FreeChunker: A Cross-Granularity Chunking Framework

Wenxuan Zhang[1], Yuan-Hao Jiang[1], Yang Cao[1], Yonghe Wu[2*]
[1]Shanghai Institute of Artificial Intelligence for Education, East China Normal University, China
[2]Education Technology, East China Normal University, China
✉ 52285901045@stu.ecnu.edu.cn [*]Corresponding author: yhwu@deit.ecnu.edu.cn

Abstract

Chunking strategies significantly impact the effectiveness of Retrieval-Augmented Generation (RAG) systems. Existing methods operate within fixed-granularity paradigms that rely on static boundary identification, limiting their adaptability to diverse query requirements. This paper presents ***FreeChunker***, a Cross-Granularity Encoding Framework that fundamentally transforms the traditional chunking paradigm: the framework treats sentences as atomic units and shifts from static chunk segmentation to flexible retrieval over configurable contiguous multi-granularity candidates. This paradigm shift not only significantly avoids the computational overhead required for semantic boundary detection, but also enhances adaptability to complex queries. Experimental evaluation on LongBench V2 demonstrates that ***FreeChunker*** possesses significant advantages in both retrieval performance and time efficiency compared to existing chunking methods. The pre-trained models and codes are available at https://github.com/mazehart/FreeChunker.

1 Introduction

The rapid advancement of large language models (LLMs) has substantially propelled progress in natural language processing, enabling strong generalization capabilities across diverse tasks (Brown et al., 2020). However, LLMs typically rely on static, parameterized knowledge, often making them less suitable for scenarios requiring up-to-date or domain-specific information. This limitation frequently leads to hallucinated content that may deviate from factual correctness (Huang et al., 2025; Li et al., 2023).

To address this limitation, RAG has emerged as a promising paradigm. By incorporating external knowledge sources, RAG enhances the generation process with more timely and verifiable content, achieving strong performance on knowledge-intensive tasks such as open-domain question answering and fact verification (Lewis et al., 2020). However, the effectiveness of RAG critically depends on the quality of retrieved content, which is substantially influenced by document chunking strategies (Ru et al., 2024). Prior research has demonstrated that chunking granularity often directly affects intra-chunk coherence, semantic coverage, and downstream generation quality. Inappropriate chunking strategies may disrupt semantic structures, potentially leading to information fragmentation or redundancy, ultimately undermining both retrieval and generation performance (Brådland et al., 2025; Lyu et al., 2025).

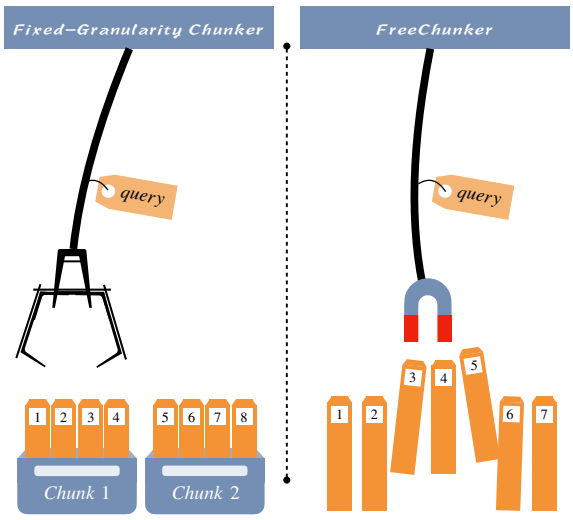

Figure 1: *FreeChunker* breaks the limitation of fixed-granularity text chunking, enabling flexible retrieval over configurable contiguous granularities.

To improve chunking quality, recent efforts have explored semantic-aware chunking techniques. For instance, SemanticChunker (LangChain, 2024) segments documents based on sentence embedding similarity. Meta-Chunking (Zhao et al., 2024) identifies semantic boundaries through perplexity analysis and probability-based segmentation decisions. LumberChunker (Duarte et al., 2024) leverages

Findings of the Association for Computational Linguistics: ACL 2026, pages 14853–14865

LLMs to predict chunk boundaries more aligned with human interpretation. However, existing approaches often lack the flexibility to adapt to diverse query intents or varying deployment constraints. Moreover, some methods rely on computationally expensive large models, where the marginal gains of semantic chunking may not always justify the computational overhead (Qu et al., 2025). Structurally, these approaches still operate within a single-granularity paradigm, consequently limiting their ability to simultaneously accommodate different information needs within the same document collection. Furthermore, the reliance on complex model inference for boundary detection introduces severe latency, rendering them impractical for large-scale real-time retrieval.

To overcome the intrinsic constraints of fixed chunking schemes and the computational inefficiency of existing semantic approaches, ***FreeChunker*** is proposed in this paper—a novel Cross-Granularity Encoding Framework that introduces an alternative chunking paradigm. As illustrated in Figure 1, unlike traditional chunking methods that are typically constrained by fixed text boundaries, the proposed framework enables retrievers to access configurable contiguous sentence spans from documents, thereby providing enhanced flexibility in content retrieval. The core contributions of this paper are summarized as follows:

- **Cross-granularity chunking paradigm**: A novel approach is proposed that breaks the limitation of fixed-granularity text chunking, enabling flexible retrieval over configurable contiguous granularities.

- **Computational efficiency**: The framework treats sentences as atomic units, avoiding time-consuming boundary identification processes. Instead, it utilizes preset granularity combinations, significantly reducing both chunking and encoding time.

- **Superior performance**: The framework achieves the best average retrieval performance by adapting multiple granularity levels to diverse query requirements, consistently outperforming existing baselines across various scenarios on the LongBench V2 benchmark.

2 Related Work

Limitations of Fixed-granularity Chunking As a foundational step in RAG, text chunking substantially influences overall system performance. The most straightforward strategy divides documents into fixed-granularity chunks, which is simple to implement but often misaligns with semantic boundaries, consequently leading to information fragmentation or redundancy (Ru et al., 2024). More critically, fixed chunking forces a single granularity choice across the entire document, thereby preventing the system from simultaneously accessing both fine-grained details and coarse-grained context that different queries may require. This granularity rigidity fundamentally limits the retrieval system's ability to adapt to diverse information needs within a single document collection.

