

One Vector, One Fact: A Capacity Law for Single-Vector Text Embeddings

I. Kennedy and T. Kennedy

baa.ai, Auckland, New Zealand
research@baa.ai

Abstract. Dense retrieval compresses every chunk of text into a single vector, and the field’s consensus on what that costs is qualitative: single vectors “lose information,” and how much “depends on the model.” For the consumers a retrieval system actually deploys, we show it does not depend on the model. Across nine encoder stacks spanning retrieval, similarity, and LLM-based objectives, three dimensionalities (384–1024), and — decisively — a mean-pooled BERT encoder with no contrastive training at all, a chunk embedding of K independent facts yields approximately *one* fact’s worth of linearly recoverable information: a query-conditioned probe recovers $\approx 1/K$ of the facts, and slot-conditioned probes, which are not bounded by the probe’s form, decode 1.0–1.6 facts per chunk at every K (spread ≤ 0.04 across encoders for the query-conditioned probe at every K). Bilinear readouts, a three-layer MLP, and a trained projector feeding a frozen reader recover nothing beyond the ridge probe; the reader pipeline collapses to chance for $K \geq 4$. A matched-error compression contrast shows that the surviving information lives in low-variance directions of the pooled vector: low-rank truncation of matched reconstruction error erases it while isotropic noise and coordinate pruning of the same error do not, which predicts that scalar quantization of an index is safe where PCA-style compression is not. Retrieval experiments show the lost facts are also the *unfindable* ones, with a strong primacy bias on contrastively trained encoders, and the effect transfers to real corpora where chunks are topically heterogeneous (late-position facts in aviation-manual chunks retrieve at $\sim$ half the early rate on dense encoders, inside every chunk-length tertile, while a BM25 baseline shows no position penalty; the fact-count penalty we also observe is mostly a chunk-length effect shared with BM25; topically unified scientific abstracts are largely spared). Downstream, for xRAG-style soft-token injection we bracket compressed context with two walls: the encoder deletes all but $\sim$ one fact, and, under one projector recipe, neither frozen reader we test combines a strong soft channel with reliable routing among multiple payloads. The practical prescriptions are immediate: embed one atomic claim per vector, lead with the entity, and measure any new encoder with the four-minute linear-probe audit we release. All encoders we test use standard parameter-free pooling (mean, [CLS], or last token); learned pooling and contextualised chunking remain open.

Keywords: Dense retrieval · Text embeddings · Retrieval-augmented generation · Chunking · Context compression.

1 Introduction

Retrieval-augmented systems are built on a fixed interface: one chunk of text, one dense vector. The limitation of that interface is well known in the qualitative sense — longer chunks force lossier compression, single vectors “obscure fine-grained evidence,” and multi-vector architectures exist precisely to relieve the bottleneck [1,2,3,4]. But the field’s working answer to the quantitative question — *how many facts does a chunk embedding actually carry?* — has been “it depends”: on the encoder, the objective, the dimensionality, the corpus. Chunk size is treated as a tunable hyperparameter, and every RAG deployment tunes it by trial and error.

This paper’s central claim is that, for independent facts and for the consumers a retrieval system deploys, the answer is a constant of the interface, not a property of the model:

The capacity law. If a chunk containing K independent facts is embedded as a single vector by an encoder with standard parameter-free pooling (mean, [CLS], or last-token), then for retrieval-class consumers — similarity search, linear and small nonlinear readouts, and projector-frozen readers — the recoverable information is approximately *one fact*: a query-conditioned readout recovers $\approx 1/K$ of the facts and a slot-conditioned readout decodes 1.0–1.6 facts in total, regardless of encoder family, training objective, or dimensionality.

We establish the law with a controlled probe protocol (§3) and five sets of experiments:

1. **Universality** (§4): nine distinct encoder stacks — Arctic, E5, BGE-M3, Nomic, a Qwen3-based LLM embedder, MiniLM, MPNet, a production fine-tune, and a *mean-pooled BERT with no retrieval training* — yield query-conditioned probe accuracies of 0.51–0.55 at $K=2$, 0.28–0.31 at $K=4$, and 0.15–0.18 at $K=8$, against a chance-corrected single-fact baseline of 0.513, 0.269, and 0.147. Because a linear readout of $[\mathbf{e}(\text{chunk}) \parallel \mathbf{e}(\text{question})]$ is additive and therefore bounded near $1/K$ by construction, we add readouts that are not: per-slot linear probes on the chunk vector alone and a trained low-rank bilinear readout. They decode 1.0–1.6 facts per chunk at every K and never exceed the ridge probe, respectively. The non-retrieval contrast obeying the same law relocates the mechanism: it is single-vector pooling of transformer representations, not contrastive training.
2. **Practical inaccessibility** (§5): a three-layer MLP probe never exceeds the ridge probe (within 0.05 on every retrieval-trained encoder, markedly worse on BERT at 1,200 training chunks and still no better at 10,000), and a trained projector feeding a frozen 1.7B reader never exceeds the linear probe’s ceiling, collapsing to chance for $K \geq 4$. We discuss why heavyweight embedding inversion [10,11] does not rescue deployed pipelines.
3. **Retrievability and primacy** (§7): the deleted facts are also unfindable. Against maximally adversarial same-template distractor pools, worst-position

retrieval collapses by $K=2$ on every encoder, and on contrastively trained encoders the surviving fact is overwhelmingly the *first* one. Retrieval quality at $K=1$ spans 0.41–1.00 across encoders while capacity does not move — quality and capacity are dissociated axes.

