SELECTING WHAT MATTERS: SEMANTIC COMPRESSION-GUIDED SELECTIVE POOLING FOR LONG-CONTEXT EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown strong potential as training-free text encoders for long-context embeddings. Existing approaches primarily improve information flow under causal attention and typically construct embeddings by uniformly averaging all token representations. However, for long documents, such mean pooling can dilute salient semantic information with abundant redundant or weakly informative content. To this end, we propose SCSP, a training-free framework that leverages semantic compression for informative token selection in long-context embedding. Specifically, SCSP first partitions a document into sentence-aware chunks and appends a semantic compression prompt to each chunk. A prompt-isolated attention mask preserves information flow among document tokens while restricting each prompt to its corresponding local context. We then use the attention patterns elicited by these prompts to estimate token importance, select informative tokens, and aggregate their intermediate-layer representations into the final embedding. Extensive experiments on long-context embedding benchmarks demonstrate that SCSP can be integrated into both zero-shot and fine-tuned models in a plug-and-play manner, consistently improving their performance.

1 INTRODUCTION

Text embeddings have a wide range of real-world applications, including information retrieval, text clustering, and recommender systems. Recently, motivated by the strong long-context modeling and zero-shot generalization capabilities of large language models (LLMs), several studies (Ding et al., 2026; Fu et al., 2025) have explored directly extracting long-context embeddings from LLMs without additional parameter updates. This training-free paradigm avoids the computational cost of fine-tuning while preserving the original generative capabilities of LLMs, making off-the-shelf models readily applicable as text encoders.

Existing training-free methods improve LLM-based text embeddings by introducing backward dependencies under causal attention, allowing earlier tokens to incorporate information from subsequent positions. Echo (Springer et al., 2025) achieves this by repeating the input sequence and extracting representations from its second occurrence. To avoid the computational overhead of input repetition, Token Prepending (TP) (Fu et al., 2025) places a global semantic representation at the beginning of the input sequence. Hierarchical Token Prepending (HTP) (Ding et al., 2026) further introduces multiple block-level representations, alleviating the information bottleneck caused by compressing the entire document into a single global token.

Although existing methods improve how token representations incorporate global or backward context, they largely overlook the final semantic aggregation step. In particular, mean pooling is commonly adopted and has been shown to outperform last-token pooling for long-context embeddings (Ding et al., 2026). However, mean pooling implicitly assigns equal importance to all tokens, even though long documents often contain substantial amounts of semantically redundant or uninformative content. Consequently, uniformly averaging all token representations may dilute the contribution of salient tokens, causing critical semantic information to be overwhelmed by less informative content (Doshi et al., 2026). This observation motivates the following research question:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Can we identify and selectively aggregate informative tokens, rather than treating all tokens equally, to construct long-context embeddings without additional training?

To answer this question, we propose a training-free Semantic Compression-guided Selective Pooling (SCSP) method for long-context embedding. Instead of treating all token representations uniformly, SCSP exploits the attention patterns elicited by semantic compression prompts as intrinsic signals of token importance. Specifically, we partition a long document into sentence-aware chunks and append a semantic compression prompt to each chunk, enabling token importance to be estimated within focused and semantically coherent local contexts. A prompt-isolated attention mask further prevents the inserted prompts from interfering with the original document representations while restricting each prompt to its corresponding chunk. Based on the resulting attention distributions, SCSP selects semantically informative tokens and aggregates their representations to construct the final long-context embedding. Extensive experiments across multiple long-context embedding benchmarks demonstrate that SCSP consistently improves existing training-free approaches and can be seamlessly integrated into fine-tuned embedding models, highlighting the general effectiveness of selective semantic aggregation.

Our main contributions are summarized as follows:

- We revisit the final aggregation step in long-context embedding extraction and show that selectively pooling informative tokens is more effective than uniformly averaging all tokens.
- We propose SCSP, a training-free selective pooling framework that leverages attention elicited by local semantic compression as an intrinsic signal of token importance, enabling informative tokens to be identified and aggregated.
- Extensive experiments across multiple long-context embedding benchmarks show that SCSP consistently improves diverse training-free baselines and further enhances fine-tuned embedding models, demonstrating the broad applicability of selective semantic aggregation.

2 Related Work

Finetuning LLMs for Embeddings. Text embedding models map variable-length texts into fixed-dimensional vectors and serve as fundamental components in information retrieval, clustering, and semantic text similarity. Early neural text encoders, such as Sentence-BERT (Reimers & Gurevych, 2019) and SimCSE (Gao et al., 2021), typically relied on metric learning to acquire semantic representations. Recent studies (BehnamGhader et al., 2024; Li & Li, 2024; Lee et al., 2025) increasingly utilize LLMs as text encoders, leveraging the semantic knowledge acquired during generative pretraining. For example, LLM2Vec (BehnamGhader et al., 2024) converts LLMs into bidirectional encoders by removing the causal attention mask and subsequently applying masked next-token prediction and contrastive learning. In addition, several studies (Günther et al., 2023; Saad-Falcon et al., 2024; Zhang et al., 2024b; Doshi et al., 2026) have focused on long-context embedding models. However, these methods require large amounts of data and incur substantial fine-tuning costs, while also turning general-purpose LLMs into specialized embedding models.

