Frozen LLMs are Native Decoders for High-Norm Semantic Vectors

Yunsheng Zeng, Yongmei Tan*
Beijing University of Posts and Telecommunications
zengyunsheng@bupt.edu.cn, ymtan@bupt.edu.cn

Abstract

Large language models (LLMs) are designed for discrete tokens, yet they operate in a continuous embedding space. Recent context compression methods exploit this property by encoding text into dense vectors for frozen LLM decoding. However, a key question remains unanswered: how does a frozen LLM interpret continuous vectors that encode complex semantics? We investigate this through controlled reconstruction experiments. Our analysis reveals a critical geometric property: successful compression encoders learn to produce vectors with L2 norms two orders of magnitude higher than standard embeddings. Norm-scaling interventions provide strong evidence that this high-norm regime is an enabling factor for frozen-LLM decoding, while leaving open whether the effect arises from attention dominance, low-norm suppression, or both. Based on this finding, we propose a landmark-based compression framework for long contexts. Our encoder uses bidirectional attention over landmark tokens, which captures global dependencies and avoids semantic fragmentation from segment-based methods. Experiments on text reconstruction and four QA benchmarks provide evidence for our method. At 4x compression, our method is strongest on SQuAD and AdversarialQA and remains competitive on average.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across diverse tasks by mapping discrete tokens into a continuous embedding space (Mikolov et al., 2013). However, the self-attention mechanism in Transformer architectures exhibits quadratic complexity with respect to sequence length (Vaswani et al., 2017), encountering a computational bottleneck for long-context applications such as document retrieval and multi-turn dialogues. Context compression has emerged as a promising solution to this challenge, aiming to condense extensive inputs into compact representations while preserving essential information.

Existing compression methods generally fall into two categories: hard compression and soft compression. Hard compression selectively prunes tokens based on information-theoretic metrics (Jiang et al., 2023b; Li et al., 2023), offering interpretability but risking the loss of fine-grained information. Soft compression, in contrast, maps text into continuous latent vectors to achieve higher information density. Methods such as AutoCompressor (Chevalier et al., 2023), xRAG (Cheng et al., 2024), and PCC (Dai et al., 2025) have demonstrated promising results in this direction. However, these methods typically treat the frozen LLM as a black-box decoder, leaving a fundamental question unanswered: how does a model optimized for discrete tokens interpret a continuous vector that encodes complex semantics?

To address this question, we conduct controlled reconstruction experiments to probe the decoding mechanism of frozen LLMs. Our work reveals a consistent geometric phenomenon: the encoder learns to produce vectors with L2 norms that are two orders of magnitude higher than standard word embeddings. Through norm-scaling interventions, we obtain strong evidence that this high-norm signal is an enabling factor for successful decoding: reducing the magnitude leads to significant performance degradation. Attention pattern analysis further confirms that the frozen LLM treats these high-norm vectors as dominant information sources during generation, while leaving open whether the effect is best understood as attention amplification, low-norm suppression, or a combination of both.

Building on this mechanistic insight, we extend our framework to document-level compression by adopting a landmark-based, chunking-free paradigm. Our encoder inserts learnable landmark tokens into the input sequence and employs bidirectional attention to aggregate global context.

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 37028–37043

This mechanism avoids the semantic fragmentation commonly observed in segment-based compression methods, where explicit text partitioning disrupts contextual coherence. The resulting landmark vectors inherit the high-norm property discovered at the sentence level, serving as compact yet information-rich representations of long documents.

In summary, our contributions are:

- We identify high L2 norm as a key geometric signal associated with successful frozen-LLM decoding of information-dense continuous vectors. This finding is supported across multiple model architectures by intervention-based experiments.
- We propose a landmark-based, chunking-free compression framework that preserves global semantic coherence without explicit text partitioning.
- We evaluate our method on text reconstruction and four reading comprehension benchmarks. At 4x and 16x compression ratios, our method achieves strong performance, particularly on SQuAD and AdversarialQA at 4x and against global soft-compression baselines at 16x.

2 Methodology

We study the problem in two stages. First, we establish a controlled text reconstruction task at the sentence level to isolate and analyze the decoding capabilities of frozen LLMs under a single-vector latent constraint; second, we extend this mechanism to document-level compression using a chunking-free paradigm that enables long-context reasoning without explicit text partitioning.

2.1 Sentence-Level Mechanism Probing

We formulate a reconstruction task by mapping a sentence S into a single continuous vector $\mathbf{v} \in \mathbb{R}^d$, as illustrated in Figure 1. This representation serves as the primary conditional input for the frozen decoder, which must leverage the semantic information encoded in $\mathbf{v}$ to recover the original text. This single-vector constraint tests whether the model can decode continuous signals beyond its discrete vocabulary.