Semantic-aware Chunking Techniques To address the aforementioned semantic boundary issues, recent work has explored more adaptive chunking strategies. SemanticChunker (LangChain, 2024) relies on sentence-level embedding similarity to enhance intra-chunk semantic coherence by identifying natural breakpoints based on semantic similarity thresholds. Meta-Chunking (Zhao et al., 2024) proposes two adaptive segmentation algorithms leveraging large language models' logical awareness: Perplexity Chunking identifies logical boundaries by analyzing inter-sentence perplexity variations, while Margin Sampling Chunking determines boundaries by comparing model probability differences for segmentation decisions. LumberChunker (Duarte et al., 2024) leverages large language models to determine chunk boundaries more aligned with human interpretation. However, despite their semantic awareness, these methods still operate under a single granularity paradigm—each document is ultimately represented at one chosen level of granularity, whether sentence-level, paragraph-level, or custom semantic units.

Adaptive Attempts Inspired by the concept of mixture of experts (MoE) (Jacobs et al., 1991; Shazeer et al., 2017), recent methods like MoC (Zhao et al., 2025) and MoG (Mix-of-Granularity) (Zhong et al., 2025) attempt to enhance adaptability by treating different chunking strategies as expert modules and employing routing mechanisms to dynamically select the most suitable chunker based on query characteristics. However, these MoE-inspired methods essentially aggregate

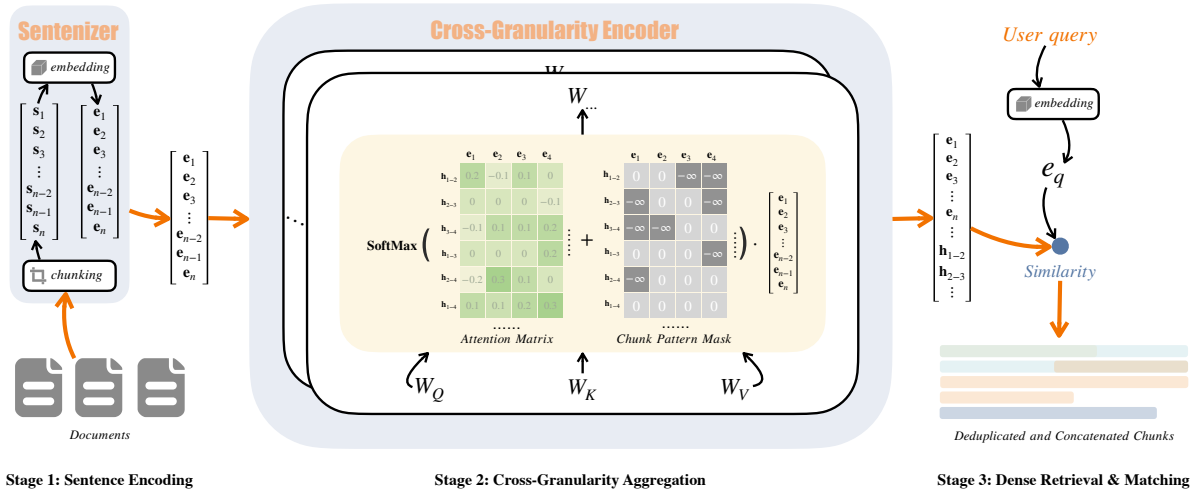

Figure 2: **The architecture of *FreeChunker***. Long text input is processed by the Sentenizer and Embedding Model to produce sentence embeddings. The Attention Matrix and Chunk Pattern Mask work together to generate multiple granularity chunk embeddings in a single forward pass, enabling flexible on-demand access to different chunk sizes.

existing chunkers as experts; consequently, both the quality of semantic chunking and the degree of granularity control remain fundamentally bounded by the capabilities of the underlying chunker experts.

Despite these advances, a fundamental limitation persists: current methods still operate within a single-chunking, single-granularity paradigm—whether fixed, semantics-driven, or query-adaptive. Consequently, retrieval systems cannot simultaneously access multiple levels of information granularity, which is precisely what is needed to better accommodate diverse query intents. Therefore, there remains a critical need for a paradigm that can transcend these single-granularity constraints.

3 Methodology

In *FreeChunker*, sentences are treated as atomic units and chunk embeddings are generated across multiple granularities simultaneously. This approach enables retrieval systems to access any desired chunk size on demand, ranging from fine-grained single sentences to coarse-grained multi-sentence contexts.

Before detailing the mathematical formulation, a high-level intuition of the approach is provided. Unlike traditional methods that physically segment text and encode each chunk separately, *FreeChunker* employs a unified encoding process. Imagine a "mask" overlaid on the document's sentence representations. By adjusting this mask, it is possible to dynamically specify which sentences should be attended to and combined into a chunk embedding. Crucially, by constructing a composite mask that contains patterns for multiple granularities (e.g., 1-sentence, 2-sentence, ..., k-sentence chunks), embeddings for all desired chunk sizes can be generated in a single forward pass of the model. This eliminates the need for redundant computations and allows for flexible, overlapping chunk definitions without repeated encoding.

3.1 Sentenizer

The **Sentenizer** serves as the initial stage of the *FreeChunker* pipeline. Given a document $\mathcal{D}$, it first decomposes it into atomic sentence units $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$. Subsequently, a token-level embedding model $\mathcal{M}$ is employed to encode these sentences into a sentence embedding matrix $\mathcal{E} = [\vec{e_1}, \vec{e_2}, \ldots, \vec{e_n}] \in \mathbb{R}^{n \times d}$, where d denotes the embedding dimension. This process provides the foundational input for the subsequent cross-granularity encoding process.

3.2 Cross-Granularity Chunk Pattern

The core innovation of *FreeChunker* lies in proposing a flexible contiguous multi-granularity paradigm that overcomes the fixed granularity limitation of existing methods by defining multiple chunk patterns, thereby enabling on-demand granularity access.

