load them. Google researchers found long context consistently outperforms RAG when resources allow; a 2024 paper is simply titled “Don't do RAG” [1].

Fewer moving parts means fewer silent errors and less maintenance.

Sources on page 2.

CarveLines © 2026

Other data-science tools

Fine-tuning rarely pays for a fund: methods change faster than training cycles, and skills deliver most of the benefit. Vector databases, knowledge graphs and custom embeddings belong to firms with an engineering team and a large, stable corpus.

For most funds: long context and caching for deal work, managed search for the archive, and spreadsheets checked by code for numbers.

EXHIBIT 2 Which approach for which job (CarveLines view; dots = fit of the simplest option)

JOB	BEST APPROACH	SIMPLICITY
One data room	Long context, cached	●●●
CIM and appendices	Long context	●●●
Sector library	Long context or CAG	●●●
All past IC memos	RAG or managed search	●●○
Years of news	RAG with dates	●○○
Numbers in Excel	Code, not retrieval	●●○

CarveLines view, consistent with [1][3].

Context rot still applies

Big windows are not unlimited attention. Recall falls as context fills (see No. 15). Load what the task needs, not everything available, and put the most important documents first.

When a data room exceeds the window, split by workstream rather than by page: one context for commercial, one for legal, one for financial.

> "So which method should you use? It's dependent on your use case, but I think CAG makes for a good starting point because of its simplicity."
>
> **Dan Cleary**, founder, PromptHub · Apr 2025 [1]

> "CAG greatly reduces generation time by eliminating retrieval overhead"
>
> **Authors of "Don't Do RAG"**, arXiv 2412.15605 · Dec 2024 [1]

5 hints: the average fund vs full power

	THE AVERAGE FUND	FULL POWER
1	Builds a vector database first	**Loads the data room and caches it**
2	Fine-tunes models	**Writes skills**
3	Trusts retrieval silently	**Tests retrieval misses**
4	Loads everything at once	**Splits large rooms by workstream**
5	Uses AI for arithmetic	**Uses code for every number**

Watch out

Building a vector database as the first AI project. It looks like progress, takes months and often answers worse than loading the documents.

Questions for the partners' meeting

1. Does our largest recent data room fit in our assistant's context?
2. Where are we paying for retrieval we do not need?
3. Is prompt caching switched on for repeated work?
4. Which archive would truly benefit from search?

Monday morning

Take your largest recent data room. Check whether it fits in your assistant's context window. If it does, compare answers with and without your retrieval tool on ten questions.

Sources: [1] PromptHub, RAG vs CAG (citing Google Research, arXiv 2407.16833, and arXiv 2412.15605), Apr 2025; [2] arXiv 2501.01880, Long context vs RAG for LLMs: an evaluation and revisits, Jan 2025; [3] RAG vs CAG: what the difference means in production (blog), Sep 2026. Quotes verbatim from the sources cited. Figures as published; survey scopes and dates vary. Vendor-reported figures are labelled; [E] = CarveLines estimate; 'CarveLines view' = our interpretation.

CarveLines © 2026