Lost in a Single Vector: Improving Long-Document Retrieval with Chunk Evidence Aggregation

Anonymous ACL submission

Abstract

Dense retrieval ranks one query vector against one document vector. On long documents, this interface can fail when a short but decisive span is weakened during document encoding before ranking. We study this failure mode as *document-side early compression* and introduce the Evidence Dilution Index (EDI) to measure how far a document-level representation falls below the strongest chunk-level evidence within the same gold document. Guided by this view, we propose DICE (**D**ocument **I**nference via **C**hunk **E**vidence), a training-free document-side strategy that splits documents into chunks, encodes them independently with a frozen model, and aggregates them back into a single vector while preserving the standard one-query-one-document interface. On LongEmbed, DICE improves retrieval across four backbones, with the largest gains on slices beyond 4k tokens: for Dream, Passkey >4k rises from 30.0 to 90.0 and Needle >4k from 23.3 to 74.0. Across 12,779 filtered samples, DICE yields lower EDI than the single-vector baseline in 92.8% of cases. These results establish document-level encoding as a practical and underexplored lever for long-document retrieval.

1 Introduction

Retrieval systems are increasingly asked to find answers inside long documents such as meeting transcripts, legal filings, and narrative texts, where the decisive evidence may occupy only a few sentences amid thousands of irrelevant tokens. Dense retrieval handles this by encoding each document into a single vector and ranking by query–document similarity. The interface is simple and scales to billions of documents, but it creates a representational bottleneck: one vector must summarize an entire document, even when relevance turns on a short local span.

This bottleneck is not merely a capacity problem. Modern encoders routinely support context windows of 4k–32k tokens, yet retrieval quality still degrades on long documents (Zhu et al., 2024), and models systematically under-utilize information away from salient positions (Liu et al., 2024). The underlying cause is representational: when a query is resolved by a few decisive sentences embedded in a long document, encoding the whole document compresses the relevant evidence together with a large volume of irrelevant context before any query comparison is made. We refer to this failure mode as **document-side early compression**. As Figure 1 illustrates, the gold document may contain the answer-bearing span and still receive a weak document-level score, because the evidence is diluted before retrieval ever happens.

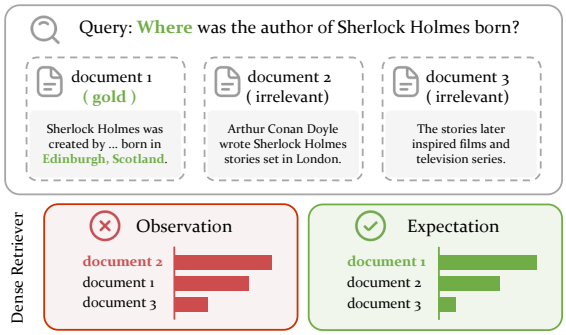

Figure 1: A motivating long-document retrieval case. The gold document contains a decisive local span, yet single-vector encoding under-ranks it because the evidence is diluted before retrieval.

This failure mode is difficult to address within the standard single-vector interface, and equally difficult to measure directly. Retrieval quality degrades with context length (Zhu et al., 2024), but that degradation conflates many factors and does not isolate the contribution of document-side compression. Methods that resolve the problem more directly, such as passage retrieval and late-interaction scoring (Khattab and Zaharia, 2020; Santhanam et al., 2022), do so by changing the re-

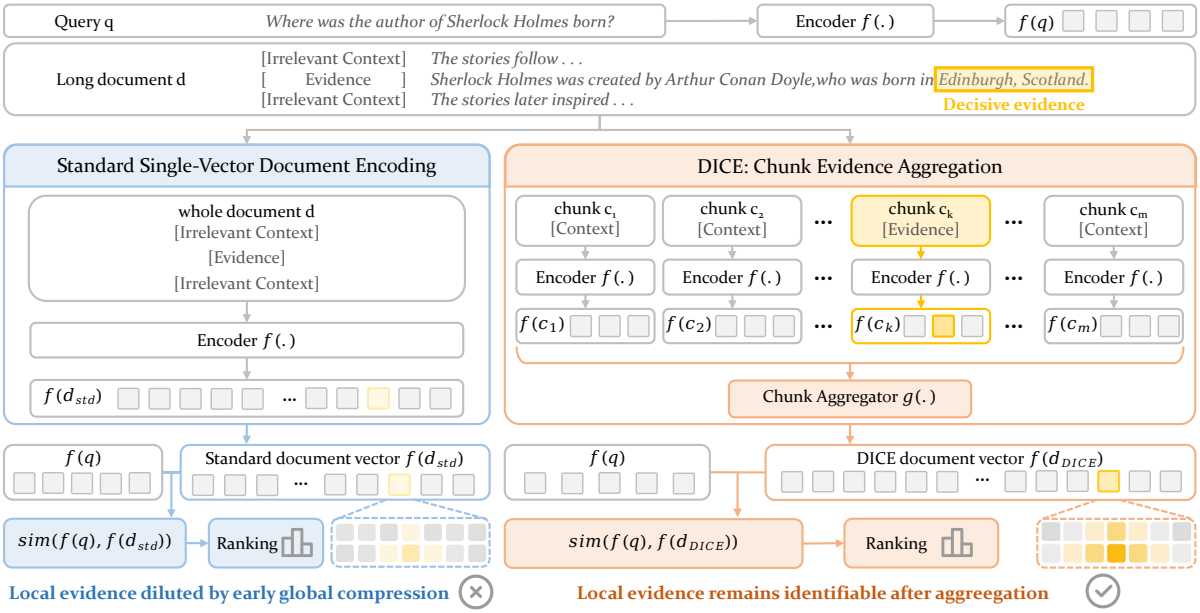

Figure 2: Overview of DICE. Only the document side changes: chunks are encoded independently with local positions and aggregated into one vector. The query path and the retrieval interface remain unchanged.

trieval unit or the scoring interface entirely. This leaves the document encoding side as a practical and underexplored degree of freedom.

We take two steps to exploit it. First, we introduce the **Evidence Dilution Index** (EDI), a per-sample diagnostic that compares a document-level similarity score against the strongest chunk-level evidence within the same gold document, turning early compression from an intuition into a measurable property. Second, guided by this diagnostic, we propose DICE (**D**ocument **I**nference via **C**hunk **E**vidence), a training-free strategy that splits a document into chunks, encodes each chunk independently with local position indices, and aggregates the resulting embeddings back into a single document vector (Figure 2). Query encoding is unchanged, so the retrieval interface remains standard one-query-one-document retrieval.