Based on the n sentences obtained from the Sentenizer, a Chunk Pattern Mask $\mathbf{P} \in \{0, -\infty\}^{m \times n}$ is constructed, where m is the number of desired

chunks. Each row $\mathbf{P}[i,:]$ represents the i-th chunk pattern, and $\mathbf{P}[i,j] = 0$ indicates that sentence s_j belongs to that pattern.

The key insight is that the matrix can encode configurable contiguous granularity combinations. For any granularity g and starting position s, chunk patterns are defined as:

$$\mathbf{P}_{g,s}[i,j] = \begin{cases} 0 & \text{if } s \le j < s+g \\ -\infty & \text{otherwise} \end{cases} \quad (1)$$

Therefore, the framework supports the superposition processing of different granularities. As illustrated in Stage 2 of Figure 2, this results in a mask with a distinctive banded structure, where valid attention weights (0) are clustered around the diagonal for adjacent sentence chunks, while other areas are masked out ($-\infty$).

3.3 Cross-Granularity Encoder

After obtaining the Chunk Pattern Mask $\mathbf{P}$ and sentence embeddings $\mathcal{E}$, independently encoding each chunk would typically cause extensive redundant computation. For any chunk spanning from sentence s_i to sentence s_j, its true embedding is conceptually equivalent to $\vec{e}_{i,j} = \mathcal{M}([s_i, s_{i+1}, \ldots, s_j])$, where $[s_i, s_{i+1}, \ldots, s_j]$ represents the concatenation of sentences.

To efficiently compute these representations, a specialized transformer architecture is modified. For each layer, a learnable chunk embedding $\mathbf{h}_{\text{chk}} \in \mathbb{R}^d$ is introduced that is replicated m times to form $\mathcal{H} = [\mathbf{h}_{\text{chk}}, \mathbf{h}_{\text{chk}}, \ldots, \mathbf{h}_{\text{chk}}]^T \in \mathbb{R}^{m \times d}$. The cross-granularity attention is then computed using the sentence embeddings $\mathcal{E}$ as Keys and Values:

$$\mathbf{Q} = \mathbf{W}_Q \mathcal{H}, \quad \mathbf{K} = \mathbf{W}_K \mathcal{E}, \quad \mathbf{V} = \mathbf{W}_V \mathcal{E} \quad (2)$$

$$\text{Attn}(\mathcal{H}, \mathcal{E}) = \text{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{P}\Big)\mathbf{V} \quad (3)$$

Following the attention mechanism, standard Transformer layer operations are applied, including Layer Normalization and Feed-Forward Networks (FFN), to produce the final chunk embeddings. This design allows parallel generation of all chunk embeddings in a single forward pass, with sentence-level embeddings being reused across different granularities, thereby avoiding redundant encoding of sentences.

3.4 Deduplicate and Concatenate

Because retrieved candidates from different granularities often overlap, a deterministic post-processing strategy is required to recover a coherent context from the top-ranked chunks. Before retrieval, each base sentence S_t is assigned a global sentence index t and stored as [Begin-t] S_t [End-t], so each retrieved chunk can be decomposed into ordered atomic sentence units. During reconstruction, sentence units are merged by global index; if the same index appears in multiple retrieved candidates, the instance from the highest-scoring candidate is retained. The remaining sentences are then sorted in ascending index order and concatenated to form the final context, with "..." inserted whenever two adjacent retained indices are not consecutive. This procedure makes reconstruction reproducible and independent of the order in which overlapping chunks are materialized; an illustrative example is provided in Appendix E.

4 Training Process

4.1 Training Setup

Token-level Embedding Models To train the cross-granularity framework (Section 3) for effective chunk representation, three fundamental token-level embedding models are selected: jina-embeddings-v2-small-en (33M parameters), nomic-embed-text-v1.5 (137M parameters), and BGE-M3 (570M parameters). These models represent different scales and architectures, enabling a thorough assessment across varying embedding capacities.

Dataset and Preprocessing The training dataset comprises 100K documents sampled from The Pile (Gao et al., 2020), each truncated to 8000 tokens and split into sentences. Training pairs $(\vec{e}, \vec{v})$ are constructed across 5 granularities (2, 4, 8, 16, 32 sentences), where $\vec{e}$ is the ground-truth embedding obtained by encoding concatenated sentences using the base models, and $\vec{v}$ denotes the corresponding output from the proposed framework. Granularity 1 used at inference corresponds to the original sentence embedding produced by the Sentenizer and is therefore used directly rather than learned through span composition.

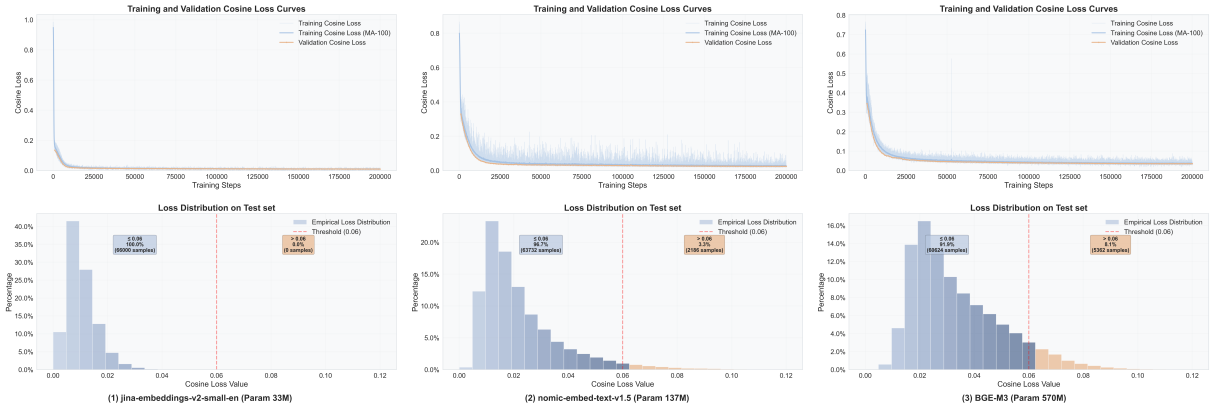

Figure 3: **Training Dynamics and Generalization Across Embedding Models.** Training and validation curves (top) illustrate training loss in blue and validation loss in orange, and test set loss distributions (bottom) show the cosine loss on an in-distribution test set, demonstrating consistent convergence and effective generalization.