4. **Real corpora** (§8): on 2,153 LLM-extracted verbatim QA pairs, heterogeneous aviation-manual chunks show the predicted position penalty on dense encoders (late-position facts retrieve at $\sim$ half the early rate, inside every chunk-length tertile) while a BM25 baseline on the same queries shows none; the fact-count penalty is largely a chunk-length effect that BM25 shares; topically unified scientific abstracts are largely spared. Topical competition and position, not fact count per se, are what the vector cannot represent.
5. **Downstream walls** (§9): for xRAG-style compressed-context injection, the encoder wall (this law) is joined by a second, *reader-dependent* wall under a single projector recipe: one frozen reader (Qwen3-1.7B) cannot route among multiple soft-token payloads even when each is individually decodable at 0.95; another (Llama-3.2-3B) routes correctly but has a far weaker soft channel under the same recipe. No configuration we tested achieves both.

The constructive output is an **audit**: a linear probe over synthetic multi-fact chunks that measures any encoder’s capacity curve in about four minutes on a laptop, with no reader, no training, and no labeled data. The prescriptions it justifies — one atomic claim per embedded vector, entity first, retrieve-small-return-big — have circulated as craft knowledge [7]; the contribution here is the measured law that makes them physics rather than preference, and the boundary conditions (topical heterogeneity) that say where they bind.

2 Related work

Single-vector limits. Theoretical work shows fixed-dimension embeddings cannot realize all top- k relevance patterns [4], operationalized by the LIMIT benchmark¹. Follow-up analysis shows dimensionality alone does not explain the observed failures — $2k+1$ dimensions suffice in principle for top- k retrieval — and attributes them instead to domain shift and misalignment between embedding similarity and relevance, a noisy proxy that degrades as the corpus grows [5]. Empirical work likewise documents that single vectors obscure evidence in long documents [2] and motivates multi-vector or multi-prefix representations [1,3]. Our contribution is complementary to both lines: a simple, reproducible *empirical capacity law* — when independent facts compete inside one vector, only $\approx$ one remains linearly recoverable, regardless of encoder family, objective, or dimension — with evidence that this limit directly predicts retrieval failures on heterogeneous real corpora, and a lightweight audit that measures it. Consistent with [5], our results locate the constraint away from raw dimensionality: capacity is flat from 384 to 1024 dimensions, far below what an information-theoretic account would allow.

¹ <https://github.com/google-deepmind/limit>

Pooling architectures. Every encoder we test aggregates token states with a parameter-free operator (mean, [CLS], or last token) over an independently encoded chunk. Encoders with learned latent-attention pooling [16], contextualised chunking that pools after full-document attention [17], and the long-context positional benchmarks LongEmbed [18] and LIMIT [4] lie outside the audit reported here; whether learned pooling escapes the law is the first question the released audit should be pointed at.

Chunking. Proposition-level retrieval [7] and systematic chunking studies [8,6] report that small, atomic, content-aware units retrieve best for factoid workloads. We supply the mechanism and the number: the unit that works is the unit that fits the interface’s $\sim$ one-fact capacity.

Position bias. Dense retrievers are known to over-weight document beginnings [9], and that work locates the bias in contrastive fine-tuning rather than in pre-training. Our per-position slices agree and separate two facts that are easy to conflate: *how many* facts survive is training-independent (the mean-pooled BERT obeys the same $1/K$), whereas *which* fact survives is training-dependent — every contrastively trained encoder keeps the first fact, while BERT without contrastive training shows no primacy at all (§7).

Embedding inversion. Trained inverters reconstruct surprising amounts of short text from embeddings [10], and per-sample optimization can cram far more [11]. These results bound what is *recoverable in principle*; we measure what is *accessible to retrieval-class consumers* (linear/small-MLP probes, trained projector+reader pipelines) and find the $1/K$ ceiling governs all of them.

Context compression. Soft-token methods compress context into few learned embeddings [12,13]. Prior evaluations use effectively single-fact payloads; we show the regime boundary for the xRAG-style design (a projector over an off-the-shelf retrieval embedding) and a reader-dependent routing wall that, to our knowledge, has not been isolated before. Architectures that pre-train or autoencode multi-slot memory tokens jointly with the reader [14,15] are a different design and are not evaluated here.

3 Probe protocol and audit

Synthetic chunks. A chunk with K facts concatenates K template sentences “*The secret code word assigned to the outpost E_i is V_i ,*” with entities E_i and values V_i . Entities are four-syllable strings drawn from 20 syllables (20^4 possible); every entity is unique across the 1,300 chunks of a run, so held-out entities never occur in training. Values are drawn with replacement from a 39-word vocabulary, so two facts in a chunk share a value with probability $1/39$. Facts are independent by construction — the worst case for a gist-preserving encoder, and therefore an upper-bound stress test. For each $K \in \{1, 2, 4, 8\}$ we generate 1,300 chunks (seed $100+K$; the last 100 held out). The question for fact i is “*What is the secret code word assigned to the outpost E_i ?*”. A control with six carrier templates of varied surface form, drawn per fact with the same entities and values, is reported in §4.

Encoding. Chunks and questions are embedded with each encoder’s released pooling and prefix conventions and L_2 -normalised: arctic-embed-l-v2 [25] ([CLS], query prefix “query: ”), e5-large-v2 [23] (mean, “query: ”/“passage: ”), bge-m3 [24] ([CLS], no prefix), nomic-embed-v1.5 [26] (mean, “search_query: ”/“search_document: ”), Qwen3-Embedding-0.6B [27] (last token, the model card’s retrieval instruction on queries), all-MiniLM-L6-v2 and all-mpnet-base-v2 [19,20,21] (mean, no prefix), bert-base-uncased [22] (mean, no prefix), and a production retrieval fine-tune of a 384-d MiniLM-class backbone (mean; its weights and its $K=1$ retrieval quality, 0.82 vs. 0.68, differ from all-MiniLM-L6-v2).