The vector $\mathbf{v}$ is generated through a two-step process. First, we extract the final hidden state from the frozen LLM when it processes the input sentence S. This hidden state captures the model's internal representation of the sentence semantics.

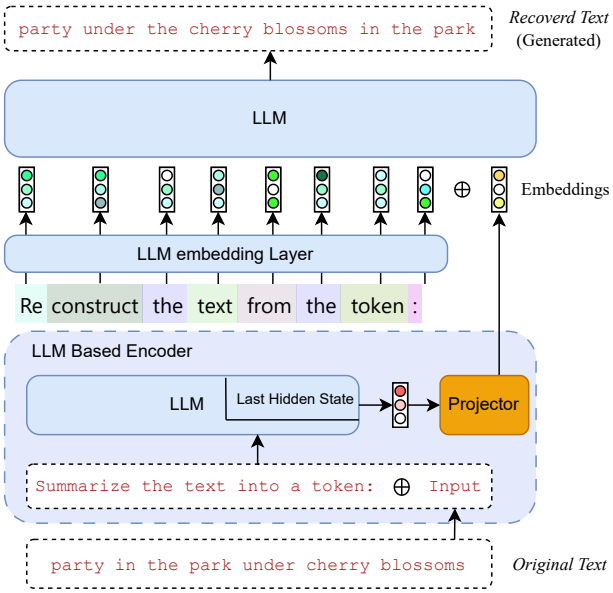

Figure 1: The pipeline of our proposed method for sentence compression and reconstruction.

Second, the representation is projected through a trainable network:

$$\mathbf{v} = \text{MLP}_\theta(\text{LLM}_{\text{frozen}}(S)) \quad (1)$$

where MLP_θ denotes a dual-layer multilayer perceptron. During training, only the projection parameters θ are optimized while θ_{LLM} remains frozen, thereby isolating the decoder's native capacity to interpret continuous representations.

For reconstruction, the vector $\mathbf{v}$ is prepended to a task instruction I_{rec} and fed into the frozen LLM. The training objective minimizes the negative log-likelihood:

$$\mathcal{L}_{probe}(\theta) = -\log P(S \mid \mathbf{v}, I_{rec}; \theta, \theta_{LLM}) \quad (2)$$

Since θ_{LLM} remains frozen, successful reconstruction directly reflects the model's innate decoding ability.

2.2 Document-Level Landmark Compression

Based on the mechanism verified at the sentence level, we extend the framework to handle long documents by representing them as sequences of latent vectors (Figure 2). This approach applies a landmark-based, chunking-free compression paradigm that maintains global semantic coherence and avoids the fragmentation typically associated with segment-based methods.

2.2.1 Landmark Token Insertion

Unlike segment-based compression methods that partition documents into fixed-size chunks, our ap-