Training-free LLMs for Embeddings. Motivated by the strong zero-shot capabilities of LLMs, recent studies (Jiang et al., 2024; Cheng et al., 2025; Springer et al., 2025; Fu et al., 2025; Ding et al., 2026) have begun to repurpose off-the-shelf LLMs as text encoders. One line of work (Jiang et al., 2024; Lei et al., 2024; Zhang et al., 2024a; Thirukovalluru & Dhingra, 2025) focuses on prompt engineering to elicit higher-quality text representations from LLMs. A representative method, PromptEOL (Jiang et al., 2024), introduces a semantic compression prompt, *This sentence: "`[TEXT]`" means in one word: "*, which encourages the model to compress the semantics of the input text into the representation of the final token. Another line of work focuses on introducing backward dependencies into LLMs to overcome the limitations of causal attention. Echo (Springer et al., 2025) repeats the input sequence, allowing the second occurrence to attend to the entire text. Token Prepending (TP) (Fu et al., 2025) prepends a global semantic representation to the input sequence, while Hierarchical Token Prepending (HTP) (Ding et al., 2026) further introduces multiple block-level semantic representations and shows that mean pooling is more effective than last-token

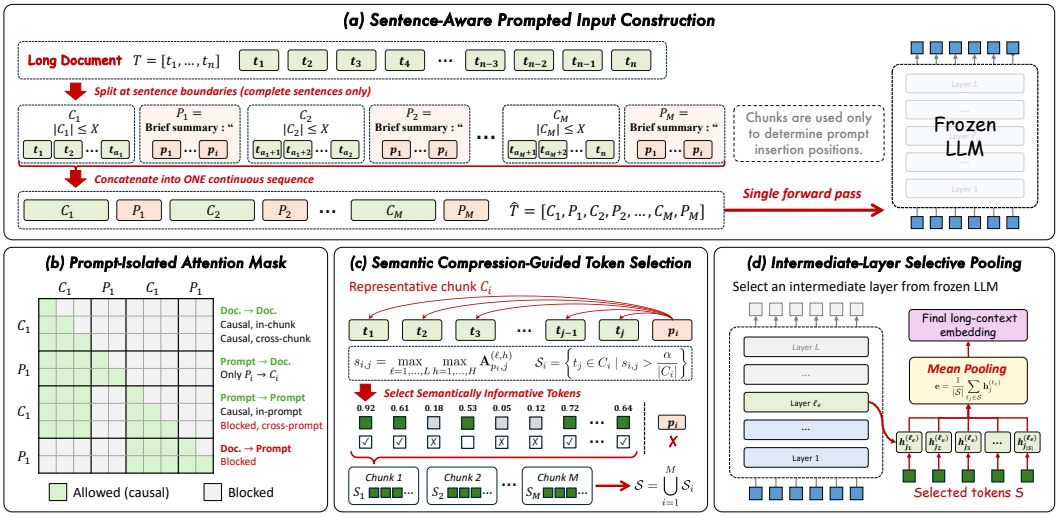

Figure 1: Illustration of the SCSP method.

pooling for long-context embeddings. Different from these approaches, our method focuses on long-context embeddings and uses the attention patterns elicited by semantic compression prompts to estimate token importance and selectively aggregate informative token representations.

3 METHOD

3.1 OVERVIEW

The key idea of our method is to leverage chunk-wise semantic compression prompts to identify informative tokens for long-context embedding construction. As illustrated in Figure 1, our framework consists of four main components: input construction, prompt-isolated attention mask, semantic compression-guided token selection, and intermediate-layer selective pooling.

Given a long document, we first partition it into multiple local sentence-aware chunks and append a semantic compression prompt to each chunk. Then, we modify the attention mask to prevent subsequent document tokens from attending to the semantic compression prompts, while restricting each prompt to attend only to tokens within its corresponding chunk. Subsequently, we use the attention distributions elicited by the semantic compression prompts to estimate token importance and select informative tokens. Finally, we aggregate the representations of all selected tokens from an intermediate layer to obtain the final long-context embedding.

3.2 INPUT CONSTRUCTION

Instead of estimating token importance over an entire long document, we first partition the input into multiple local contexts. By reducing the semantic scope associated with each compression prompt, this partitioning enables more reliable estimation of token importance within a semantically coherent context.

Given an input text sequence $T = [t_1, \cdots, t_n]$, we partition it into M non-overlapping sentence-aware chunks $T = [C_1, \cdots, C_M]$. Each chunk consists of several complete sentences and is constrained to contain no more than X tokens, where X denotes the maximum chunk size. This sentence-level partitioning preserves semantic coherence within each chunk and prevents the semantic information of a sentence from being divided across different chunks.

After partitioning the input into chunks, we augment each chunk by appending a short semantic compression prompt, $P = \texttt{Brief summary :}$ “, to its end, inspired by PromptEOL (Jiang et al., 2024). The prompt elicits the LLM’s intrinsic semantic compression capability, producing attention

patterns that reflect the semantic importance of tokens within each local context. In particular, the trailing colon and opening quotation mark in the prompt are used to prevent the model from generating punctuation as the next token, thereby enabling more effective semantic compression.