Extraction probe. For every fact we form the feature $[\mathbf{u} \parallel \mathbf{q}_i]$ with $\mathbf{u} = \mathbf{e}(\text{chunk})$ and $\mathbf{q}_i = \mathbf{e}(\text{question}_i)$ and train (i) a ridge readout ($\lambda=1$, one-hot targets, no bias term) and (ii) a three-layer MLP ($2d \rightarrow 512 \rightarrow 512 \rightarrow 39$, ReLU, Adam 10^{-3} , batch 512, 60 epochs) to classify the 39-way value on training chunks; we report held-out accuracy, overall and by fact position. Chance is $1/39 \approx 0.026$. A readout that recovers exactly one fact per chunk and guesses on the rest scores $1/K + (K-1)/(39K)$: 0.513, 0.269, 0.147 at $K = 2, 4, 8$. We call this the *single-fact baseline* and compare against it, not against $1/K$.

Why the probe needs slot-conditioned controls. The readout above is additive: logits are $W_u \mathbf{u} + W_q \mathbf{q}_i + \mathbf{b}$, and since values are independent of entities the question term carries no information about the value. For a fixed chunk the probe therefore emits one ranking for all K questions and can be right on at most one of them (plus chance), whatever $\mathbf{u}$ stores. Its $\approx 1/K$ is a bound of the probe, not yet a measurement of the vector. We therefore add three readouts that are not so bounded: (a) *per-slot ridge probes* $W^{(j)} \mathbf{u}$ that decode the value in slot j from the chunk vector alone, whose per-slot accuracies sum to the number of linearly decodable facts per chunk; (b) a *diagonal-bilinear* ridge on $[\mathbf{u} \parallel \mathbf{q}_i \parallel \sqrt{d} \mathbf{u} \odot \mathbf{q}_i]$; and (c) a *low-rank bilinear* readout, logits = $((\mathbf{u}A) \odot (\mathbf{q}_i B))C$ with rank 128, trained with AdamW and early-stopped on 100 training chunks. We also retrain the MLP with weight decay and early stopping, and on 10,000 training chunks for the two fastest encoders.

Retrievability. Each held-out question is scored against all 1,300 chunk embeddings (cosine); we report top-1 accuracy of the source chunk by fact position. Same-template distractors differ only in entities and values — maximal topical competition.

The audit. The full protocol — generation, embedding, both probes, retrievability, and a prescription (largest K whose worst-position top-1 stays $\geq 0.9 \times$ the $K=1$ value) — runs in ~ 4 minutes per encoder on one laptop GPU. No labels, no reader, no training beyond the probe itself. The slot-conditioned controls add about two minutes.

Table 1. Held-out extraction accuracy of the query-conditioned ridge probe vs. facts-per-chunk K . The single-fact baseline $1/K + (K-1)/(39K)$ is what a readout that recovers exactly one fact and guesses on the rest would score. Every encoder, including the non-retrieval BERT contrast, sits within ± 0.04 of it; spread across encoders at each K is ≤ 0.04 . Held-out $n = 100$ chunks ($100K$ queries) per cell.

Encoder	Objective	Dim	$K=1$	$K=2$	$K=4$	$K=8$
<i>reference:</i> $1/K$			1.00	0.50	0.25	0.125
<i>single-fact baseline</i> $1/K + \frac{K-1}{39K}$			1.00	0.513	0.269	0.147
arctic-embed-l-v2	retrieval	1024	1.00	0.52	0.29	0.17
e5-large-v2	retrieval	1024	1.00	0.51	0.29	0.18
bge-m3	retrieval	1024	1.00	0.54	0.31	0.17
nomic-embed-v1.5	retrieval	768	1.00	0.53	0.28	0.16
qwen3-embedding-0.6b	LLM retrieval	1024	1.00	0.53	0.30	0.15
all-MiniLM-L6-v2	similarity	384	1.00	0.53	0.28	0.17
all-mpnet-base-v2	similarity	768	1.00	0.53	0.29	0.17
prod-mini (our FT)	retrieval FT	384	1.00	0.53	0.28	0.16
bert-base (mean-pool)	none	768	0.99	0.55	0.28	0.16

4 The law, and its universality

Table 1 reports query-conditioned ridge-probe accuracy for nine distinct encoder stacks (three production fine-tunes in our fleet share stacks with public backbones and reproduce their curves to two decimals; we report distinct stacks).

Four observations. First, the fit is tight and universal: no encoder deviates by more than 0.04 from any other at any K in Table 1, and every cell sits within ± 0.04 of the single-fact baseline. The excess over exact $1/K$ that we previously read as partial recovery of a second fact is mostly the chance floor on the unrecovered facts; what remains after chance correction is at most ± 0.04 . Second, the result is not an artefact of the additive probe. Table 2 shows that slot-conditioned linear probes, which can in principle decode every fact if the vector stores them in separable directions, decode 1.0–1.6 facts per chunk at every K , and the trained bilinear readout never exceeds the ridge probe (within 0.03 everywhere). Later slots are decodable above chance (0.14–0.43 at $K=2$) but far below the first slot, and their sum never approaches K . Third — the decisive contrast — **bert-base with mean pooling and no contrastive training obeys the same law** (1.18, 0.92, 0.99 decodable facts at $K = 2, 4, 8$). The mechanism is therefore not the retrieval objective selecting a query-matching subspace; it is single-vector pooling of transformer token representations itself. Contrastive training changes *which* single fact dominates and how well queries find the vector (§7), not how many facts survive. Fourth, the law does not depend on the repeated carrier sentence: with six varied templates (same entities and values) the query-conditioned probe scores 0.50–0.53, 0.26–0.29 and 0.15–0.17 at $K = 2, 4, 8$ on arctic, e5, MiniLM and BERT, and the slot-conditioned probes decode 1.0–1.4 facts per chunk. Training on 10,000 chunks instead of

Table 2. Readouts that are not bounded by the additive probe’s form (held-out $n = 100$ chunks, fixed template). *Slot mean*: mean over slots j of the per-slot ridge accuracy $W^{(j)}\mathbf{u}$. *Facts*: linearly decodable facts per chunk, $\sum_j \text{acc}_j$. *Slot 0 / 1*: per-slot accuracy of the first and second fact. *Bilinear*: trained rank-128 bilinear readout. Chance per slot is 0.026. The diagonal-bilinear ridge equals the query-conditioned ridge to three decimals in every cell and is omitted.