We evaluate DICE on LongEmbed (Zhu et al., 2024) across four backbones spanning diffusion (Dream) and autoregressive (Mistral (Jiang et al., 2023), Llama3 (Grattafiori et al., 2024), Qwen (Yang et al., 2024)) architectures. DICE consistently improves over single-vector baselines, with the largest gains on the hardest long-context slices: for Dream, Passkey >4k rises from 30.0 to 90.0 and Needle >4k from 23.3 to 74.0. Ablations identify chunk granularity as the key design factor, and EDI analysis explains the mechanism: across 12,779 filtered samples, DICE yields lower EDI than the single-vector baseline in 92.8% of cases. FollowIR (Weller et al., 2024) provides secondary transfer evidence beyond explicitly long-document benchmarks, though the best chunk configuration is task dependent.

Our contributions are:

- We identify document-side early compression as a failure mode in long-document dense retrieval and introduce EDI to quantify it.
- We propose DICE, a training-free document-side remedy that preserves the standard one-query-one-document retrieval interface.
- Across four backbones and two benchmarks, we show that DICE improves long-document retrieval and that these gains coincide with systematically lower evidence dilution.

2 Evidence Dilution in Long-Document Retrieval

2.1 Document-Side Early Compression

Single-vector document encoding fails when relevance is localized. Given a query q and a long document d, a dense retrieval encoder $f(\cdot)$ must compress the entire document into one vector before retrieval begins. Documents are then ranked by cosine similarity:

$$s(q, d) = \mathrm{sim}(f(q), f(d)), \mathbf{d}_{\text{single}} = f(d). \quad (1)$$

When d contains a short evidence span e embedded in a large volume of non-relevant context c, the

encoding $f(e, c)$ is pulled away from the representation of the decisive evidence before the similarity score is ever computed. The document may contain the answer and still rank poorly, because the document-level vector under-represents the short local span that actually resolves the query. Modern encoders routinely process 4k–32k tokens, so this is not primarily a failure of context-window capacity. Rather, it is a failure of representation: $\mathbf{d}_{\text{single}}$ can lie far from the query in embedding space even though the gold document contains the answer-bearing evidence. Figure 1 gives the motivating intuition. The analytical question is therefore not merely whether relevant evidence exists in the document, but whether single-vector compression preserves that evidence strongly enough to influence ranking.

2.2 From Local Evidence to EDI

To make this failure mode measurable, we compare document-level representations against chunk-level evidence inside the same gold document. For a gold document split into M chunks, let a_j be the query–chunk similarity, $m = \max_j a_j$ the strongest local evidence, and $\bar{a}$ the mean chunk similarity. The max-over-chunks score is used here only as a local-evidence oracle for analysis; it is not intended as a deployment-equivalent baseline.

Evidence Concentration (EC). $\text{EC} = \dfrac{m - \bar{a}}{|m| + \epsilon}$. EC captures whether query relevance is sharply localized in a few chunks (high EC) or spread more uniformly across the document (low EC). This tells us whether a sample is structurally susceptible to evidence dilution.

Evidence Dilution Index (EDI). $\text{EDI}(E) = \dfrac{m - s_E}{m - \bar{a} + \epsilon}$, where $s_E = \text{sim}(q, \mathbf{d}_E)$ is the document-level similarity score under encoding method E. EDI measures how far the document vector falls below the strongest chunk-level evidence. Lower EDI indicates that the document representation remains closer to the most relevant local span; negative EDI means the document vector exceeds the single-chunk oracle by integrating signal across multiple chunks.

Both metrics use the evidence margin $m - \bar{a}$ as a normalizer. When this margin is near zero, the ratio can become unstable; this occurs most often in synthetic settings where nearly all chunks are irrelevant to the query. We therefore filter samples with $m - \bar{a} < 0.01$ and report median statistics throughout.

2.3 Implications for Document Encoding

This framework turns early compression from an intuition into a concrete design target. If a single document vector systematically falls below the strongest chunk-level evidence, then a natural remedy is to delay compression: preserve local evidence first, and aggregate only afterward. DICE follows exactly this strategy. It does not alter the query encoder or the retrieval interface; it changes only how the document vector is constructed so that decisive local evidence is less likely to be washed out before ranking.

3 Method

Guided by the evidence-dilution framework in Section 2, we now describe DICE.

3.1 DICE Document Encoding

DICE addresses the problem at the document encoding stage. The key idea is to delay compression: instead of encoding the whole document at once, we encode it in chunks and aggregate only afterward.

Chunking. Given a document d, we tokenize it and split the token sequence into chunks of size k with optional overlap o. The split is performed in token space, preserving token identities and boundaries. Unless otherwise noted, we use non-overlapping chunks ($o = 0$) and vary k to study chunk granularity.

Local position encoding. Each chunk $d^{(j)}$ is encoded independently by the frozen encoder f. Position indices are reset to start from zero within each chunk, rather than being inherited from the original document:

$$\mathbf{h}_j = f(\mathbf{x}^{(j)}, \mathbf{p}^{(j)}), \tag{2}$$

where $\mathbf{x}^{(j)}$ are the token ids of chunk $d^{(j)}$ and $\mathbf{p}^{(j)}$ is the local position sequence that starts from zero within each chunk. Local positions ensure that each chunk is processed as a self-contained context window, unaffected by its offset in the source document. Query encoding is unchanged because our goal is to isolate document-side compression: queries are short relative to documents, so chunking them would add complexity on the query side while changing the retrieval interface we are trying to preserve.

Aggregation. Chunk embeddings are fused into one document vector through a query-independent aggregation function g:

$$\mathbf{d}_{\text{DICE}} = g(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_m). \tag{3}$$

Aggregation is performed before the query is seen, so it cannot use query–chunk interactions. We therefore focus on a small family of simple, query-independent pooling rules that span the main design choices in this setting: preserving all chunks equally, emphasizing high-activation chunks, or selecting only a subset of chunks. Specifically, we study mean pooling, max pooling, and top-k pooling by embedding norm. The default strategy is mean pooling, $g_{\text{mean}} = \frac{1}{m}\sum_{j=1}^{m} \mathbf{h}_j$, which preserves evidence from every chunk equally. Max pooling, $g_{\max}$, tests a more selective alternative by taking the elementwise maximum across chunks. The norm-based variant uses $\|\mathbf{h}_j\|_2$ as a query-independent proxy for chunk salience: g_{topk} averages the top-k highest-norm chunks. All variants are query-independent by construction and are evaluated as ablations in Section 4.3.