Finally, the resulting augmented input sequence is given by $\hat{T} = [C_1, P_1, C_2, P_2, \cdots, C_M, P_M]$. Notably, chunking is used solely to determine where to insert the semantic compression prompts. We still feed the entire augmented input sequence into the LLM at once, rather than processing each chunk separately.

3.3 Prompt-Isolated Attention Mask

After constructing the input, we modify the attention mask to prevent the semantic compression prompts from interfering with the original tokens in the long document and to restrict the attention scope of each prompt.

For the original document tokens, we preserve causal attention among all preceding document tokens while masking out all inserted prompt tokens. Consequently, the document is still processed jointly as a whole, allowing information to propagate across chunk boundaries, while the inserted prompts do not directly affect the representations of the original document tokens through attention. For the prompt appended to chunk C_i, we restrict its attention to the tokens in C_i and the preceding tokens within the prompt itself. Although each prompt directly attends only to its corresponding chunk, the document tokens themselves retain causal dependencies across chunk boundaries. Therefore, each prompt serves as a local semantic readout over globally contextualized token representations.

Formally, let $\mathcal{D}$ denote the positions of the original document tokens, $\mathcal{C}_i$ denote the positions of the tokens in chunk C_i, and $\mathcal{P}_i$ denote the positions of the prompt appended to C_i. The modified attention mask matrix $\mathbf{M}$ is defined as

$$M_{q,k} = \begin{cases} 0, & q \in \mathcal{D},\ k \in \mathcal{D},\ k \le q, \\ 0, & q \in \mathcal{P}_i,\ k \in \mathcal{C}_i, \\ 0, & q \in \mathcal{P}_i,\ k \in \mathcal{P}_i,\ k \le q, \\ -\infty, & \text{otherwise}. \end{cases}$$

This design prevents information from the inserted prompts from directly propagating into the document token representations through attention, while constraining each prompt to estimate token importance within a localized semantic context. Notably, we also explore modifying the positional encodings to keep the position indices of the original document tokens contiguous. In this way, each document token preserves the same representation as in the unaugmented input. However, this strategy does not yield any performance improvement.

3.4 Semantic Compression-Guided Token Selection

After constructing the prompt-augmented input and applying the modified attention mask, we use the attention patterns induced by the semantic compression prompts to identify informative tokens. The intuition is that semantic compression requires the model to focus on the most informative content within each chunk, which is reflected in the attention assigned to individual tokens. We therefore use the attention weights produced by each semantic compression prompt as estimates of token importance.

For each chunk C_i, let p_i denote the position of the final token in its appended prompt. Under the modified attention mask, the token at position p_i can attend only to the original tokens in C_i and the preceding tokens within its own prompt. Given the normalized self-attention score matrix $\mathbf{A}^{(\ell,h)}$ of the h-th attention head in the ℓ-th Transformer layer, we define the importance score of token $t_j \in C_i$ as

$$s_{i,j} = \max_{\ell=1,\ldots,L} \max_{h=1,\ldots,H} A^{(\ell,h)}_{p_i,j},$$

where L and H denote the numbers of Transformer layers and attention heads, respectively. We take the maximum across layers and attention heads so that a strong semantic association captured by an individual layer or head can contribute to token selection. Notably, since the normalized attention scores are used solely for token selection within each chunk, the attention assigned to the semantic compression prompt itself is excluded when normalizing the attention scores over the chunk.

Table 1: NDCG@10 (in percentage) on five datasets using Mistral-7B-Instruct-v0.3. We report the context length of 512 and an extended length of 8192. For the first four datasets that were also evaluated in HTP, † denotes the results reported in HTP, while ‡ denotes our reproduced results, which closely match the reported results.

CXT Len	Method	QMSum	2WikiMQA	SumFD	NQA	MultiFieldQA	Avg	Times
512	PromptEOL†	4.57	7.16	6.07	2.80	21.72	8.46	–
	TP w. PromptEOL†	5.44	6.51	5.52	1.72	17.58	7.35	–
	TP w. Mean†	11.97	18.62	38.50	3.08	44.80	23.39	–
	Vanilla Mean‡	15.01	16.07	40.39	2.56	48.61	24.53	1.00×
	Vanilla Mean + SCSP (***Ours***)	14.07	21.21	**45.63**	3.27	52.06	27.25 (+2.72)	1.19×
	Echo Mean‡	14.34	25.63	32.36	7.06	60.37	27.95	1.65×
	Echo Mean + SCSP (***Ours***)	**16.62**	**30.06**	40.48	**10.10**	**66.74**	**32.80 (+4.85)**	1.79×
	HTP‡	14.94	18.22	38.79	2.67	44.23	23.77	1.03×
	HTP + SCSP (***Ours***)	14.83	21.79	44.39	3.14	46.71	26.17 (+2.40)	1.12×
8192	PromptEOL†	6.36	7.49	9.86	4.08	26.33	10.82	–
	TP w. PromptEOL†	4.56	6.89	6.86	3.42	17.90	7.93	–
	TP w. Mean†	23.08	23.11	56.50	4.78	42.57	30.01	–
	Vanilla Mean‡	26.67	25.92	58.04	5.39	46.18	32.44	1.00×
	Vanilla Mean + SCSP (***Ours***)	**28.89**	34.46	70.58	10.62	51.19	**39.15 (+6.71)**	1.12×
	Echo Mean‡	17.36	29.49	39.24	8.75	53.88	29.74	2.68×
	Echo Mean + SCSP (***Ours***)	26.43	**36.65**	55.09	**11.22**	**58.08**	37.49 (+7.75)	2.88×
	HTP‡	25.36	25.64	56.38	5.37	43.88	31.33	1.05×
	HTP + SCSP (***Ours***)	28.21	34.60	**71.43**	8.80	50.84	38.77 (+7.44)	1.19×