Encoder	$K=2$				$K=4$				$K=8$	
	Slot 0	/ 1	Facts	Bilinear	Slot 0	/ 1	Facts	Bilinear	Facts	Bilinear
arctic-embed-l-v2	0.88	/ 0.22	1.10	0.51	0.95	/ 0.14	1.36	0.27	1.38	0.16
e5-large-v2	0.64	/ 0.43	1.07	0.52	0.46	/ 0.30	1.13	0.29	1.37	0.18
bge-m3	0.64	/ 0.43	1.07	0.52	0.70	/ 0.22	1.20	0.28	1.40	0.18
nomic-embed-v1.5	0.91	/ 0.19	1.10	0.52	0.80	/ 0.17	1.27	0.29	1.43	0.15
qwen3-embedding-0.6b	0.91	/ 0.31	1.22	0.54	0.93	/ 0.20	1.31	0.29	1.63	0.16
all-MiniLM-L6-v2	0.82	/ 0.28	1.10	0.52	0.71	/ 0.21	1.24	0.29	1.41	0.17
all-mpnet-base-v2	0.89	/ 0.31	1.20	0.52	0.81	/ 0.16	1.39	0.29	1.38	0.15
bert-base (mean-pool)	0.57	/ 0.61	1.18	0.52	0.20	/ 0.17	0.92	0.26	0.99	0.15

1,200 (BERT and MiniLM, $K=2$) raises the decodable total from 1.1–1.2 to 1.3 and leaves the query-conditioned probe at 0.52. Section 6 asks where in the vector the surviving fact lives.

5 Nonlinear probes add nothing; readers never exceed the probe

A natural objection: a linear probe lower-bounds recoverable information, and trained inverters can reconstruct much more from short-text embeddings [10,11]. Three measurements answer it at the level that matters for deployed systems.

MLP $\leq$ ridge. The three-layer MLP probe matches ridge within 0.05 at every K on every retrieval-trained encoder (e.g., at $K=4$: MLP 0.26–0.31 vs. ridge 0.28–0.31). On the non-contrastive BERT features it is markedly *worse* (0.30 vs. 0.55 at $K=2$), a small-sample overfitting effect: with weight decay and early stopping it reaches 0.37, and with 10,000 training chunks 0.48 against a ridge 0.525. Nowhere does nonlinearity reveal reserve capacity.

Readers never exceed the probe. In our injection rig (arctic embedding $\rightarrow$ trained 2-layer projector $\rightarrow m=8$ soft tokens $\rightarrow$ frozen Qwen3-1.7B), held-out QA accuracy at $K = 1, 2, 4, 8, 16$ is 0.97, 0.45, 0.03, 0.05, 0.02, against a probe curve of 1.00, 0.52, 0.29, 0.17, 0.13 and a full-text control ≥ 0.975 everywhere. The trained pipeline tracks the probe at $K \leq 2$ and then falls to the 0.026 chance floor: past $K=2$ it loses even the first fact that remains linearly decodable in the embedding (Table 2). The projector was trained at a data scale previously shown to saturate this rig at $K=1$ (held-out 0.08 $\rightarrow$ 0.99 as training documents scale 40 $\rightarrow$ 1200); we did not re-run the data-scaling sweep at $K \geq 4$, so saturation there is assumed, not shown. The statement that matters for system design is the bound: no reader we trained recovered more than the probe, at any K .

Table 3. Extraction accuracy under *matched*-error compression of the chunk embedding ($e^* \approx 0.21$ relative Frobenius), with 95% cluster-bootstrap CIs over 400 held-out chunks and retention ρ in parentheses. Low-rank truncation is CI-disjoint below both incoherent controls for $K \geq 4$ (low-rank minus noise: $-0.036 [-0.051, -0.023]$, $-0.064 [-0.075, -0.053]$, $-0.081 [-0.088, -0.074]$ absolute accuracy at $K = 4, 8, 16$). Retention values above 1.00 are within the CI of the clean cell.

	$K=1$		$K=2$		$K=4$		$K=8$		$K=16$	
clean	1.00		0.514	[.503,.525]	0.286	[.276,.296]	0.171	[.163,.179]	0.121	[.115,.127]
low-rank (coherent)	1.00 (1.00)	0.510	[.493,.528]	(0.99)	0.251	[.236,.266] (0.88)	0.107	[.098,.117] (0.63)	0.034	[.030,.038] (0.28)
isotropic noise	1.00 (1.00)	0.520	[.508,.533]	(1.01)	0.288	[.278,.298] (1.01)	0.171	[.163,.180] (1.00)	0.115	[.109,.121] (0.96)
coordinate prune	1.00 (1.00)	0.516	[.505,.528]	(1.01)	0.286	[.276,.295] (1.00)	0.171	[.164,.179] (1.00)	0.120	[.114,.126] (0.99)
rank r at e^*	64		50		40		26		4	

Scope of the claim. We do not claim the information is destroyed in the information-theoretic sense — per-sample optimization can store orders of magnitude more in a vector [11]. We claim, and measure, that no consumer of the class deployed in retrieval systems (similarity search, linear/small-MLP readouts, trained projector-plus-frozen-reader channels) accesses more than the $\sim$ one-fact ceiling. For system design, that is the operative bound.