Retrieval interface. After aggregation, retrieval proceeds exactly as in the single-vector baseline: $s(q, d) = \text{sim}(f(q), \mathbf{d}_{\text{DICE}})$. DICE does not retrieve chunks, increase the number of corpus entries, or add a second-stage scorer. The change is entirely on the document encoding side, and any difference in ranking comes only from the document representation.

4 Experiments

4.1 Setup

Tasks and Metrics. Our main benchmark is LongEmbed (Zhu et al., 2024), which contains two synthetic tasks, Passkey and Needle, and four real retrieval tasks, NarrativeQA, QMSum, SummScreenFD, and WikiMQA. We report Hit@1 (%) for Passkey and Needle, split into contexts up to 4k tokens and beyond 4k tokens. We report nDCG@10 for the real tasks. The split metrics are macro-averages over the corresponding LongEmbed length slices.

All evaluations use the MTEB framework (Muennighoff et al., 2023). We additionally evaluate Dream on FollowIR (Weller et al., 2024) (News21InstructionRetrieval, Core17InstructionRetrieval, Robust04InstructionRetrieval), using nDCG@5 for News21 and MAP@1000 for Core17 and Robust04, with p-MRR as an auxiliary metric.

Comparison scope. Our main comparisons focus on *deployment-equivalent* settings: one stored vector per document, unchanged query encoding, and the same first-stage query–document scoring interface. Under this scope, SINGLE and the alternative chunk-aggregation rules in Section 4.3 are the direct baselines. We discuss passage retrieval, ColBERT-style late interaction, and late chunking as adjacent references rather than like-for-like competitors, since they change the retrieval unit, the scoring mechanism, or both.

Models. The primary backbone is a Dream-based diffusion embedder (Zhang et al., 2025), built on the Dream 7B language model (Ye et al., 2025). To test whether the effect is specific to diffusion architectures, we also evaluate three autoregressive backbones converted to bidirectional embedders (BehnamGhader et al., 2024): Llama3 8B (Grattafiori et al., 2024), Mistral 7B (Jiang et al., 2023), and Qwen2.5 7B (Yang et al., 2024). In all comparisons, the encoder weights are frozen between the SINGLE and DICE settings.

Implementation. DICE is implemented as document-side logic within a unified evaluation wrapper, built on an extended version of the LLM2Vec codebase (BehnamGhader et al., 2024) that supports both diffusion and autoregressive backbones. Query inputs are routed through ordinary encoding; document inputs are routed through either single-vector encoding or chunk aggregation depending on the configuration. In the chunked path, document windows are formed directly from token ids and each window is encoded with local positional indices. Unless otherwise stated, DICE uses chunk size 1024, no overlap, mean aggregation, bfloat16 inference, and cosine similarity. All evaluation runs use a single NVIDIA A6000 GPU with matched SINGLE and DICE settings.

4.2 Main Results

Table 1 reports the core result: across four backbones, replacing single-vector document encoding with DICE improves LongEmbed, with the largest gains concentrated on the hard long-context synthetic slices, exactly where the early-compression hypothesis predicts. For Dream, Passkey >4k increases by 60.0 points and Needle >4k by 50.7 points. The same trend holds across architectures:

Table 1: Main LongEmbed results. DICE uses chunk size 1024 with mean aggregation. Avg. is the unweighted mean over eight metrics, including the two omitted synthetic ≤4k columns.

Backbone	Method	Avg.	Synthetic (Hit@1, %)		Real (nDCG@10)			
			Pk>4k	Nd>4k	NQA	QMS	SFD	WQA
Llama3	SINGLE (Zhang et al., 2025)	34.64	3.33	7.33	18.06	**29.24**	77.45	40.08
	DICE (*ours*)	**50.56**	**35.33**	**25.33**	**45.64**	27.59	**93.00**	**61.16**
	Δ	+15.92	+32.00	+18.00	+27.58	-1.64	+15.55	+21.08
Qwen	SINGLE (Zhang et al., 2025)	43.80	8.67	9.33	19.51	35.82	87.89	51.95
	DICE (*ours*)	**59.06**	**66.67**	**20.00**	**35.25**	**42.06**	**91.83**	**61.51**
	Δ	+15.27	+58.00	+10.67	+15.75	+6.24	+3.94	+9.56
Mistral	SINGLE (Zhang et al., 2025)	57.24	30.00	18.00	33.10	43.50	92.64	61.08
	DICE (*ours*)	**74.38**	**86.00**	**58.67**	**55.73**	**50.79**	**96.48**	**71.38**
	Δ	+17.14	+56.00	+40.67	+22.63	+7.29	+3.84	+10.31
Dream	SINGLE (Zhang et al., 2025)	63.41	30.00	23.33	43.63	43.93	97.77	79.43
	DICE (*ours*)	**81.92**	**90.00**	**74.00**	**65.01**	**50.54**	**98.53**	**86.04**
	Δ	+18.50	+60.00	+50.67	+21.39	+6.61	+0.76	+6.61

Table 2: Dream ablation on LongEmbed. The top block varies chunk size with mean aggregation, and the bottom block fixes chunk size to 1024 while varying the aggregation rule. "—" denotes the SINGLE baseline, and bold indicates the best per-column value.

Chunk	Agg.	Avg	Synthetic (Hit@1, %)				Real (nDCG@10)			
			Pk≤4k	Pk>4k	Nd≤4k	Nd>4k	NQA	QMS	SFD	WQA
—	—	63.41	**100.0**	30.00	89.20	23.33	43.63	43.93	97.77	79.43
128	mean	61.21	48.80	33.33	71.20	56.00	60.41	44.98	97.74	77.22
256	mean	69.83	75.20	34.00	84.80	**74.67**	63.09	47.43	98.36	81.12
512	mean	76.30	88.80	66.67	84.40	74.00	64.17	49.60	**98.66**	84.07
1024	mean	**81.92**	98.80	90.00	**92.40**	74.00	**65.01**	**50.54**	98.53	**86.04**
1024	max	64.63	99.20	**100.0**	89.20	44.67	26.65	25.39	88.40	43.53
1024	topk	56.65	86.80	9.33	88.00	18.00	49.60	39.99	95.39	66.09

Mistral and Qwen show similarly large Passkey >4k gains, while Llama3 benefits most on NarrativeQA, SummScreenFD, and WikiMQA. The few regressions are small and occur where documents are shorter or relevance is less localized.