We then define the selected token set $\mathcal{S}_i$ for chunk C_i as

$$\mathcal{S}_i = \left\{ t_j \in C_i \mid s_{i,j} > \frac{\alpha}{|C_i|} \right\},$$

where α is a threshold scaling factor and $|C_i|$ denotes the number of tokens in chunk C_i. Since $1/|C_i|$ corresponds to uniform attention over the chunk, a token is retained if its importance score exceeds this baseline by a factor of α.

Finally, the selected tokens from all chunks are combined into a global set for subsequent embedding construction:

$$\mathcal{S} = \bigcup_{i=1}^{M} \mathcal{S}_i.$$

3.5 Intermediate-Layer Selective Pooling

Since final-layer representations are primarily optimized for next-token prediction, while intermediate layers tend to preserve richer semantic information (Fu et al., 2025; Liu et al., 2024), we use an intermediate layer as the output layer to extract token representations and aggregate them into the final long-context embedding.

Given the representation $\mathbf{h}_j^{(\ell_e)}$ of token t_j from layer ℓ_e, we obtain the final long-context embedding by mean-pooling the representations of the selected tokens:

$$\mathbf{e} = \frac{1}{|\mathcal{S}|} \sum_{t_j \in \mathcal{S}} \mathbf{h}_j^{(\ell_e)}.$$

4 Experiments

4.1 Datasets and Experimental Settings

We evaluate the performance of long-context embeddings on five real-world datasets from LongBench (Bai et al., 2024) and LongEmbed (Zhu et al., 2024): QMSum (Zhong et al., 2021), 2WikiMultiHopQA (Ho et al., 2020), SummScreenFD (Chen et al., 2022), NarrativeQA (Kociský et al., 2018), and MultiFieldQA (Bai et al., 2024). We use nDCG@10 as the evaluation metric.

Table 2: NDCG@10 (in percentage) on five datasets using LLaMA-3.1-8B-Instruct.

CXT Len	Method	QMSum	2WikiMQA	SumFD	NQA	MultiFieldQA	Avg	Times
512	PromptEOL†	10.17	15.20	11.39	5.22	46.70	17.73	–
	TP w. PromptEOL†	8.33	8.29	10.97	1.91	25.65	11.03	–
	TP w. Mean†	7.13	5.29	6.95	1.57	17.72	7.73	–
	Vanilla Mean‡	12.21	18.09	35.92	2.82	45.39	22.89	1.00×
	Vanilla Mean + SCSP (***Ours***)	11.60	26.08	48.57	3.48	49.86	27.92 (+5.03)	1.34×
	Echo Mean‡	13.86	22.85	36.35	7.56	67.83	29.69	1.90×
	Echo Mean + SCSP (***Ours***)	**13.68**	**28.02**	**50.79**	**9.15**	**72.93**	**34.91 (+5.22)**	2.27×
	HTP‡	10.37	8.64	27.81	2.22	27.21	15.25	1.01×
	HTP + SCSP (***Ours***)	11.54	13.58	30.91	2.28	30.53	17.77 (+2.52)	1.38×
8192	PromptEOL†	8.16	15.86	20.47	7.40	43.85	19.15	–
	TP w. PromptEOL†	4.95	9.87	24.19	2.42	19.81	12.25	–
	TP w. Mean†	10.00	8.75	17.95	2.66	19.28	11.73	–
	Vanilla Mean‡	27.04	19.76	54.39	7.50	47.83	31.30	1.00×
	Vanilla Mean + SCSP (***Ours***)	**29.24**	33.27	**76.66**	12.15	51.62	40.59 (+9.28)	1.08×
	Echo Mean‡	18.00	17.62	38.70	8.14	49.57	26.41	2.74×
	Echo Mean + SCSP (***Ours***)	23.52	**39.28**	62.38	**17.27**	**69.26**	**42.34 (+15.94)**	2.92×
	HTP‡	15.73	8.93	41.48	2.97	26.97	19.21	1.05×
	HTP + SCSP (***Ours***)	21.49	14.66	44.88	4.58	29.61	23.05 (+3.83)	1.22×

We conduct all experiments using 4 NVIDIA Tesla V100 GPUs. We use Spacy's parser to split the text into sentences. We primarily evaluate *LLaMA3.1-8B-Instruct* (Dubey et al., 2024) and *Mistral-7B-Instruct-v0.3* (Jiang et al., 2023) with context lengths of 512 and 8,192 tokens. For a context length of 512, we set the chunk size to 256 and perform a grid search over the ratio values α in $\{1, 1.25, 1.5, 1.75, 2\}$. For a context length of 8,192, we set the chunk size to 512 and search α over $\{1.5, 1.75, 2, 2.25, 2.5\}$. For *Mistral-7B-Instruct-v0.3*, following HTP (Ding et al., 2026), we use the third-to-last layer as the output layer. For *LLaMA3.1-8B-Instruct*, we search for the optimal output layer among the final seven layers.