6 Where the surviving fact lives: a matched-error contrast

Section 4 localizes the law to single-vector pooling rather than the training objective. A sharper question is where in the pooled vector the recoverable information sits, and which index compressions therefore destroy it. We answer with a controlled operator contrast on the chunk embedding itself (arctic-embed-l-v2; 1,200 training and 400 held-out chunks per K), holding the relative Frobenius reconstruction error $e^* = \|\tilde{\mathbf{X}} - \mathbf{X}\|_F / \|\mathbf{X}\|_F$ on the held-out matrix *matched* across operators. A *coherent* distortion — low-rank truncation that projects embeddings onto the top- r principal subspace of the training embeddings, centred at the training mean — is contrasted with two *incoherent* controls at the same e^* : isotropic Gaussian noise, and zeroing a random shared subset of coordinates (fraction e^{*2} , 4.4–4.9% of 1,024). The rank is chosen from a grid to bring the held-out error nearest 0.22 (achieved 0.21–0.22); because averaging K independent facts shrinks the across-chunk variance, the matching rank falls with K : $r = 64, 50, 40, 26, 4$ for $K = 1, 2, 4, 8, 16$ (variance explained 0.89, 0.82, 0.74, 0.59, 0.14). The ridge probe ($\lambda=1$, not re-tuned) is re-fit on the compressed training embeddings and evaluated on the compressed held-out set, so the readout is charitable to the compressor. We report absolute accuracy with cluster-bootstrap 95% CIs over held-out chunks and retention $\rho(K) = \text{acc}_{\text{op}}(K) / \text{acc}_{\text{clean}}(K)$.

Three things follow from Tables 3 and 4. First, the clean embedding has already lost all but $\sim$ one fact before any operator is applied (clean accuracy is itself the $1/K$ curve), so the contrast measures what compression does to

Table 4. The same contrast by fact position (arctic-embed-l-v2, $e^* \approx 0.21$). Accuracy on the first fact (position 0) and on all later facts (positions ≥ 1 ; chance 0.026). Low-rank truncation erases the first fact’s decodability at large K ; the incoherent controls leave it intact.

	position 0				positions ≥ 1			
	$K=2$	$K=4$	$K=8$	$K=16$	$K=2$	$K=4$	$K=8$	$K=16$
clean	0.98	0.98	0.83	0.62	0.05	0.06	0.08	0.09
low-rank	0.76	0.57	0.23	0.04	0.26	0.15	0.09	0.03
isotropic noise	0.97	0.95	0.78	0.55	0.07	0.07	0.08	0.09
coordinate prune	0.98	0.98	0.83	0.62	0.05	0.06	0.08	0.09

the *surviving* information, not how pooling deleted the rest. We previously described the low-rank result as “reproducing” the pooling collapse; that was an over-reading. Second, the surviving information is fragile in a specific way: the query-conditioned readout depends on low-variance directions of the pooled vector. Truncating to the principal subspace at matched error removes them — at $K=16$ the rank that matches e^* is 4 and the surviving first fact falls from 0.62 to 0.04 — while noise or pruning of equal Frobenius error leaves them intact ($\rho \geq 0.96$ everywhere). At $K=2$ truncation redistributes rather than removes: the first fact drops ($0.98 \rightarrow 0.76$) while later facts rise ($0.05 \rightarrow 0.26$), and the total is unchanged. Third, and practically: scalar or near-incoherent quantization of an index perturbs all directions in proportion and stays inside the budget, whereas PCA-style or low-rank embedding compression of the same magnitude removes precisely the directions the remaining fact-discriminative signal occupies. Whether token pooling deletes subordinate facts by the same geometric route is a plausible conjecture that this post-pooling experiment does not test.

7 The lost facts are unfindable, and after contrastive training they are the late ones

Extraction measures what is *in* the vector; retrieval measures whether a query about fact j *finds* the chunk. Against 1,300 same-template competitors, worst-position top-1 collapses at $K=2$ on every encoder: $0.68 \rightarrow 0.08$ (all-MiniLM-L6-v2), $0.82 \rightarrow 0.12$ (prod-mini), $1.00 \rightarrow 0.29$ (arctic-embed-l-v2 and its production twin, which we call the arctic class), $0.99 \rightarrow 0.56$ (bge-m3, the softest failure we measured). By $K=4$ the worst position is at or near zero everywhere. On contrastively trained encoders the surviving fact is the first one: at $K=2$ top-1 by position is 0.93/0.29 for the arctic class, 0.93/0.13 for nomic, 0.83/0.31 for qwen3-embedding, 0.73/0.42 for e5, 0.81/0.56 for bge-m3, 0.49/0.08 for all-MiniLM-L6-v2, 0.26/0.03 for all-mpnet-base-v2. The per-slot probes of Table 2 show the same primacy inside the vector. The non-contrastive bert-base shows none: 0.07/0.11 in retrieval and 0.57/0.61 in the per-slot probe. This is consistent with the beginning-bias literature [9], which locates the bias in contrastive fine-

tuning, and it separates the two halves of our claim: capacity ($\sim$ one fact) is training-independent; which fact survives is a product of training.

The same experiment dissociates quality from capacity. At $K=1$, top-1 against the adversarial pool ranks encoders from 0.41 (mpnet) through 0.68 (MiniLM) to 1.00 (arctic class, qwen3-embedding, nomic): retrieval *quality* varies by $2.4\times$ across models. Capacity (Tables 1 and 2) varies by ≤ 0.04 in the query-conditioned probe and by 1.0–1.6 decodable facts. An encoder upgrade buys findability of the one fact the vector keeps; it does not buy a second fact.