Optimization The training objective is formulated to minimize the cosine similarity loss:

$$\mathcal{L} = 1 - \frac{1}{|\mathcal{B}|} \sum_{(\vec{e},\vec{v})\in\mathcal{B}} \cos(\vec{e},\vec{v}) \tag{4}$$

where $\mathcal{B}$ denotes the training batch. The AdamW optimizer is employed with learning rate 1×10^{-4} and batch size 1. Training is conducted for 2 epochs with cosine schedule learning rate decay, incorporating warmup over the first one-third of total training steps. Model validation is performed every 1000 steps on a held-out validation set of 200 samples to monitor generalization performance.

Computational Resources All training experiments are conducted on NVIDIA A800 GPUs with 80GB memory. The total computational cost exceeds 500 GPU hours, encompassing dataset encoding construction and model training processes.

4.2 Training Dynamics and Evaluation

Convergence Analysis The upper panels of Figure 3 present the training and validation cosine loss curves for three base models. All configurations demonstrate smooth and stable convergence within 200K steps, validating the effectiveness of the overall optimization strategy. Notably, the smallest model, jina-embeddings-v2-small-en (33M parameters), exhibits the fastest and cleanest convergence with lower training loss. As model parameters increase, both training and validation losses show degradation across the three models. This phenomenon may be attributed to the use of a fixed 330M-parameter sentence-level encoder for all three base models, suggesting that scaling the sentence-level encoder capacity could potentially improve fitting performance for BGE-M3 (570M parameters).

Test Set Generalization Analysis The lower panels of Figure 3 illustrate the cosine loss distributions on the independent test set. A threshold of 0.06 is employed to distinguish well-fitted samples from under-fitted ones, enabling coarse-grained assessment of chunk embedding accuracy. For jina-embeddings-v2-small-en, 100% of test samples fall below this threshold, with loss distribution highly concentrated in the 0.01–0.02 range, reflecting excellent generalization capability and minimal approximation error. In contrast, nomic-embed-text-v1.5 (137M parameters) exhibits a more dispersed distribution: 96.7% of samples remain below the 0.06 threshold, yet long-tail high-loss samples emerge, indicating increased representation noise or model uncertainty. BGE-M3 (570M parameters) shows further distribution broadening, consistent with the training loss patterns observed.

5 Theoretical Analysis

While Section 4 shows that the cross-granularity encoder fits the target embeddings well ($\rho \approx 1$), it remains necessary to quantify how the residual approximation error affects downstream retrieval.

5.1 Setup and Constraints

Let $\vec{e}_{i,j} = \mathcal{M}([s_i, \ldots, s_j])$ denote the *true* chunk embedding of the concatenated span $[s_i, \ldots, s_j]$, $\vec{v}_{i,j}$ the *approximate* chunk embedding produced by the cross-granularity encoder, and $\vec{q}$ a query embedding. All vectors are assumed to be unit-norm.

The substitution loss is studied as follows:

$$\varepsilon \;:=\; \big|\cos(\vec{q},\vec{e}_{i,j}) - \cos(\vec{q},\vec{v}_{i,j})\big|. \tag{5}$$

Two parameters characterize the geometric configuration:

$$\begin{aligned} \rho &= \cos(\vec{e}_{i,j},\vec{v}_{i,j}), \\ s &= \cos(\vec{q},\vec{e}_{i,j}), \end{aligned} \tag{6}$$

where ρ quantifies substitution quality and s captures query alignment with the true chunk.

5.2 Upper Bound Analysis

Using concentration of measure, a high-probability bound can be derived for high-dimensional embedding spaces (e.g., $d \geq 768$). Under the quasi-random error assumption stated below, the probability that a query aligns with a specific approximation-error direction is exponentially small; empirical statistics under real query distributions are reported in Appendix C.

Step 1: Orthogonal Decomposition Decompose vectors relative to the true chunk embedding $\vec{e}_{i,j}$. Any unit vector $\vec{x}$ can be written as $\vec{x} = (\vec{x}\cdot\vec{e}_{i,j})\vec{e}_{i,j} + \vec{x}_\perp$, where $\vec{x}_\perp \in \vec{e}_{i,j}^{\perp}$.

Since $\vec{v}_{i,j}\cdot\vec{e}_{i,j} = \rho$, its orthogonal component has magnitude $\sqrt{1-\rho^2}$, so

$$\vec{v}_{i,j} = \rho\vec{e}_{i,j} + \sqrt{1-\rho^2}\,\mathbf{u}, \tag{7}$$

where $\mathbf{u} \in \mathbb{R}^d$ is a unit vector in the orthogonal subspace ($\mathbf{u} \perp \vec{e}_{i,j}$).

Similarly, because $\vec{q}\cdot\vec{e}_{i,j} = s$,

$$\vec{q} = s\vec{e}_{i,j} + \sqrt{1-s^2}\,\mathbf{w}, \tag{8}$$

where $\mathbf{w} \in \mathbb{R}^d$ is a unit vector in the orthogonal subspace ($\mathbf{w} \perp \vec{e}_{i,j}$).