4.2 Baselines

We compare our method with the following baselines. **PromptEOL** (Jiang et al., 2024) compresses the semantics of a text into the last token to extract embeddings. **TP** (Fu et al., 2025) prepends a global semantic token to the beginning of the input. For TP, we also explore two embedding extraction strategies: **TP w. PromptEOL**, which uses last-token pooling, and **TP w. Mean**, which uses mean pooling. **Vanilla Mean** directly applies mean pooling over the hidden states of all tokens produced by the LLM. **Echo Mean** (Springer et al., 2025) repeats the text twice and uses mean-pooling to extract embeddings from the second occurrence. **HTP** (Ding et al., 2026) extends TP by introducing multiple block-level representations, and constructs the final embedding through mean pooling over all token representations. In particular, to ensure a fair comparison, all baseline methods use the same output layer for representation extraction.

4.3 Results

As shown in Tables 1 and 2, our method consistently improves the performance of Vanilla Mean, Echo Mean, and HTP across both Mistral-7B-Instruct-v0.3 and LLaMA-3.1-8B-Instruct models. These results demonstrate that our method is effective and can be seamlessly integrated with baselines in a plug-and-play manner. Using LLaMA with a context length of 8,192, combining our method with Echo Mean improves the average performance by 60% relative to Echo Mean alone. Notably, when combined with our method, the simplest Vanilla Mean achieves the best performance on Mistral-7B-Instruct-v0.3 at a context length of 8,192.

At a context length of 512, Echo Mean serves as a strong baseline because input repetition enhances the model's contextual understanding. Our method is complementary to Echo Mean and further improves its performance. At a context length of 8,192, Echo Mean generally fails to yield performance improvements, possibly because the context becomes excessively long. In contrast, our method achieves larger gains in this setting, likely because selectively preserving salient semantic information becomes increasingly important as the context length grows.

Table 3: Ablation study results on Mistral-7B-Instruct-v0.3 with an 8,192-token context length.

Method	QMSum	2WikiMQA	SumFD	NQA	MultiFieldQA	Avg
Vanilla Mean	26.67	25.92	58.04	5.39	46.18	32.44
Vanilla Mean + SCSP (***Ours***)	28.89	34.46	**70.58**	10.62	51.19	**39.15**
Input Construction:						
w/o Sentence-aware Chunking	28.65	33.47	69.17	8.27	51.13	38.14
w/o Chunk-wise Prompt	24.59	29.56	67.25	8.43	**52.30**	36.42
w/o Semantic Compression Prompt	26.17	29.08	59.50	8.59	48.55	34.38
Prompt Isolation and Position Encoding:						
w/o Prompt-Isolated Attention Mask	27.90	29.61	61.41	8.40	50.96	35.66
w/ Positional Encoding Modification	**29.05**	34.60	69.14	10.07	51.13	38.80
Token Selection and Aggregation:						
Random Token Selection	26.22	27.88	57.51	5.79	45.91	32.66
Aggregation of Prompt-Compressed Representations	26.03	26.93	59.70	10.43	47.56	34.13
Importance-Weighted Token Aggregation	28.85	**34.80**	68.86	**10.80**	50.83	38.83

Notably, compared with the substantial runtime overhead incurred by Echo's input repetition, our method remains lightweight and introduces only a marginal additional runtime cost.

4.4 Ablation Study

We conduct ablation studies to investigate the effects of key design choices in SCSP, as shown in Table 3.

We investigate the effect of input construction on performance. First, we replace the sentence-aware semantic chunking strategy with fixed-length chunking. This variant leads to an average performance drop of 1.01%, demonstrating that sentence-level chunking better preserves semantic coherence and facilitates effective token selection. Second, instead of appending a semantic compression prompt to each chunk, we append a single prompt to the end of the entire long text for token selection. This modification results in a performance drop of 2.73%, highlighting the importance of performing token selection within localized chunk-level contexts. This may be because the input text is excessively long and attention scores are influenced by the relative distances between tokens. Finally, we ablate the prompt and directly use the attention scores from the final token of each chunk to the preceding tokens for token selection. The performance decreases by 4.77%, indicating that using semantic compression prompts to select tokens is a more effective strategy.