As a Dream-side adjacent reference, we additionally test an adapted LATECHUNK-DOC variant based on Late Chunking (Gunther et al., 2024). On Dream, LATECHUNK-DOC reaches 81.51 average points, substantially above direct single-vector encoding (63.41) and close to DICE (81.92). We therefore treat this result as a supplementary design reference rather than a primary baseline (Appendix C.1).

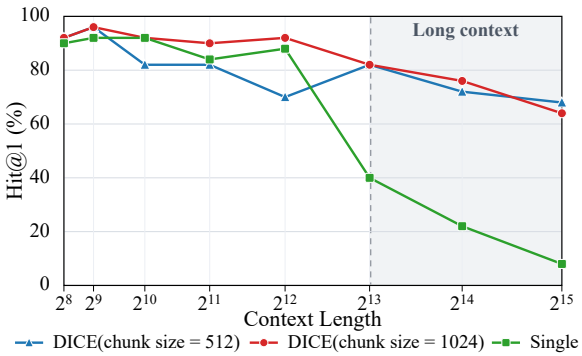

Figure 3: Needle Hit@1 (%) by context length on Dream. Chunk-level encoding prevents the sharp long-context degradation observed under single-vector document compression.

4.3 Analysis

Length-Stratified Needle. Figure 3 breaks Needle Hit@1 (%) down by context length for Dream. At 4k tokens, the single-vector encoder matches the chunked variants; by 32k, it collapses to 0.08 while chunk-512 and chunk-1024 both remain above 0.64. The gap widens monotonically beyond 4k, indicating that early compression, rather than context-window capacity alone, is a major bottleneck in long-document retrieval.

Ablation Analysis. Table 2 reports a full ablation of the two design dimensions in DICE, chunk granularity and aggregation strategy, on the Dream

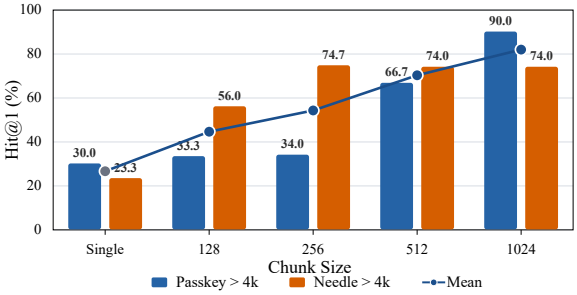

Figure 4: Dream chunk-size ablation on LongEmbed. Bars show Passkey >4k and Needle >4k Hit@1 (%), and the line shows their mean. Larger chunk sizes consistently improve the hardest long-context slices, with chunk size 1024 giving the highest hard-slice mean.

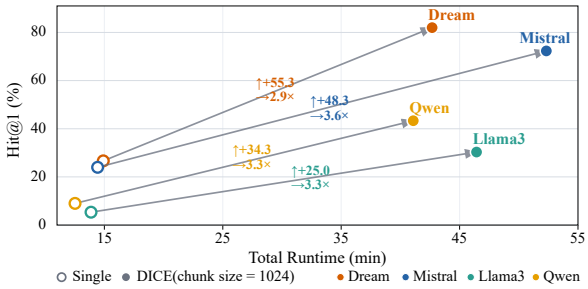

Figure 5: Quality-cost trade-off on LongEmbed. Quality is the mean of Passkey >4k and Needle >4k Hit@1 (%), and arrows point from SINGLE to DICE for each backbone.

backbone. All runs use non-overlapping chunks, bfloat16 inference, and cosine similarity; only the ablated variable changes.

Two clear patterns emerge. First, **chunk granularity matters decisively**. At 128 tokens, DICE underperforms the SINGLE baseline on average (61.21 vs. 63.41), suggesting that overly small chunks fail to preserve enough useful local context. From 256 upward, the overall average rises monotonically, while the hard long-context slices (Passkey >4k and Needle >4k) improve even more sharply. Figure 4 visualizes these two hardest slices together with their mean, while Figure 3 shows the corresponding length-stratified Needle breakdown. Chunk size 1024 achieves the best average (81.92) and the best or near-best score on seven of eight metrics. The same ordering largely persists under Hit@n for $n \in \{2, 3, 5, 10\}$ on the synthetic tasks (Appendix B.2) and on Mistral (avg. rises from 52.77 at chunk-128 to 74.38 at chunk-1024; Appendix B.1).

Second, **more sophisticated aggregation is not better**. Max pooling reaches 100.0 on Passkey >4k, the highest of any setting, but collapses on real tasks: NQA drops from 65.01 to 26.65, QMS from 50.54 to 25.39, and WQA from 86.04 to 43.53. The norm-based variant (topk) also underperforms mean pooling across all metrics and often falls below even the SINGLE baseline. The gains therefore come from preserving enough local context and aggregating conservatively, not from sophisticated chunk selection. We use mean pooling and chunk size 1024 as the default throughout the main results.

We also probed an alternative position scheme that preserves each chunk's absolute document offset instead of resetting positions locally. Across the completed backbones, this variant never improved over local reset: Avg. drops from 81.92 to 81.85 on Dream, from 50.56 to 43.19 on Llama3, and from 74.38 to 73.74 on Mistral (Appendix C.2). We therefore keep local reset as the default. We do not report Qwen for this probe because absolute offsets pushed chunk-level position ids beyond the backbone's supported range on very long inputs, exposing a practical robustness weakness that local reset avoids.

4.4 Deployment Trade-offs

The analyses above establish the main mechanism claim. We now turn to two secondary deployment considerations that do not change that interpretation but matter for practical use: overlap and document-side encoding cost.

Effect of Overlap. Introducing token overlap between adjacent chunks (chunk size 1024) does not improve average performance over the non-overlapping baseline. With 256-token overlap, the average score drops from 81.92 to 70.79, largely because Passkey >4k falls from 90.0 to 46.7. Increasing overlap to 512 tokens partially recovers the average (80.78) and improves Passkey >4k to 94.0 and Needle >4k to 79.3, but at a steep runtime cost (80 minutes for the full LongEmbed suite vs. 43 minutes without overlap). Since overlap increases document encoding time without a consistent average gain, we keep $o = 0$ in the main setting.

Quality-Cost Trade-off. DICE is not an efficiency method: encoding multiple chunks per document increases document-side computation.