Step 2: Algebraic Expansion of Loss Substituting these decompositions into $\vec{q}\cdot\vec{v}_{i,j}$ gives

$$\begin{aligned} \vec{q}\cdot\vec{v}_{i,j} &= (s\vec{e}_{i,j} + \sqrt{1-s^2}\mathbf{w})\cdot(\rho\vec{e}_{i,j} + \sqrt{1-\rho^2}\mathbf{u}) \\ &= s\rho(\vec{e}_{i,j}\cdot\vec{e}_{i,j}) + \sqrt{1-s^2}\sqrt{1-\rho^2}(\mathbf{w}\cdot\mathbf{u}) \\ &= s\rho + \sqrt{1-s^2}\sqrt{1-\rho^2}(\mathbf{w}\cdot\mathbf{u}). \end{aligned} \tag{9}$$

The cross-terms vanish by orthogonality. Substituting into $\varepsilon = |s - \vec{q}\cdot\vec{v}_{i,j}|$ yields

$$\begin{aligned} \varepsilon &= \Big|s - \Big(s\rho + \sqrt{1-s^2}\sqrt{1-\rho^2}(\mathbf{w}\cdot\mathbf{u})\Big)\Big| \\ &= \Big|s(1-\rho) - \sqrt{1-s^2}\sqrt{1-\rho^2}(\mathbf{w}\cdot\mathbf{u})\Big|. \end{aligned} \tag{10}$$

Applying the triangle inequality $|a-b| \leq |a| + |b|$ gives

$$\varepsilon \leq \underbrace{s(1-\rho)}_{\text{Radial Error}} + \sqrt{1-s^2}\sqrt{1-\rho^2}\;\underbrace{|\mathbf{w}\cdot\mathbf{u}|}_{\text{Tangential Noise}}. \tag{11}$$

Step 3: Probabilistic Bound via Concentration The term $|\mathbf{w}\cdot\mathbf{u}|$ measures how strongly the approximation-error direction aligns with the query direction inside the $(d-1)$-dimensional orthogonal subspace.

Assumption (Quasi-Random Error). For an independent query, the direction of the approximation error $\mathbf{u}$ is uniformly distributed on the sphere S^{d-2} relative to $\mathbf{w}$.

This assumption is plausible because $\mathbf{u}$ is induced by aggregation, whereas $\mathbf{w}$ is query-specific; in high-dimensional spaces, accidental alignment is unlikely.

By **Levy's Lemma** (Milman and Schechtman, 1986; Ledoux, 2001), for a random vector $\mathbf{u}$ uniformly distributed on the sphere and a fixed vector $\mathbf{w}$, the probability that $|\mathbf{w}\cdot\mathbf{u}|$ exceeds a threshold t decays exponentially:

$$P(|\mathbf{w}\cdot\mathbf{u}| \geq t) \leq 2\exp\left(-\frac{(d-1)t^2}{2}\right). \tag{12}$$

Let $\delta \in (0,1)$ be a failure probability. Setting the tail bound equal to δ gives

$$\begin{aligned} 2\exp\left(-\frac{(d-1)t^2}{2}\right) &= \delta \\ \Longrightarrow \quad t &= \sqrt{\frac{2\ln(2/\delta)}{d-1}}. \end{aligned} \tag{13}$$

Theorem (Probabilistic Substitution Bound). With probability at least $1-\delta$, the substitution loss is bounded by:

$$\varepsilon \leq s(1-\rho) + \sqrt{1-s^2}\sqrt{1-\rho^2}\cdot\sqrt{\frac{2\ln(2/\delta)}{d-1}}. \tag{14}$$

Implications. As d increases, the tangential term scales as $O(1/\sqrt{d})$ and vanishes, so the loss is dominated by the radial term $s(1-\rho)$. Since $s < 1$ and training drives ρ close to 1, the remaining substitution loss is small. This supports the robustness of *FreeChunker* in high-dimensional embedding spaces.

Embeddings	Traditional		SemanticChunker		PPL Chunking		Margin Sampling		LumberChunker		*FreeChunker*	
	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10
I. Single-Document QA												
jina-small-en	30.86	35.43	33.71	33.71	34.10	32.57	28.57	34.29	32.00	35.43	**35.43**	32.57
nomic-text-v1.5	29.90	29.71	29.71	31.43	32.19	33.71	32.57	31.81	29.52	29.14	**35.62**	32.57
BGE-M3	33.33	37.71	32.57	31.43	33.33	34.29	32.57	33.14	35.81	36.38	**36.57**	34.29
II. Multi-Document QA												
jina-small-en	24.80	28.00	28.80	28.80	28.00	28.27	29.60	27.20	25.87	28.80	26.40	26.40
nomic-text-v1.5	26.40	26.40	22.40	24.00	25.07	27.20	28.00	26.40	25.60	28.80	24.00	28.00
BGE-M3	27.20	28.80	25.87	31.20	27.20	28.00	27.20	29.60	28.80	29.60	28.00	29.60
III. Code Repository Understanding												
jina-small-en	44.00	51.33	36.00	44.00	38.00	48.00	42.00	48.00	44.00	52.67	**46.00**	44.00
nomic-text-v1.5	42.00	46.00	38.00	36.00	35.33	38.00	44.00	42.00	32.00	36.00	34.00	42.00
BGE-M3	42.00	44.00	44.00	36.67	48.00	46.00	40.00	36.00	40.67	38.00	42.00	**48.00**
IV. Long In-context Learning												
jina-small-en	27.57	30.86	29.63	29.63	28.40	28.40	25.93	20.99	32.10	29.63	**32.10**	28.40
nomic-text-v1.5	28.40	28.40	28.40	29.63	26.75	19.75	26.75	22.22	22.63	29.63	20.99	24.69
BGE-M3	26.75	25.10	34.57	30.86	27.57	28.40	32.10	30.45	28.40	25.93	27.16	25.93
V. Long-dialogue History Understanding												
jina-small-en	26.50	33.33	25.64	17.95	23.08	16.24	28.21	25.64	30.77	23.08	28.21	23.08
nomic-text-v1.5	35.04	25.64	30.77	20.51	20.51	17.95	23.08	33.33	23.08	25.64	**38.46**	**38.46**
BGE-M3	23.08	23.08	33.33	29.06	14.53	30.77	25.64	37.61	25.64	20.51	**33.33**	33.33
VI. Long Structured Data Understanding												
jina-small-en	15.15	24.24	25.25	29.29	33.33	33.33	30.30	36.36	26.26	21.21	**42.42**	**45.45**
nomic-text-v1.5	35.35	30.30	27.27	21.21	33.33	37.37	39.39	39.39	12.12	12.12	36.36	33.33
BGE-M3	30.30	38.38	23.23	27.27	33.33	33.33	33.33	30.30	21.21	16.16	**45.45**	**42.42**
VII. Overall												
jina-small-en	28.76	33.53	30.88	31.34	31.15	31.15	29.82	31.21	31.21	32.67	**33.60**	31.61
nomic-text-v1.5	30.75	30.02	28.43	28.23	29.03	29.29	31.34	30.55	26.04	28.43	30.48	**31.61**
BGE-M3	30.62	33.00	31.81	31.35	30.88	32.61	31.41	32.27	31.61	30.62	**33.80**	**33.60**
Average	30.04	32.18	30.37	30.31	30.35	31.01	30.86	31.35	29.62	30.57	**32.63**	**32.27**