We investigate the effect of prompt isolation and position encoding on performance. First, we retain the original causal attention pattern without modifying the attention mask, which results in a 3.49% performance decrease. This finding indicates that allowing tokens to attend to the additionally appended prompts interferes with their representations and consequently degrades the quality of long-context embeddings. Second, we explore modifying the positional encodings to keep the position indices of all document tokens contiguous, which results in a 0.35% performance drop. This may be because retaining the positional gaps introduced by the inserted prompts helps the model better distinguish and capture semantic information within each chunk.

We further investigate token selection strategies and aggregation methods. First, we randomly select the same number of tokens within each chunk. The performance is close to that of Vanilla Mean, indicating that the gains of our method do not simply come from aggregating fewer tokens, but from identifying tokens that capture the core semantics of long documents. Second, we directly aggregate the representation of the final token in the semantic compression prompt for each chunk, which serves as the compressed semantic representation of that chunk. This modification leads to a 5.02% performance drop. Consistent with the findings of HTP (Ding et al., 2026), this result suggests that aggregating token-level representations is more effective than directly aggregating compressed prompt representations. Finally, we use the token importance scores as weights to perform weighted aggregation of token representations. We observe no further improvement over SCSP, suggesting that token importance scores are effective for token selection but are less suitable as weights for representation aggregation.

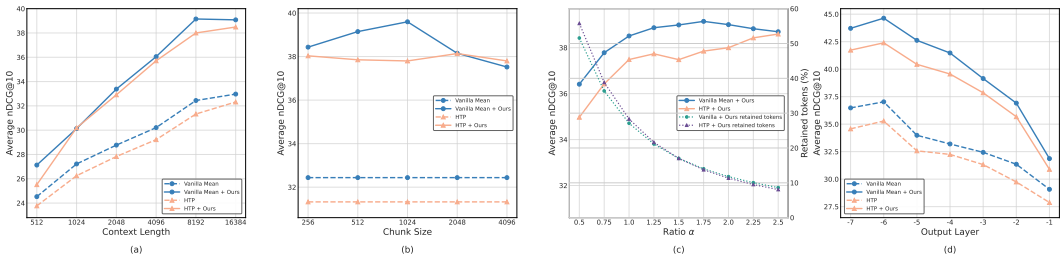

Figure 2: The average effects of context length, chunk size, ratio α, and output layer on performance across five datasets are evaluated using Mistral-7B-Instruct-v0.3.

Table 4: Generalization across different LLM backbones on five datasets with an 8,192-token context length.

Backbone	Method	QMSum	2WikiMQA	SumFD	NQA	MultiFieldQA	Avg
Gemma2-9B	Vanilla Mean	30.78	33.82	70.55	7.95	49.99	38.62
	Vanilla Mean + SCSP (***Ours***)	**32.72**	**43.56**	**73.61**	**11.55**	**53.35**	**42.96 (+4.34)**
Qwen2.5-7B-Instruct	Vanilla Mean	25.55	22.43	47.49	6.33	**50.16**	30.39
	Vanilla Mean + SCSP (***Ours***)	**26.66**	**25.90**	**54.77**	**7.90**	45.52	**32.15 (+1.76)**

4.5 Generalizability across Different Context Lengths

We further investigate the effect of different context lengths in Figure 2(a). First, our method consistently improves the performance of both baseline methods across multiple context lengths, demonstrating its effectiveness. Second, larger context lengths generally lead to better performance. However, the performance gains gradually diminish as the context window increases. Notably, at a context length of 16,384, Vanilla Mean achieves further performance improvements on SumFD and NQA, suggesting that these two datasets benefit from a larger context window.

4.6 Effects of Hyperparameters

We further investigate the effects of different hyperparameters in Figure 2(b), 2(c), and 2(d). Our method consistently improves the performance of both Vanilla Mean and HTP across a wide range of hyperparameter settings, demonstrating its effectiveness and robustness.

For *chunk size*, Vanilla Mean achieves its best performance with a chunk size of 1,024, whereas HTP performs best with a chunk size of 2,048. This suggests that the optimal chunk size varies across baselines, although our method yields consistent improvements for both. For the *ratio*, Vanilla Mean achieves the best performance at 1.75, while HTP performs best at 2.5. One possible explanation is that the token representations used by HTP incorporate more global semantic information and are therefore more contextualized, allowing a higher selection threshold to be applied. Notably, under their respective best-performing configurations, SCSP retains only 14.03% of the tokens for Vanilla Mean and 8.64% for HTP. This result indicates that aggregating representations from a small subset of informative tokens is sufficient to construct effective long-context embeddings. For the *output layer*, the best performance is obtained at the sixth-to-last layer, after which the performance gradually decreases as the output layer moves closer to the final layer.

4.7 Generalization across Different Backbones

We further report the performance of vanilla mean on Gemma2-9B (Team, 2024a) and Qwen-2.5-7B-Instruct (Team, 2024b) to validate the generalizability of our method across different backbones.

Experimental results in Table 4 show that our method is also effective on these two LLMs, further demonstrating its generalizability. Among the four LLMs, Gemma2-9B achieves the best performance with Vanilla Mean at a context length of 8,192.

Table 5: Comparison of different strategies for computing token importance scores on Mistral-7B-Instruct-v0.3 with an 8,192-token context length.