Figure 5 plots the accuracy-runtime trade-off across backbones. The arrows show that DICE consistently moves models toward higher long-

Table 3: EDI comparison on the robust subset of LongEmbed on Dream with chunk size 1024 and mean aggregation. The robust subset consists of all samples with $m - \bar{a} \geq 0.01$, and DBR denotes the fraction of samples where EDI(DICE) $<$ EDI(SINGLE).

Split	SINGLE EDI ↓	DICE EDI ↓	DBR (%) ↑
Overall	0.055	−0.709	92.8%
NQA	0.046	−0.737	95.7%
QMS	−0.131	−1.020	94.7%
SFD	−0.009	−0.693	92.7%
WQA	0.286	0.205	61.0%

Table 4: FollowIR results on Dream. Improvements persist beyond LongEmbed, but the best chunk configuration remains task dependent.

Method	News nDCG@5	Core MAP@1000	Robust MAP@1000
SINGLE	36.66	24.87	25.81
DICE-256	39.03	29.02	29.18
DICE-256+ov32	**42.25**	29.61	29.79
DICE-256+ov128	38.80	**31.14**	**30.27**
DICE-512	35.42	25.74	27.93

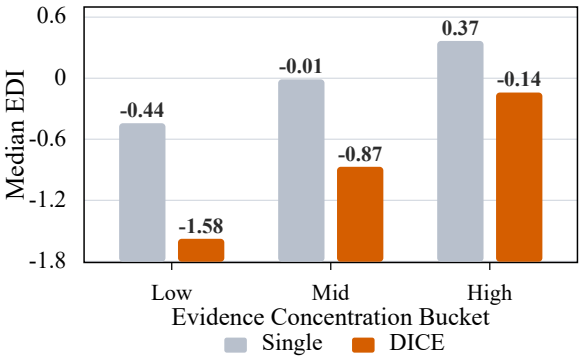

Figure 6: Median EDI across EC terciles on Dream for the LongEmbed robust subset. EDI rises with EC for both methods, but DICE remains consistently lower across all three buckets. The relative gap between SINGLE and DICE is largest in the low-EC bucket.

context Hit@1 (%) at roughly three to four times the document-side runtime. The practical operating point depends on whether long-document recall justifies the added encoding cost, a trade-off that may be acceptable when documents are indexed offline and queried many times.

4.5 EDI-Based Validation

The ablations in Section 4.3 establish that chunk granularity and conservative aggregation drive DICE's gains, but they do not directly test our central hypothesis: that single-vector encoding systematically dilutes localized evidence. Using the EDI framework from Section 2.2, we now test that hypothesis directly. All analyses in this subsection use the Dream backbone with chunk size 1024 and mean aggregation. Here, the *robust subset* means all LongEmbed samples that survive the margin filter from Section 2.2, i.e., $m - \bar{a} \geq 0.01$; for Dream, this leaves 12,779 samples in total. We report the overall comparison on this full subset, and then break out the four real tasks separately.

Table 3 reports both the overall comparison and the real-task breakdown. Overall, DICE reduces median EDI from 0.055 to -0.709 and yields lower EDI than SINGLE in 92.8% of samples. The same pattern holds strongly on NarrativeQA, QMSum, and SummScreenFD, where median EDI drops substantially and DBR remains above 92%. WikiMQA shows a smaller but still positive reduction (0.286 to 0.205), indicating that the effect is not perfectly uniform across tasks but remains directionally consistent. A lightweight Mistral check in Appendix A.2 shows the same qualitative pattern, suggesting that EDI is not merely a Dream-specific diagnostic.

Figure 6 further decomposes the result by EC tercile (low ≤ 0.085, mid 0.085–0.126, high > 0.126; about 4,260 samples each). EDI rises with EC for both methods, indicating that sharply localized evidence is intrinsically harder to preserve in a single vector. At the same time, DICE remains better in every bucket (low: -1.61 vs. -0.48; mid: -0.91 vs. -0.04; high: -0.16 vs. 0.35).

This breakdown sharpens the mechanism interpretation in two ways. First, it matches the length-stratified degradation observed earlier on the hardest long-context slices: as relevance becomes both more localized and more deeply buried, single-vector compression fails more often. Second, the relative gap is largest in the low-EC bucket, indicating that DICE helps not only when one chunk dominates, but also when useful evidence is distributed across multiple chunks.

4.6 Transfer to FollowIR

As a secondary transfer check, Table 4 evaluates Dream on FollowIR. Document-side chunk aggregation consistently improves over the single-vector baseline, reaching 42.25 nDCG@5 on News21 (vs. 36.66), 31.14 MAP@1000 on Core17 (vs. 24.87), and 30.27 MAP@1000 on Robust04 (vs. 25.81). Unlike LongEmbed, however, the best FollowIR setting is not a single no-overlap chunk size: News21 benefits most from 256-token chunks with 32-token overlap, while Core17 and Robust04 pre-

fer a larger 128-token overlap at the same chunk size. We therefore treat FollowIR as transfer evidence that the gains extend beyond explicitly long-document benchmarks, while highlighting that the best granularity remains benchmark dependent.

Taken together, the most reliable gains from DICE occur when documents are long and relevance is localized, and they come from preserving enough local context before document-level compression rather than from sophisticated chunk selection.

5 Related Work

Dense retrieval and the single-vector bottleneck. Dense retrieval has steadily improved through stronger encoders while preserving a stable interface: one vector per retrieval unit. This trajectory runs from Sentence-BERT (Reimers and Gurevych, 2019) and DPR (Karpukhin et al., 2020), through contrastive embedders such as E5 (Wang et al., 2022), Contriever (Izacard et al., 2021), BGE (Xiao et al., 2023), and GTE (Li et al., 2023), to LLM-based encoders such as LLM2Vec (BehnamGhader et al., 2024), NV-Embed (Lee et al., 2025), M3-Embedding (Chen et al., 2024), GritLM (Muennighoff et al., 2025), and recent diffusion-based embedders (Zhang et al., 2025; Eslami et al., 2026). The downside is that the same interface creates a bottleneck for long documents. Long-context studies show that even models with extended context windows under-use information distributed across long sequences (Liu et al., 2024; Hsieh et al., 2024; Li et al., 2024), and LongEmbed (Zhu et al., 2024) makes this failure visible through length-stratified retrieval evaluation. DICE departs from this encoder-centric line of work by leaving the encoder frozen and changing only how the document vector is computed at inference time.