Table 1: **Average accuracy (%) of different chunking methods on LongBench V2.** "Overall" denotes the average accuracy across all questions. Standard deviations (SDs) across 3 runs are minimal: 78.7% of accuracy SDs are 0.00%. Specifically, FreeChunker show all accuracy SDs $< 0.50\%$, while others range up to 3.50%. The best and second-best results for ***FreeChunker*** and the **Average** score are highlighted in **bold** and underlined, respectively.

6 Experiments

6.1 Experimental Setup

Datasets Comprehensive experiments are conducted on LongBench V2 (Bai et al., 2024), a challenging benchmark specifically designed for long-context understanding tasks. This dataset is suitable for evaluating chunking methods as it contains documents with extensive contexts and covers diverse task types with varying difficulty levels, thereby providing a realistic testbed for assessing the effectiveness of different chunking strategies in practical RAG scenarios.

Embedding Models To ensure fair comparison, identical text encoding models are utilized across all methods. Three embedding models are employed for comprehensive evaluation: jina-embeddings-v2-small-en (33M parameters), nomic-embed-text-v1.5 (137M parameters), and BGE-M3 (570M parameters). These models typically represent different scales and architectures, thereby enabling thorough assessment of chunking performance across varying embedding capacities.

Generative Model For text generation, Qwen3-8B (Team, 2025) is employed as the backbone language model, deployed via vLLM (Kwon et al., 2023) for efficient inference. To guarantee fair comparison and reproducibility, greedy decoding with temperature set to 0 is employed by Qwen3-8B.

Baseline Methods *FreeChunker* is compared against several representative chunking approaches. **(1) Traditional Chunking** represents the traditional approach that first splits text by periods to preserve semantic integrity, then iteratively accumulates sentences until exceeding predefined token limits (256 tokens); **(2) Semantic Chunking** employs SemanticChunker (LangChain, 2024) to perform embedding similarity-based chunking using sentence embeddings to identify natural breakpoints in the text. The breakpoint thresh-

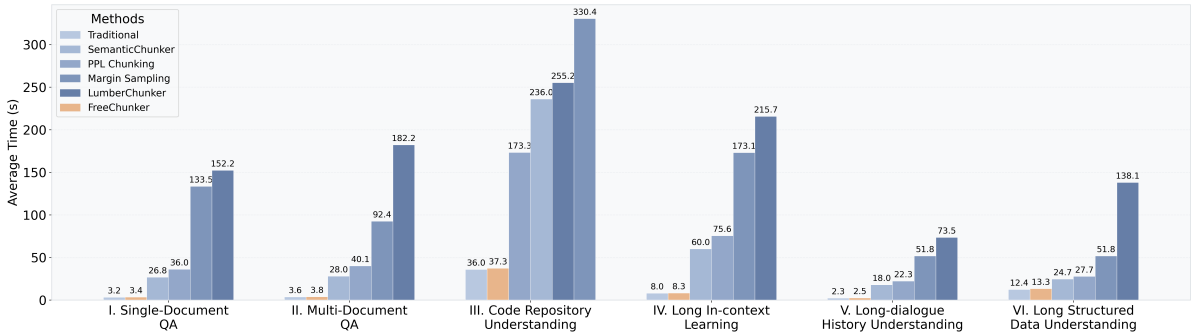

Figure 4: **Average processing time per instance across baselines on LongBench V2.** The processing time comprises the sum of chunking and encoding durations. *FreeChunker* (highlighted in orange) achieves comparable speed to Traditional Chunking and significantly outperforms other methods.

old type is set to "percentile" and the breakpoint threshold amount is set to 50.0; **(3) MetaChunking** (Zhao et al., 2024) employs perplexity-based semantic boundary detection with two submethods: PPL Chunking and Margin Sampling, utilizing Qwen/Qwen2.5-1.5B-Instruct model (Yang et al., 2024) for text chunking. PPL Chunking adopts the threshold of 0.5 as recommended in the original paper, while Margin Sampling uses default settings; **(4) LumberChunker** (Duarte et al., 2024) provides LLM-guided chunk boundary prediction, utilizing the same configuration as the Generative Model.

***FreeChunker* Setting** The Sentenizer module adopts the same sentence-splitting rule as **Traditional Chunking** to obtain atomic sentence units. Unlike the Traditional baseline, these sentence units are not accumulated into fixed 256-token chunks but are directly used as the inputs to the Cross-Granularity Encoder. For the granularity configuration, the simplest combination of three granularities is directly preset: (1, 2, 4). This setup ensures efficient processing while providing multi-scale context coverage. Additional analyses on larger contiguous granularities are reported in Appendix D.

6.2 Main Results

Overall Performance Table 1 presents the comparative results on LongBench V2 across three embedding models. *FreeChunker* achieves the best average performance among all methods.

Ablation Study Since *FreeChunker* utilizes Traditional Chunking as the underlying chunking method for its Sentenizer to segment basic sentence units, and subsequently applies the Cross-Granularity Encoder to obtain cross-granularity embeddings, the performance gap between them can be regarded as an ablation study for the proposed cross-granularity framework. As shown in Table 1, *FreeChunker* consistently outperforms the Traditional baseline, which verifies the superposition effect of multi-granularity embeddings and the effectiveness of the cross-granularity framework.