8 Real corpora: heterogeneity is the moderator

We extracted verbatim factoid QA pairs from two real corpora: a proprietary wide-body aircraft operations manual (7,183 chunks) and the SciFact abstracts (5,183). From each corpus 250 chunks of 50–220 words were sampled uniformly at random; Qwen2.5-7B-Instruct was prompted to extract up to five atomic question–answer pairs per chunk with answers of one to five words copied verbatim and questions that name their subject explicitly, spread across the passage. Pairs whose answer string does not occur verbatim in the chunk were dropped, and chunks with fewer than two surviving pairs were discarded, leaving 239 manual chunks (954 pairs; mean 4.0 per chunk) and 248 abstracts (1,199 pairs; mean 4.8), 2,153 pairs in total. Each question retrieves against its full corpus; Table 5 slices source-chunk hit@1 by the answer’s character offset within the chunk (early $< 1/3$, mid, late $> 2/3$) and by the number of extracted facts in the chunk (exactly 2, 3, 4, or ≥ 5). Cell sizes: manual 378 / 304 / 272 queries by position and 30 / 234 / 160 / 530 by fact count; SciFact 473 / 396 / 330 by position and 0 / 9 / 140 / 1,050 by fact count — no SciFact abstract in the sample yielded exactly two facts, so the two-fact cell does not exist there, and the manual’s two-fact cell (15 chunks) is small. A BM25 baseline over the same chunks and queries (Okapi, $k_1=1.5$, $b=0.75$) is reported alongside.

On the manual — whose chunks mix procedures, limits, and values — the dense position prediction transfers: late-position facts retrieve at roughly half the early rate on e5-large-v2 (0.36 [0.31, 0.41] $\rightarrow$ 0.18 [0.14, 0.22]) and MiniLM (0.24 [0.20, 0.29] $\rightarrow$ 0.12 [0.09, 0.16]) and at 70% on the arctic class (0.35 [0.31, 0.41] $\rightarrow$ 0.25 [0.20, 0.30]), and hit@1 also falls with fact count on every dense encoder (e5, 3 $\rightarrow$ 5+ facts: 0.37 [0.31, 0.44] $\rightarrow$ 0.25 [0.21, 0.29]). Two controls separate the confounds a reader would raise. *Query underspecification*: BM25 answers early- and late-offset questions equally well (0.36 [0.31, 0.41] vs. 0.36 [0.31, 0.42]), and late questions are no shorter (8.3 vs. 7.9 words), so the late questions carry enough lexical signal to find their chunk; it is the single vector that fails to carry the late fact. *Chunk length*: splitting the manual’s chunks into word-count tertiles (< 58 , 58–68, > 68 words), the position penalty holds inside every tertile (e5 early $\rightarrow$ late: 0.46 $\rightarrow$ 0.27, 0.42 $\rightarrow$ 0.26, 0.19 $\rightarrow$ 0.02), so it is not a length artefact. The fact-count penalty does not survive the same split: inside a tertile hit@1 is non-monotone in fact count (e5, shortest tertile: 0.49, 0.32, 0.42 for 3, 4, 5+ facts), every channel including BM25 loses heavily on the longest tertile (5+-fact

Table 5. Real-corpus retrievability (hit@1 of the source chunk) by answer position and by facts-per-chunk. On the heterogeneous manual, dense encoders show a position penalty that BM25 on the same queries does not, and a count penalty that BM25 shares; on topically unified abstracts neither channel is penalised. Cell sizes: manual 378 / 304 / 272 queries (early / mid / late) and 30 / 234 / 160 / 530 (2 / 3 / 4 / 5+ facts); SciFact 473 / 396 / 330 and 0 / 9 / 140 / 1,050. The 3 $\rightarrow$ 5+ comparison is given because the manual’s 2-fact cell has 30 queries and SciFact has no 2-fact cell. 95% bootstrap CIs for the headline cells are in the text; all cells are in the released results file.

Encoder	Corpus	overall	early	$\rightarrow$ late	2 $\rightarrow$ 5+ facts	3 $\rightarrow$ 5+ facts
e5-large-v2	Manual	0.30	0.36	$\rightarrow$ 0.18	0.53 $\rightarrow$ 0.25	0.38 $\rightarrow$ 0.25
arctic class	Manual	0.32	0.35	$\rightarrow$ 0.25	0.47 $\rightarrow$ 0.28	0.37 $\rightarrow$ 0.28
MiniLM-L6-v2	Manual	0.20	0.25	$\rightarrow$ 0.12	0.33 $\rightarrow$ 0.17	0.23 $\rightarrow$ 0.17
BM25	Manual	0.39	0.36	$\rightarrow$ 0.36	0.47 $\rightarrow$ 0.32	0.48 $\rightarrow$ 0.32
e5-large-v2	SciFact	0.61	0.61	$\rightarrow$ 0.59	—	0.56 $\rightarrow$ 0.62 (4 $\rightarrow$ 5+)
arctic class	SciFact	0.59	0.59	$\rightarrow$ 0.56	—	0.54 $\rightarrow$ 0.59 (4 $\rightarrow$ 5+)
MiniLM-L6-v2	SciFact	0.40	0.37	$\rightarrow$ 0.39	—	0.38 $\rightarrow$ 0.40 (4 $\rightarrow$ 5+)
BM25	SciFact	0.81	0.79	$\rightarrow$ 0.80	—	0.81 $\rightarrow$ 0.81 (4 $\rightarrow$ 5+)