Multi-vector and late-interaction methods. A parallel line of work relaxes the single-vector constraint. Passage retrieval, central to RAG pipelines (Lewis et al., 2020), splits documents into chunks before indexing and retrieves them independently, trading the clean one-vector interface for downstream synthesis complexity. Multi-vector late-interaction methods (ColBERT (Khattab and Zaharia, 2020), ColBERTv2 (Santhanam et al., 2022)) retain token-level representations and compute fine-grained query–document alignment, improving fidelity at the cost of a more complex indexing and scoring pipeline. These methods demonstrate that preserving per-token or per-chunk evidence improves retrieval on long documents, but they require changes to both the indexing infrastructure and the retrieval interface. DICE borrows the intuition that fine-grained representations help, but preserves the standard one-query-one-document interface by aggregating chunk evidence back into a single document vector before retrieval.

Chunk-level and segment-level representations. The closest line of work operates at the chunk or segment level. Late chunking (Gunther et al., 2024) contextualizes token embeddings through a long-context forward pass before pooling them into chunk representations for chunk-level retrieval. DICE shares the premise that evidence should be captured before global compression, but differs in both retrieval granularity and encoding path: it returns a single document vector and encodes chunks independently with local positions. SeDR (Chen et al., 2022) similarly models long documents at the segment level, but requires training a dedicated encoder, whereas DICE is training-free. More recently, Bhat et al. (2025) show that chunk granularity substantially affects passage retrieval quality. Our ablations extend this observation to document-level representations, where the benefit of larger chunks persists when aggregation is kept query-independent and conservative. Learned compression of multi-vector representations into a single vector remains a promising next step, but it would move beyond the training-free setting studied here.

6 Conclusion

We framed long-document retrieval through the lens of document-side early compression and introduced EDI to measure it directly. Within that framework, DICE provides a training-free document-side remedy that preserves the standard single-vector retrieval interface while markedly improving retrieval when localized evidence would otherwise be diluted by document-level compression. Our ablations show that the central design choice is to preserve enough local context before aggregation, and that simple mean pooling is the most robust query-independent rule in this setting. More broadly, the results identify document-side encoding as a practical and underexplored degree of freedom for improving long-document retrieval.

Limitations

DICE increases document-side encoding cost by roughly 3–4×, a trade-off most acceptable when documents are indexed offline. The method aggregates chunk representations without seeing the query, which preserves the single-vector interface but foregoes the chance to dynamically emphasize query-relevant chunks. Finally, DICE still compresses each document into one vector, so it necessarily loses finer-grained evidence compared with multi-vector or passage-level retrieval. These three constraints (encoding cost, query-independent aggregation, and the single-vector ceiling) define the operating point on the accuracy–efficiency frontier that DICE occupies.

Ethical Statement

This work does not involve human subjects, personally identifiable information, or private user data. All experiments are conducted on publicly available benchmark datasets and publicly released models, and we follow their licenses and terms of use. Our method is a training-free change to document encoding for retrieval and is intended to improve the reliability of long-document retrieval systems. However, better retrieval does not guarantee correct downstream decisions, and the best chunk configuration may not transfer uniformly across domains. We therefore recommend task-specific validation before deployment.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *ArXiv*, abs/2404.05961.

Sinchana Ramakanth Bhat, Max Rudat, Jannis Spiekermann, and Nicolas Flores-Herr. 2025. Rethinking chunk size for long-document retrieval: A multi-dataset analysis. *ArXiv*, abs/2505.21700.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Annual Meeting of the Association for Computational Linguistics*.

Junying Chen, Qingcai Chen, Dongfang Li, and Yutao Huang. 2022. Sedr: Segment representation learning for long documents dense retrieval. *ArXiv*, abs/2211.10841.

Sedigheh Eslami, M. V. Gaiduk, Markus Krimmel, Louis Milliken, Bo Wang, and Denis A. Bykov. 2026. Diffusion-pretrained dense and contextual embeddings. *ArXiv*, abs/2602.11151.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Michael Gunther, Isabelle Mohr, Bo Wang, and Han Xiao. 2024. Late chunking: Contextual chunk embeddings using long-context embedding models. *ArXiv*, abs/2409.04701.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *ArXiv*, abs/2404.06654.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Nv-embed: Improved techniques for training llms as generalist embedding models. In *International Conference on Learning Representations*, volume 2025, pages 79310–79333.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020.

Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024. LooGLE: Can long-context language models understand long contexts? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16304–16333, Bangkok, Thailand. Association for Computational Linguistics.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *ArXiv*, abs/2308.03281.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Niklas Muennighoff, Hongjin SU, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative representational instruction tuning. In *International Conference on Learning Representations*, volume 2025, pages 45544–45613.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv*, abs/2212.03533.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn J. Lawrie, and Luca Soldaini. 2024. Followir: Evaluating and teaching information retrieval models to follow instructions. In *North American Chapter of the Association for Computational Linguistics*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian yun Nie. 2023. C-pack: Packed resources for general chinese embeddings. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b: Diffusion large language models. *ArXiv*, abs/2508.15487.

Siyue Zhang, Yilun Zhao, Liyuan Geng, Arman Cohan, Anh Tuan Luu, and Chen Zhao. 2025. Diffusion vs. autoregressive language models: A text embedding perspective. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4273–4303, Suzhou, China. Association for Computational Linguistics.

Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2024. LongEmbed: Extending embedding models for long context retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 802–816, Miami, Florida, USA. Association for Computational Linguistics.

Appendix A. Supplementary Mechanism Evidence

This appendix strengthens the mechanism claim behind DICE. Whereas the main paper establishes the overall pattern quantitatively, the materials here answer two narrower questions: what evidence dilution looks like in a concrete retrieval example, and whether the EDI trend survives beyond the Dream backbone. The appendix is therefore meant to make the argument more interpretable.

A.1 QMSum Case

Figure 7 gives a concrete example of the failure mode discussed in Section 2. The point is not that QMSum is special, but that the decisive evidence is visibly localized: a small set of short phrases resolves the query, yet the full-document SINGLE representation ranks the gold document at 128. By contrast, DICE(chunk size = 1024) recovers rank 1,

Table 5: Mistral chunk-size sweep on LongEmbed. Synthetic scores are Hit@1 (%), and real-task scores are nDCG@10. The same chunk-granularity trend observed on Dream remains visible on Mistral.