Method	Layer	Head	QMSum	2WikiMQA	SumFD	NQA	MultiFieldQA	Avg
Vanilla Mean	–	–	26.67	25.92	58.04	5.39	46.18	32.44
SCSP	Max	Max	28.89	**34.46**	**70.58**	10.62	**51.19**	**39.15**
	Max	Mean	**29.35**	34.53	69.06	9.91	50.87	38.75
	Mean	Max	27.58	33.20	67.38	**10.69**	50.57	37.89
	Mean	Mean	25.91	30.30	63.08	10.10	49.15	35.71

Table 6: Generalization of SCSP to fine-tuned embedding models on five datasets with an 8,192-token context length.

Models	Method	QMSum	2WikiMQA	SumFD	NQA	MultiFieldQA	Avg
GritLM	Vanilla Mean	19.98	27.63	29.48	5.89	72.46	31.09
	Vanilla Mean + SCSP (***Ours***)	**24.34**	**37.27**	**57.41**	**14.74**	**78.31**	**42.42 (+11.33)**
NV-EMBED	Vanilla Mean	24.87	34.77	67.11	24.77	74.61	45.23
	Vanilla Mean + SCSP (***Ours***)	**28.12**	**34.99**	**72.55**	**29.15**	**76.63**	**48.29 (+3.06)**

4.8 Effects of Token Importance Scores

We further investigate the effect of different strategies for computing token importance scores in Table 5. All four strategies improve performance, with taking the maximum across all layers and attention heads achieving the best results, while averaging across all layers and heads performs the worst. This suggests that the max operation is more effective at uncovering salient semantic signals captured by individual layers or attention heads, whereas mean aggregation may smooth out such informative signals.

4.9 Results on Finetuned Embedding Models

We further investigate whether our method is also effective for fine-tuned embedding models in Table 6. Specifically, we replace their mean pooling operation with our selective pooling method and use the final-layer representations to construct the embeddings. The results show that our method consistently improves the performance of both GritLM (Muennighoff et al., 2025) and NV-EMBED (Lee et al., 2025). Notably, NV-EMBED employs latent attention to update each token representation. This suggests that semantically irrelevant tokens remain unable to capture useful semantic information even after latent-attention refinement. These results demonstrate that our method is effective not only for general-purpose LLMs but also for further improving fine-tuned embedding models, highlighting its generalizability.

5 Conclusion

In this work, we introduced SCSP, a training-free framework for constructing long-context embeddings through semantic compression-guided selective pooling. SCSP first partitions a long document into sentence-aware chunks and appends a semantic compression prompt to each chunk to estimate token importance. A prompt-isolated attention mask prevents the inserted prompts from interfering with the original document representations, while the resulting attention patterns are used to identify and selectively aggregate semantically informative tokens. Extensive experiments demonstrate that SCSP consistently improves Vanilla Mean, Echo Mean, and HTP across different context lengths and LLM backbones. Moreover, SCSP also enhances finetuned embedding models, demonstrating its broad applicability. These results highlight selective token aggregation as an effective alternative to conventional mean pooling.

AI use statement

Generative AI tools were used to assist language editing, translation, and LaTeX preparation. The authors take responsibility for the final content, including all AI-assisted text and artifacts.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pp. 3119–3137, 2024. URL https://doi.org/10.18653/v1/2024.acl-long.172.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=IW1PR7vEBf.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. Summscreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*, pp. 8602–8615, 2022. URL https://doi.org/10.18653/v1/2022.acl-long.589.

Zifeng Cheng, Zhonghui Wang, Yuchen Fu, Zhiwei Jiang, Yafeng Yin, Cong Wang, and Qing Gu. Contrastive prompting enhances sentence embeddings in llms through inference-time steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, pp. 3475–3487, 2025. URL https://doi.org/10.18653/v1/2025.acl-long.174.

Xueying Ding, Xingyue Huang, Mingxuan Ju, Liam Collins, Yozen Liu, Leman Akoglu, Neil Shah, and Tong Zhao. Hierarchical token prepending: Enhancing information flow in decoder-based LLM embeddings. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026*, pp. 19047–19066, 2026.

Meet Doshi, Aashka Trivedi, Vishwajeet Kumar, Parul Awasthy, Yulong Li, Jaydeep Sen, Radu Florian, and Sachindra Joshi. LMK $>$ CLS: landmark pooling for dense embeddings. *CoRR*, abs/2601.21525, 2026. URL https://doi.org/10.48550/arXiv.2601.21525.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuchen Fu, Zifeng Cheng, Zhiwei Jiang, Zhonghui Wang, Yafeng Yin, Zhengliang Li, and Qing Gu. Token prepending: A training-free approach for eliciting better sentence embeddings from llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, pp. 3168–3181, 2025. URL https://doi.org/10.18653/v1/2025.acl-long.159.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pp. 6894–6910, 2021. URL https://doi.org/10.18653/v1/2021.emnlp-main.552.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *CoRR*, abs/2310.19923, 2023. URL https://doi.org/10.48550/arXiv.2310.19923.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pp. 6609–6625, 2020. URL https://doi.org/10.18653/v1/2020.coling-main.580.