chunks: e5 0.42 short vs. 0.11 long, BM25 0.41 vs. 0.21), and 5+-fact chunks are over-represented there (220 of their 530 queries). The count penalty we reported in the 2026-08 draft is therefore mostly a chunk-length penalty shared with the lexical channel; the position penalty is specific to the single vector. On abstracts, where every extracted fact concerns the same study, the penalties largely vanish for both channels (position spread ≤ 0.03 overall; 4 $\rightarrow$ 5+ facts flat or slightly up) and BM25 reaches 0.81: SciFact questions carry study-specific terms that find the abstract regardless of where the answer sits. Within the shortest tertile of abstracts (< 152 words) a late-position penalty does appear on the dense encoders (e5 0.65 $\rightarrow$ 0.46, arctic 0.63 $\rightarrow$ 0.48) and not on BM25 (0.75 $\rightarrow$ 0.76), so topical unity attenuates the effect rather than abolishing it. This refines the mechanism: the vector preserves the chunk’s *gist direction*; facts aligned with the gist remain findable, facts that *compete* with it are the ones lost. Independent synthetic facts are maximal competition — the correct worst case for an audit — and heterogeneous corpora are the deployments where atomic chunking pays. The cross-corpus comparison remains observational: the manual’s chunks also share more vocabulary with one another than abstracts of distinct studies do, and we have not isolated topical heterogeneity from lexical distinctiveness.

9 Downstream: two walls for compressed context

Soft-token injection promises large token savings by replacing retrieved text with projected embeddings [12,13]. We test the xRAG-style design: a 2-layer projector (1,024 hidden units, GELU) maps a frozen arctic-embed-l-v2 embedding to m

Table 6. Multi-payload injection, held-out QA accuracy. m is the number of soft tokens per source vector (single-vector arms use $m=8$). Each per-fact embedding is individually decodable ($K=1$, $m=2$: 0.95 on Qwen). Routing among payloads is reader-dependent, and under this projector recipe no reader combines a strong soft channel with reliable routing. The recipe was tuned on Qwen and deliberately not re-tuned for Llama; cells marked — were not run.

Arm	Qwen3-1.7B	Llama-3.2-3B
text control ($K=2$)	0.98	1.00
single vector, $K=1$, $m=8$ (soft ceiling)	0.97	0.50
single vector, $K=2$, $m=8$	0.45	0.18
multi-vector, $K=2$, $m=2$ /fact (4 tokens)	0.07	—
multi-vector, $K=2$, $m=8$ /fact (16 tokens)	0.14	0.48
single fact, $K=1$, $m=2$ (control)	0.95	—

reader input embeddings, prepended to the chat-templated prompt “*Question: ... Answer with only the answer word, no explanation*”; the reader is frozen and the projector is trained for 2,500 AdamW steps on 1,200 training chunks with every fact as a sample, then evaluated on 100 held-out chunks by exact match of the answer word. The capacity law sets the first wall: through a single chunk embedding, injection matches full text only for single-fact payloads (§5). The obvious repair — embed each fact sentence separately, project each with the shared projector to m soft tokens, and concatenate the groups in document order with no delimiter between them, training the projector end-to-end on the concatenated prefix — exposes a second wall.

On Qwen3-1.7B (Table 6), multi-vector injection *underperforms* the single vector it was meant to fix (0.14 vs. 0.45 at $m=8$ per fact; 0.07 at $m=2$ per fact, and 0.08 and 0.04 at $K=4$ and $K=8$), although two soft tokens per fact are sufficient for one fact ($K=1$, $m=2$: 0.95) and eight per fact do not help: the information is present, addressable group-by-group, and the frozen reader cannot select the right group. The projector was trained on the concatenated prefixes, so this is not a zero-shot prefix-length mismatch; whether delimiter tokens between groups would restore routing we did not test. On Llama-3.2-3B the pattern *reverses*: multi-vector reaches that reader’s own soft ceiling (0.48 vs. 0.50) with balanced positions (0.43/0.53) — it routes — but the ceiling itself is low under a recipe tuned for Qwen, and we have not re-tuned the projector for Llama, so its soft ceiling is a lower bound. Binding among soft payloads is thus a *reader-dependent* capability, not an architectural constant; in the configurations we tested no reader combined a strong soft channel with reliable routing, and settling the question requires per-reader tuning and the jointly trained multi-slot designs [14,15] that we do not evaluate. The cross-reader-safe regime for xRAG-style compressed context remains **one atomic payload per prompt** — which is also precisely the regime the capacity law forces on the encoder side.

10 Discussion

Why $\sim$ one fact? A pooled transformer representation is dominated by a single summary direction per chunk; competing facts superpose and interfere rather than occupying orthogonal, linearly addressable subspaces — the slot-conditioned probes of Table 2 find later facts above chance but never a second whole fact. §6 adds that what survives is carried by low-variance directions, so it is removed by exactly the compressions that keep the dominant direction. The constancy across objectives — including no objective at all — suggests the bottleneck is what parameter-free pooling preserves of token-level structure, not what contrastive training selects. A full theoretical account connecting this to embedding-geometry limits and the similarity-as-noisy-proxy account [4,5] remains future work.