Chunk	Avg	Pk≤4k	Pk>4k	Nd≤4k	Nd>4k	NQA	QMS	SFD	WQA
— (SINGLE)	57.24	99.20	30.00	80.40	18.00	33.10	43.50	92.64	61.08
128	52.77	37.20	18.00	75.60	42.00	51.58	42.35	91.97	63.48
256	58.66	51.60	29.33	76.00	52.67	53.35	46.06	94.24	66.05
512	65.65	80.00	44.67	79.20	54.67	54.52	48.80	95.43	67.89
1024	74.38	96.00	86.00	80.00	58.67	55.73	50.79	96.48	71.38

consistent with the claim that delaying document-side compression helps preserve concentrated evidence.

Localized Evidence in QMSum

Query. "Industrial Designer thought the meeting was not friendly to the brainstorming. The restriction was not about the atmosphere but related to the actual environment and the limited time for discussion. Besides, the interaction was structured, meaning each individual took charge of one particular task without enough collaboration between each other. Also, communication through email was inefficient."

Localized evidence in the gold document. "the meetings ... are more brainstorming sessions than meetings"; "the room not being ... very friendly"; "the time given also restricts"; "the interactions are very structured"; "each individual is structured to one particular task"; "it just comes back to us so slow in the email"; "they don't support collaboration."

Gold document rank. SINGLE: 128 DICE: 1

Figure 7: A QMSum example with highly localized evidence. A short discussion segment contains the decisive evidence needed to recover the gold document.

A.2 Cross-Backbone EDI Check

To complement the Dream-based main analysis, we report a lightweight Mistral check under the same robust-subset criterion. The purpose is not to replicate every analysis from Section 4.3, but to test whether the same mechanism appears under a different encoder family. Table 5 first verifies that the chunk-size trend itself transfers to Mistral, and Table 6 then asks whether that transfer is accompanied by the same direction of EDI improvement.

Taken together, the two Mistral tables support the same interpretation as the Dream results. Larger chunks again perform better than overly fine chunking, and the best-performing configuration also yields markedly lower EDI and high DBR on every real-task split. The magnitudes differ from Dream, especially on WikiMQA, but the qualitative alignment between better retrieval and lower dilution remains intact.

Table 6: Mistral EDI on the robust subset. Lower EDI and high DBR replicate the main qualitative pattern beyond Dream.

Split	SINGLE ↓	DICE ↓	DBR (%) ↑
Overall	0.404	−0.361	91.8%
NQA	0.415	−0.407	94.6%
QMS	0.311	−0.394	93.1%
SFD	0.478	−0.228	92.2%
WQA	0.489	0.076	87.7%

Appendix B. Supplementary Retrieval Analyses

This appendix extends the retrieval-side analysis behind the main ablation story. Its goal is not to introduce new claims, but to test whether the two most important empirical patterns from Section 4.3 are robust under additional views: the chunk-size trend across backbones, and the long-context ranking improvements beyond Hit@1.

B.1 Mistral Chunk-Size Sweep

Table 5 mirrors the Dream-side chunk-size ablation with a second backbone. Its role is to show that the main chunk-granularity trend remains visible on Mistral rather than being an artifact of a single encoder family. The same ordering appears here as in Dream: chunk-128 is too small on average, performance rises steadily with larger chunks, and chunk-1024 gives the strongest overall result together with the clearest gains on the difficult >4k synthetic slices.

B.2 Synthetic Top-k Retrieval

These results complement Figure 4 by showing that the same ranking trend persists beyond Hit@1 (%). Table 7 summarizes bucket-level Hit@k (%) on the hardest >4k synthetic slices, Figure 8 breaks Hit@5 (%) down by length, and Figures 9 and 10 show how the full Hit@k (%) profile evolves with length for Needle and Passkey, respectively. The improvement persists across broader top-k cutoffs and remains most pronounced in the longest con-

Table 7: Synthetic Hit@k (%) on Dream for the >4k slices. The ranking gains from DICE persist beyond Hit@1 across broader top-k cutoffs.

Method	Passkey >4k					Needle >4k				
	Hit@1	Hit@2	Hit@3	Hit@5	Hit@10	Hit@1	Hit@2	Hit@3	Hit@5	Hit@10
SINGLE	30.00	30.00	30.00	30.00	30.00	23.33	25.33	26.00	26.00	26.67
DICE-128	33.33	40.67	48.67	58.67	73.33	56.00	71.33	78.67	87.33	92.67
DICE-256	34.00	52.67	62.00	78.67	89.33	74.67	88.67	92.67	95.33	100.00
DICE-512	66.67	80.67	90.00	96.00	100.00	74.00	86.00	90.67	96.67	98.67
DICE-1024	90.00	100.00	100.00	100.00	100.00	74.00	88.00	90.67	94.00	98.67

Table 8: Dream adjacent single-vector reference under the same interface: LATECHUNK-DOC. Synthetic scores are Hit@1 (%), and real-task scores are nDCG@10. This probe situates DICE relative to a neighboring design that contextualizes token evidence before compression.

Method	Avg.	Pk≤4k	Pk>4k	Nd≤4k	Nd>4k	NQA	QMS	SFD	WQA
SINGLE (Zhang et al., 2025)	63.41	**100.00**	30.00	89.20	23.33	43.63	43.93	97.77	79.43
LATECHUNK (Gunther et al., 2024)	81.51	96.00	**93.33**	90.00	73.33	**65.05**	50.13	**98.53**	85.69
DICE (*ours*)	**81.92**	98.80	90.00	**92.40**	**74.00**	65.01	**50.54**	**98.53**	**86.04**

texts, which is the pattern expected if chunk evidence is being preserved rather than accidentally re-ranked.

Appendix C. Alternative Design References

This appendix collects supplementary design probes that are adjacent to, but not part of, the main deployment-equivalent comparison. Their role is to sharpen the scope of the main claim: DICE is not presented as the only way to improve long-document retrieval, but as a particularly simple training-free design point within the one-query-one-document setting. The probes below clarify what happens when chunk evidence is contextualized differently or when local positional reset is removed.

C.1 Dream-Side Adjacent References

Table 8 reports one Dream-side adjacent reference under the same single-vector interface. LATECHUNK-DOC contextualizes token evidence through a long-context forward pass before compressing it back into one document vector; it is included here as a neighboring design point rather than a deployment-equivalent primary baseline. The comparison is useful because it separates two ideas that are easy to conflate: delaying compression helps, but doing so through long-context contextualization is not the same design choice as independently encoding local chunks. The results suggest that richer contextualization before compression is indeed a strong neighboring design and can approach the performance of DICE while preserving a single stored vector per document.