Time Efficiency Figure 4 illustrates the time cost distribution across different tasks. Compared to Traditional Chunking, *FreeChunker* merely adds the forward propagation encoding time of the Cross-Granularity Encoder, resulting in negligible extra overhead. In contrast, LLM-based methods (e.g., LumberChunker) consume hundreds of seconds. This achieves a speedup of up to 30× compared to complex semantic chunkers, demonstrating that *FreeChunker* effectively circumvents the efficiency bottleneck of semantic-aware chunking without compromising retrieval quality.

7 Conclusion

This paper introduces *FreeChunker*, a novel cross-granularity chunking framework that fundamentally transforms the traditional text chunking paradigm. By shifting from "static boundary identification and re-encoding" to "flexible semantic combination of sentence encodings," this framework significantly reduces the computational overhead of semantic analysis required for boundary detection, while enhancing adaptability to diverse query requirements. Experimental results obtained using the simplest preset chunk patterns demonstrate significant advantages over existing methods, thereby validating the potential of this framework.

Beyond the preset chunk patterns explored in

the current study, the chunk-pattern mechanism $\mathbf{P}_{g,s}[i, j]$ could in principle be extended to more complex non-contiguous compositions. However, defining and validating such candidate spaces requires additional task-specific constraints and is left for future work. The present study therefore focuses on retrieval-ready representations for configurable contiguous multi-granularity candidates.

Limitations

Although ***FreeChunker*** introduces additional encoder-side GPU memory (VRAM) overhead relative to using sentence embeddings alone, this overhead is manageable. When compared with repeated direct concatenated encoding of overlapping multi-sentence candidates on long inputs, *FreeChunker* reduces both VRAM and latency, as reported in Appendix F.

Ethics Statement

This work does not involve human subjects, animal experiments, or the collection of private or sensitive data. All datasets utilized in this research are publicly available and used in accordance with their respective licenses.

During the preparation of this manuscript, the authors used AI-based tools for grammatical refinement and stylistic polishing to improve the clarity of the presentation. Following this process, the authors conducted a thorough manual review and edit of the content to ensure its technical accuracy and original intent. The authors take full responsibility for the final content of the paper.

References

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Henrik Brådland, Morten Goodwin, Per-Arne Andersen, Alexander S. Nossum, and Aditya Gupta. 2025. A new hope: Domain-agnostic automatic evaluation of text chunking. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, pages 1–10, Padua, Italy. ACM. © 2025 Copyright held by the owner/author(s). Licensed under CC BY 4.0.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

André V. Duarte, João Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L. Oliveira. 2024. Lumberchunker: Long-form narrative document segmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 6473–6486. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP '23)*.

LangChain. 2024. Semantic chunker. https://python.langchain.com/docs/how_to/semantic-chunker/. Accessed: 2025-06-06.

Michel Ledoux. 2001. *The concentration of measure phenomenon*. 89. American Mathematical Soc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language

models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6449–6464. Association for Computational Linguistics.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2025. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Transactions on Information Systems*, 43(2):1–32.

Vitali D Milman and Gideon Schechtman. 1986. *Asymptotic theory of finite dimensional normed spaces*. Springer.

Renyi Qu, Ruixuan Tu, and Forrest Sheng Bao. 2025. Is semantic chunking worth the computational cost? In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 2155–2177. Association for Computational Linguistics.

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Qwen Team. 2025. Qwen3 technical report.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Jihao Zhao, Zhiyuan Ji, Jason Zhaoxin Fan, Hanyu Wang, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2025. Moc: Mixtures of text chunking learners for retrieval-augmented generation system. *CoRR*, abs/2503.09600.

Jihao Zhao, Zhiyuan Ji, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Meta-chunking: Learning efficient text segmentation via logical perception. *CoRR*, abs/2410.12788.

Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. 2025. Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 5756–5774. Association for Computational Linguistics.

A Teacher-Space Distillation and Alignment Diagnostics

The supervision used in Section 4 is *distillation-style* in form. For each contiguous sentence span, the reference target is the embedding produced by directly encoding the concatenated text with the base encoder, while *FreeChunker* learns a composition operator that reconstructs this target from sentence-level units.

This setup defines a clear boundary for what is and is not claimed. No claim is made that *FreeChunker* produces representations that are semantically better than the base encoder, nor that it corrects potential limitations of the base model on long-context semantics. Instead, the claim is that *FreeChunker* preserves the *ranking-relevant geometry* of a fixed reference embedding space while enabling efficient retrieval over configurable contiguous multi-granularity candidates.

To quantify this alignment without relying on passage-level labels for every possible chunk candidate, two label-free diagnostics are used throughout the appendix:

$$\mathrm{Dist}(\vec{v}, \vec{e}) = 1 - \cos(\vec{v}, \vec{e}), \tag{15}$$

$$\mathrm{Diff}(\vec{q}; \vec{v}, \vec{e}) = \left|\cos(\vec{q}, \vec{e}) - \cos(\vec{q}, \vec{v})\right|, \tag{16}$$

where $\vec{e}$ is the teacher embedding obtained from direct concatenated encoding, $\vec{v}$ is the composed embedding produced by *FreeChunker*, and $\vec{q}$ is a query embedding. Dist measures how closely the composed vector matches the teacher target, while Diff directly measures the perturbation to query-side similarity caused by substitution. In the granularity-discovery setting studied in this work, these metrics provide a direct way to localize gains even when standard passage-level retrieval labels are unavailable for arbitrary chunk candidates.

B Training Necessity and Aggregation Baselines

To assess whether training is necessary, three alternatives are compared: **fc-no**, an untrained cross-granularity encoder with the same architecture; **Avg**, simple mean pooling over sentence embeddings; and **FC**, the trained *FreeChunker* encoder. Table 2 reports aggregated results at coarser granularities, where composition quality is most critical.

Two observations are consistent across models. First, the near-unit cosine distance of **fc-no** shows that training is necessary; architectural bias alone is insufficient. Second, the trained encoder consistently improves over mean pooling, especially at granularity 16, indicating that learned composition captures cross-sentence dependencies that naive averaging does not preserve.