Albert Q. 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. Mistral 7b. *CoRR*, abs/2310.06825, 2023. URL https://doi.org/10.48550/arXiv.2310.06825.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3182–3196, 2024. URL https://doi.org/10.18653/v1/2024.findings-emnlp.181.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Trans. Assoc. Comput. Linguistics*, 6:317–328, 2018. URL https://doi.org/10.1162/tacl_a_00023.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025. URL https://openreview.net/forum?id=lgsyLSsDRe.

Yibin Lei, Di Wu, Tianyi Zhou, Tao Shen, Yu Cao, Chongyang Tao, and Andrew Yates. Meta-task prompting elicits embeddings from large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pp. 10141–10157, 2024. URL https://doi.org/10.18653/v1/2024.acl-long.546.

Xianming Li and Jing Li. Bellm: Backward dependency enhanced large language model for sentence embeddings. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 792–804, 2024.

Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. Fantastic semantics and where to find them: Investigating which layers of generative llms reflect lexical semantics. In *Findings of the Association for Computational Linguistics, ACL 2024*, pp. 14551–14558, 2024. URL https://doi.org/10.18653/v1/2024.findings-acl.866.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025. URL https://openreview.net/forum?id=BC4lIvfSzv.

Nils Reimers and Iryna Gurevych. 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 2019*, pp. 3980–3990, 2019. URL https://doi.org/10.18653/v1/D19-1410.

Jon Saad-Falcon, Daniel Y. Fu, Simran Arora, Neel Guha, and Christopher Ré. Benchmarking and building long-context retrieval models with loco and M2-BERT. In *Forty-first International Conference on Machine Learning, ICML 2024*, Proceedings of Machine Learning Research, pp. 42918–42946, 2024. URL https://proceedings.mlr.press/v235/saad-falcon24a.html.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025. URL https://openreview.net/forum?id=Ahlrf2HGJR.

Gemma Team. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024a. URL https://doi.org/10.48550/arXiv.2408.00118.

Qwen Team. Qwen2.5: A party of foundation models, September 2024b. URL https://qwenlm.github.io/blog/qwen2.5/.

Raghuveer Thirukovalluru and Bhuwan Dhingra. Geneol: Harnessing the generative power of llms for training-free sentence embeddings. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 2295–2308, 2025. URL https://doi.org/10.18653/v1/2025.findings-naacl.122.

Bowen Zhang, Kehua Chang, and Chunping Li. Simple techniques for enhancing sentence embeddings in generative language models. *arXiv preprint arXiv:2404.03921*, 2024a.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track*, pp. 1393–1412, 2024b. URL https://doi.org/10.18653/v1/2024.emnlp-industry.103.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir R. Radev. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pp. 5905–5921, 2021. URL https://doi.org/10.18653/v1/2021.naacl-main.472.

Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Longembed: Extending embedding models for long context retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pp. 802–816, 2024.

Table 7: Effects of semantic compression prompts on Mistral-7B-Instruct-v0.3 with an 8,192-token context length.

Prompt	QMSum	2WikiMQA	SumFD	NQA	MultiFieldQA	Avg
Brief summary: “	**28.89**	34.46	**70.58**	**10.62**	51.19	**39.15**
Brief summary:	28.62	34.14	67.59	10.03	51.54	38.38
This text means in one word: “	28.22	31.88	65.24	9.41	50.98	37.15
This text can be summarized as: “	28.01	33.72	65.92	9.5	**52.23**	37.87
Summary: “	28.42	**35.05**	70.07	10.05	52.15	**39.15**

Table 8: Results on the LOCOV1 dataset using Mistral-7B-Instruct-v0.3 with an 8,192-token context length.

Dataset	Vanilla Mean	Vanilla Mean + SCSP (*Ours*)
SumFD	57.68	**70.24**
Gov. Report	97.30	**98.12**
QMSUM	50.94	**55.12**
QASPER Title	37.27	**46.40**
QASPER Abstract	89.19	**93.45**
2WikiMQA	46.90	**51.76**
Passage Retrieval	49.05	**52.99**
CourtListener - Plain Text	17.85	**18.04**
Avg.	55.77	**60.77 (+5.00)**

A Effects of Semantic Compression Prompt

We investigate the effects of different prompts on performance in Table 7.

All evaluated prompts yield performance improvements, indicating that our method generalizes well across different prompts and exhibits broad applicability. We further remove the opening quotation mark from the prompt and observe a performance decrease of 0.77%. This finding suggests that the opening quotation mark provides a more effective cue for eliciting the desired semantic continuation.

B Results on the LOCOV1 Dataset

We further evaluate our method on the LOCOV1 dataset (Saad-Falcon et al., 2024).

SCSP consistently improves Vanilla Mean across all eight datasets, increasing the average score from 55.77 to 60.77, with an absolute gain of 5.00 points. The improvements are particularly pronounced on SumFD and QASPER Title, where SCSP yields gains of 12.56 and 9.13 points, respectively. Consistent gains are also observed across QA, retrieval, and long-document understanding tasks, indicating that selective aggregation can effectively suppress less informative tokens and produce more discriminative long-context representations.