C.2 Absolute-Offset Probe

Table 9 compares the default local-reset position scheme with an absolute-offset variant that preserves each chunk's original document offset. This probe asks whether the gains of DICE come mainly from chunking itself or from the specific positional reset used in the default implementation. Across the completed backbones, absolute offsets never improve the average result and sometimes degrade it substantially, especially on Llama3. This makes the interpretation more specific: the benefit is not just that the document is split into pieces, but that each piece is encoded as a self-contained local context rather than as a late fragment in a very long positional range.

Table 9: Absolute-offset probe at chunk size 1024. Synthetic scores are Hit@1 (%), and real-task scores are nDCG@10. Preserving absolute document offsets does not improve over the default local-reset position scheme.

Backbone	Method	Avg.	Pk≤4k	Pk>4k	Nd≤4k	Nd>4k	NQA	QMS	SFD	WQA
Dream	DICE	**81.92**	98.80	90.00	**92.40**	**74.00**	**65.01**	**50.54**	98.53	86.04
	AbsPos	81.85	**99.20**	**90.67**	90.80	**74.00**	64.98	50.28	**98.54**	**86.34**
Llama3	DICE	**50.56**	**66.00**	**35.33**	**50.40**	**25.33**	**45.64**	27.59	**93.00**	**61.16**
	AbsPos	43.19	37.20	18.00	48.40	18.67	30.07	**36.38**	82.94	48.63
Mistral	DICE	**74.38**	96.00	**86.00**	**80.00**	**58.67**	**55.73**	**50.79**	**96.48**	**71.38**
	AbsPos	73.74	**96.40**	85.33	76.80	57.33	55.71	50.72	96.41	71.20

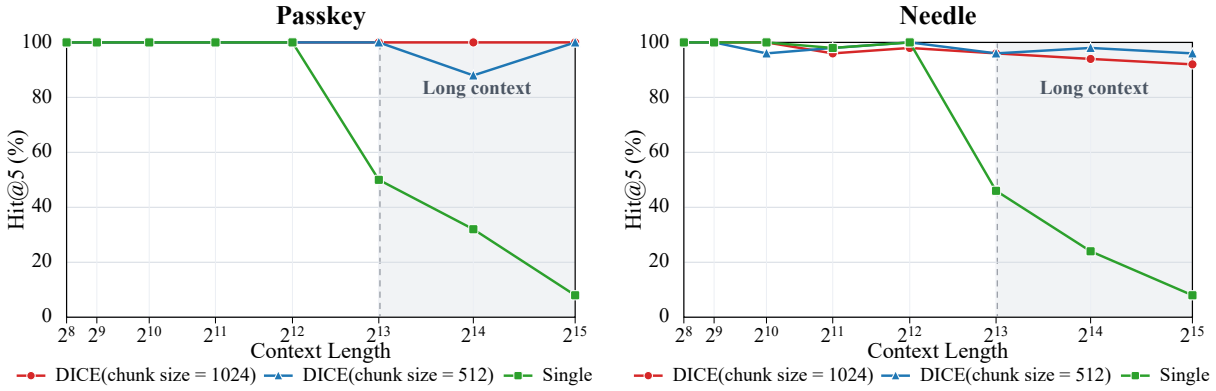

Figure 8: Per-length Hit@5 (%) on synthetic tasks. The advantage of DICE remains concentrated in the longer contexts.

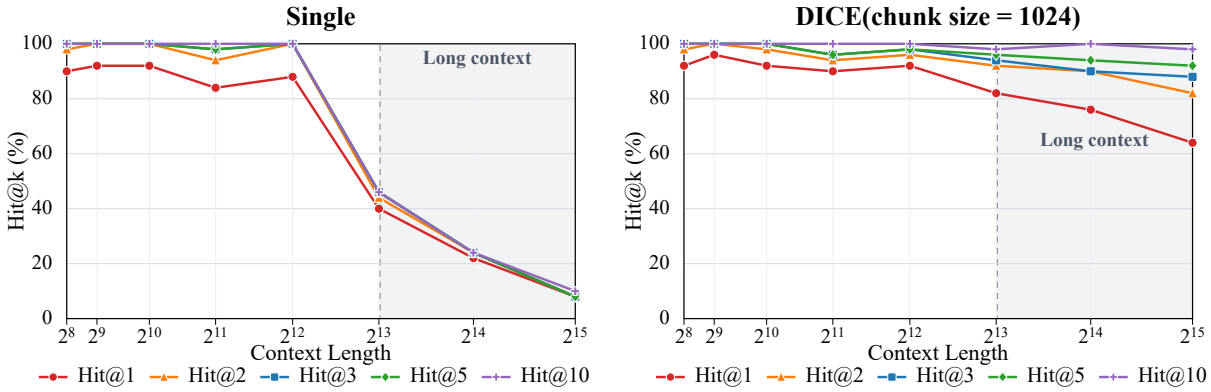

Figure 9: Needle Hit@k (%) by length. The full top-k profile shows that DICE maintains its advantage as context length increases.

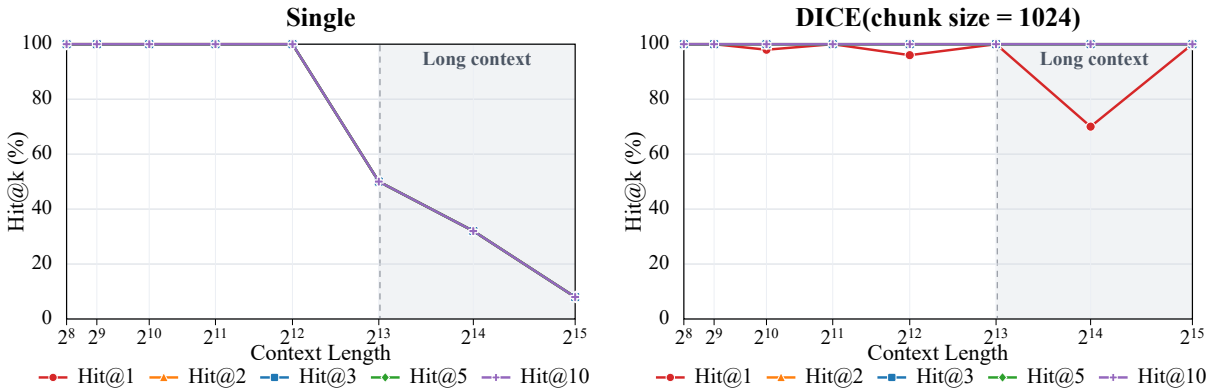

Figure 10: Passkey Hit@k (%) by length. Curve overlap here reflects substantive retrieval behavior rather than a plotting artifact.