C Real-Query Statistics for Theoretical Validation

Section 5 derives a probabilistic substitution bound under an idealized assumption that the error direction is approximately random relative to the query. To connect this hypothesis to practical retrieval settings, the projection term $|\mathbf{w} \cdot \mathbf{u}|$ and the resulting substitution error ε are additionally measured under three empirical distributions: real query–chunk pairs, shuffled query–chunk pairs that preserve the marginal real distribution, and an isotropic random baseline.

These results should be interpreted as empirical diagnostics complementing, rather than replacing, the idealized bound. Real queries indeed exhibit stronger directional structure than a random baseline, yet the observed substitution error remains around 2×10^{-2}, supporting the practical stability of the approximation when used for ranking.

D Generalization to Larger Granularities

The main retrieval experiments in Section 6 use the representative inference configuration $(1, 2, 4)$. To test whether the learned composition remains stable for coarser contiguous chunks, the alignment diagnostics are additionally evaluated at granularities 2, 4, 8, and 16, aggregated across tasks. Table 4 compares the trained *FreeChunker* encoder with mean pooling.

The results show that the learned composition remains stable beyond the smallest inference configuration used in the main experiments. This evidence supports the claim that *FreeChunker* generalizes to larger *contiguous* granularities, while non-contiguous combinations remain outside the empirical scope of this study.

E Illustrative Reconstruction Example

To illustrate the reconstruction rule in Section 3.4, consider three retrieved candidates:

$$A = \{12, 13\},$$
$$B = \{11, 12, 13, 14\},$$
$$C = \{18, 19\}.$$

Model	Gran.	Dist(fc-no, Enc)	Dist(Avg, Enc)	Dist(FC, Enc)	Diff(Q, fc-no, Enc)	Diff(Q, Avg, Enc)	Diff(Q, FC, Enc)
BGE-M3	8	0.9638	0.1249	0.0828	0.4040	0.0423	0.0321
BGE-M3	16	0.9744	0.1526	0.0887	0.3761	0.0447	0.0338
jina	8	0.9614	0.0201	0.0092	0.5636	0.0233	0.0119
jina	16	0.9893	0.0226	0.0088	0.4371	0.0266	0.0129
nomic-embed-text-v1.5	8	0.9757	0.1158	0.0989	0.4729	0.0734	0.0474
nomic-embed-text-v1.5	16	0.9952	0.3070	0.2117	0.3805	0.1741	0.1243

Table 2: **Comparison of aggregation strategies at coarse granularities** (lower is better). The untrained encoder (**fc-no**) fails to approximate the teacher space, while the trained *FreeChunker* encoder (**FC**) consistently improves over simple mean pooling (**Avg**).

Distribution	Projection	Substitution Error
Real Query–Chunk	0.1661	0.0223
Shuffled (Real Dist.)	0.1438	0.0201
Random (Isotropic)	0.0296	0.0076

Table 3: **Observed mean statistics under different query distributions.** Real query distributions are more anisotropic than the isotropic baseline, but the resulting substitution error remains small in absolute magnitude.

where each set lists the global sentence indices contained in the retrieved chunk. Candidates A and B overlap on indices 12 and 13, so deduplication retains the unique ordered set $\{11, 12, 13, 14, 18, 19\}$. The final reconstructed context is therefore

$$[S_{11}, S_{12}, S_{13}, S_{14};\ \ldots;\ S_{18}, S_{19}],$$

where "..." indicates a skipped span in the original document.

F Long-Input Memory and Latency

To complement the end-to-end timing results in Figure 4, memory usage and latency are also measured on 100 long samples using `jina-embeddings-v2-small-en`. The benchmark uses the granularity set $\{2, 4, 8, 16\}$ and compares two settings: **Concat**, which directly encodes each concatenated candidate with the base encoder, and ***FreeChunker***, which computes all chunk representations in one forward pass via shared cross-granularity composition.

Five representative examples further illustrate the same pattern. For samples with lengths 37579, 28290, 30084, 36418, and 34267, direct concatenated encoding requires 8576.71–10141.09 MB and 120.06–144.95 ms, while *FreeChunker* stays within 4231.79–4669.72 MB and 77.21–95.69 ms. These measurements support the practical advantage of replacing repeated token-level encoding of overlapping spans with a single shared composition pass.

Model	Granularity	Avg Dist(FC, Enc)	Avg Dist(Avg, Enc)	Diff(Q, FC-Enc)	Diff(Q, Avg-Enc)
BGE-M3	2	0.0397	0.0440	0.0181	0.0230
BGE-M3	4	0.0649	0.0850	0.0262	0.0350
BGE-M3	8	0.0828	0.1249	0.0321	0.0423
BGE-M3	16	0.0887	0.1526	0.0338	0.0447
jina	2	0.0086	0.0106	0.0090	0.0137
jina	4	0.0094	0.0160	0.0106	0.0187
jina	8	0.0092	0.0201	0.0119	0.0233
jina	16	0.0088	0.0226	0.0129	0.0266
nomic-embed-text-v1.5	2	0.0313	0.0336	0.0227	0.0316
nomic-embed-text-v1.5	4	0.0636	0.0701	0.0334	0.0510
nomic-embed-text-v1.5	8	0.0989	0.1158	0.0474	0.0734
nomic-embed-text-v1.5	16	0.2117	0.3070	0.1243	0.1741

Table 4: **Alignment diagnostics for larger contiguous granularities** (lower is better). Across all three embedding models, the trained *FreeChunker* encoder remains closer to the teacher space than mean pooling, and the relative benefit becomes more pronounced for coarser chunks.

Method	Avg VRAM (MB)	Peak VRAM (MB)	Avg Time (ms)	Peak Time (ms)
Concat	9679.49	10141.09	137.47	144.95
FreeChunker	4344.25	4669.72	84.76	95.69
Reduction	-55.12%	-53.95%	-38.34%	-33.98%

Table 5: **Memory and latency on 100 long inputs** using `jina-embeddings-v2-small-en`. *FreeChunker* reduces both VRAM and latency relative to repeated direct concatenated encoding.