Prescriptions. (1) *Retrieve small, return big*: embed small units, lead with the entity, and return the parent section to the LLM for context. In our retrieval bake-off on a 2,566-chunk technical-PDF corpus (200 held-out questions, gold = source chunk, arctic-embed-l-v2 with a cross-encoder reranker), parent-document retrieval — 380-character child windows with 60-character overlap, retrieve children, collapse to the parent, rerank — scored hit@3 0.945 against 0.860 for whole-chunk embedding, re-confirmed on a 450-question holdout (0.949 vs. 0.858); the law explains why. The extreme of the scheme has costs the law also predicts. On the aviation manual of §8, sentence-atomic children (one sentence per vector) lifted dense-only hit@1 for facts buried in the middle of a 380-character window by +0.095 [+0.015, +0.175] but lowered it for leading facts by −0.085 [−0.165, −0.010], because the packed window’s embedding was already dominated by its first fact; with BM25 fusion and a reranker over the parents the end-to-end difference was zero (+0.003 [−0.005, +0.013] hit@3), while the index grew from 9,820 to 17,923 vectors and dense latency from 1.6 to 2.7 ms. Atomic chunking multiplies the vector index by roughly the facts-per-chunk, needs a proposition-extraction pass at indexing time, and pays off in the dense channel; a strong lexical-plus-rerank pipeline recovers most of the same buried facts. (2) *Audit before indexing*: the four-minute probe yields any encoder’s capacity curve and a chunking prescription; on our production embedder fleet it also exposed shared embedder stacks via identical curves, a useful provenance check. (3) *Spend upgrades correctly*: buying a better encoder improves findability of the kept fact, not the number of kept facts; if queries target mid- and late-chunk content on heterogeneous corpora, the fix is chunking, not model size.

Limitations. Synthetic facts are independent and unambiguous; real capacity is lower where facts share entities, and effectively higher where they share topic (§8) — the synthetic number is an upper-bound stress test, not a production forecast. The probes bound retrieval-class access, not information-theoretic content [10,11]: the slot-conditioned and bilinear readouts remove the additive probe’s structural cap, but all readouts are linear or small. All nine encoders use parameter-free pooling over independently encoded chunks; learned latent-attention pooling [16], late chunking [17], and the LongEmbed and LIMIT benchmarks [18,4] are untested, so the law is established for the standard pooling

family, not for every single-vector encoder. The real-corpus comparison is observational on two corpora and does not separate topical heterogeneity from lexical distinctiveness. The reader experiments use one xRAG-style projector recipe tuned on one reader; Llama’s soft ceiling (0.50) is a lower bound, the Llama $K=1$, $m=2$ control and a delimiter control were not run, and per-reader tuning might shift the routing picture. Encoder coverage, while broad (nine stacks, three objectives, one non-contrastive contrast), excludes API-only embedders and one public model whose remote-code loader hung in our environment.

11 Conclusion

“How many facts fit in a chunk embedding” has a measured answer: about one — for every standard-pooling encoder we tested, by every consumer a retrieval system deploys, with a primacy bias deciding which one on contrastively trained encoders, a topical-heterogeneity condition deciding where it hurts, and a reader-dependent routing wall blocking the obvious multi-payload escape under the recipe we tested. Chunk size is not a hyperparameter to tune. It is an interface constant to respect, and a four-minute probe tells you exactly where it binds for the encoder you ship.

References

1. O. Khattab and M. Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. *SIGIR*, 2020.
2. S. Lyu et al. Lost in a single vector: Improving long-document retrieval with chunk evidence aggregation. *arXiv:2606.18781*, 2026.
3. Z. Yu et al. Improving long-context retrieval with multi-prefix embedding. *arXiv:2606.23642*, 2026.
4. O. Weller et al. On the theoretical limitations of embedding-based retrieval. *arXiv:2508.21038*, 2025.
5. A. S et al. On strengths and limitations of single-vector embeddings. *arXiv:2603.29519*, 2026.
6. M. A. Shaukat, M. Adnan, and C. C. N. Kuhn. A systematic investigation of document chunking strategies and embedding sensitivity. *arXiv:2603.06976*, 2026.
7. T. Chen et al. Dense X retrieval: What retrieval granularity should we use? *EMNLP*, 2024.
8. S. R. Bhat et al. Rethinking chunk size for long-document retrieval: A multi-dataset analysis. *arXiv:2505.21700*, 2025.
9. J. Coelho et al. Dwell in the beginning: How language models embed long documents for dense retrieval. *ACL*, 2024.
10. J. X. Morris et al. Text embeddings reveal (almost) as much as text. *EMNLP*, 2023.
11. Y. Kuratov et al. Cramming 1568 tokens into a single vector and back again: Exploring the limits of embedding space capacity. *arXiv:2502.13063*, 2025.
12. X. Cheng et al. xRAG: Extreme context compression for retrieval-augmented generation with one token. *NeurIPS*, 2024.

13. J. Mu, X. Li, and N. Goodman. Learning to compress prompts with gist tokens. *NeurIPS*, 2023.
14. D. Rau et al. Context embeddings for efficient answer generation in RAG. *arXiv:2407.09252*, 2024.
15. T. Ge et al. In-context autoencoder for context compression in a large language model. *arXiv:2307.06945*, 2023.
16. C. Lee et al. NV-Embed: Improved techniques for training LLMs as generalist embedding models. *arXiv:2405.17428*, 2024.
17. M. Günther et al. Late chunking: Contextual chunk embeddings using long-context embedding models. *arXiv:2409.04701*, 2024.
18. D. Zhu et al. LongEmbed: Extending embedding models for long context retrieval. *arXiv:2404.12096*, 2024.
19. N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *EMNLP*, 2019.
20. W. Wang et al. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv:2002.10957*, 2020.
21. K. Song et al. MPNet: Masked and permuted pre-training for language understanding. *arXiv:2004.09297*, 2020.
22. J. Devlin et al. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.
23. L. Wang et al. Text embeddings by weakly-supervised contrastive pre-training. *arXiv:2212.03533*, 2022.
24. J. Chen et al. M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv:2402.03216*, 2024.
25. L. Merrick et al. Arctic-Embed: Scalable, efficient, and accurate text embedding models. *arXiv:2405.05374*, 2024.
26. Z. Nussbaum et al. Nomic Embed: Training a reproducible long context text embedder. *arXiv:2402.01613*, 2024.
27. Y. Zhang et al. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv:2506.05176*, 2025.