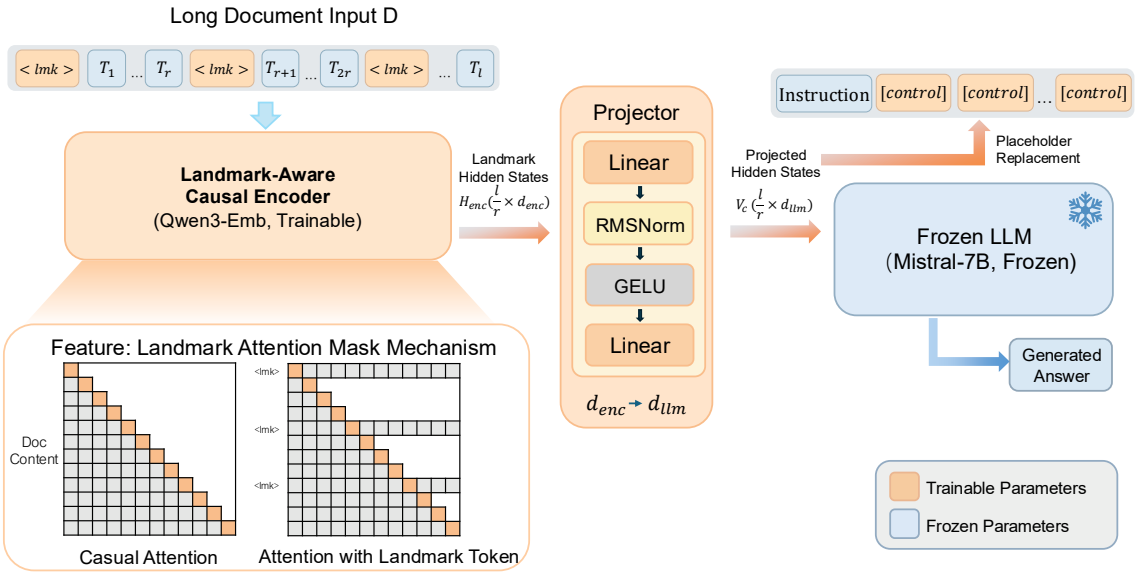

Figure 2: The framework of Landmark-based Context Compression. It comprises: (1) A Landmark-Aware Encoder extracting contextual features with bidirectional attention; (2) A Projection Network aligning encoder representations with the LLM's embedding space; and (3) A Frozen LLM performing native decoding on projected landmark vectors.

proach inserts learnable landmark tokens directly into the input sequence. This insertion strategy preserves the sequential structure of the document while providing fixed positions for aggregating contextual information. Given a document D and a target compression rate r, we insert a special landmark token $\langle \text{lmk} \rangle$ for every r text tokens. For example, with $r = 4$, one landmark token is inserted after every four text tokens, achieving a $4\times$ compression ratio.

The key advantage of this approach is that landmark tokens can attend to the entire document context through bidirectional attention, rather than being limited to local segments. This enables each landmark vector to capture global semantic dependencies, which is particularly important for tasks requiring cross-document reasoning.

2.2.2 Representation Generation

The processed sequence (containing both text tokens and landmark tokens) is processed by a transformer-based encoder. We employ bidirectional attention masks that allow landmark tokens to attend to all text tokens in the document, while text tokens attend only to preceding tokens (causal masking). This asymmetric attention design enables landmarks to aggregate global context while preserving the autoregressive nature of text token processing.

The landmark vector sequence $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ is extracted from the encoder outputs and projected into the LLM's embedding space:

$$\mathbf{v}_i = \text{MLP}_\theta(\text{Encoder}_\theta(D, r)[\text{idx}_i]) \quad (3)$$

where idx_i denotes the position of the i-th landmark token in the sequence, and $k = \lfloor |D|/r \rfloor$ is the total number of landmark vectors. In this phase, both the transformer encoder and the projection network are jointly optimized, with their parameters collectively denoted by θ. This joint optimization ensures that the encoder learns to aggregate contextual dependencies into each landmark vector $\mathbf{v}_i$ in a way that maximizes decodability by the frozen LLM.

During inference, the resulting sequence $\mathbf{V}$ replaces specific placeholder tokens in the frozen LLM's input template. These placeholder tokens are predefined in the prompt format, and their embeddings are substituted with the corresponding landmark vectors before being fed to the decoder.

2.2.3 Training Objectives

The training procedure is structured into two sequential stages to optimize θ while keeping θ_{LLM} frozen.

Stage 1: Paraphrase Pretraining. This stage establishes a high-fidelity mapping between the continuous latent space and the discrete text distribution. The encoder is trained to condense a document D into the landmark sequence $\mathbf{V}$ such that the original text can be fully recovered by the frozen LLM. Given a reconstruction instruction I_{rec}, the optimization objective minimizes the negative log-likelihood:

$$\mathcal{L}_{rec}(\theta) = -\log P(D \mid \mathbf{V}, I_{rec}; \theta, \theta_{LLM}) \quad (4)$$

This objective forces the encoder to maximize the information density within $\mathbf{V}$, ensuring that the landmark vectors contain sufficient semantic cues for full-text recovery. The paraphrase pretraining uses a large-scale corpus to develop robust compression capabilities across diverse domains.

Stage 2: Context-Aware Instruction Tuning. In this stage, the encoder is refined to bridge the gap between semantic reconstruction and downstream task utility. While Stage 1 ensures that $\mathbf{V}$ preserves document content, Stage 2 adapts the representations to be compatible with the LLM's instruction-following and reasoning mechanisms. Given a question Q and ground-truth answer A, the optimization objective is:

$$\mathcal{L}_{inst}(\theta) = -\log P(A \mid \mathbf{V}, Q; \theta, \theta_{LLM}) \quad (5)$$

This stage fine-tunes the encoder on question-answering data to ensure the landmark vectors support downstream reasoning. The two-stage strategy enables the encoder to first acquire general compression capabilities before adapting to task-specific requirements.

3 Experimental Setup

We evaluate the native decoding capabilities of frozen LLMs and the effectiveness of our landmark-based compression framework through a two-phase experimental design. The first phase focuses on mechanism probing at the sentence level, establishing the fundamental decoding capability. The second phase evaluates long-context question answering, providing evidence for the practical utility of our method.

3.1 Phase 1: Sentence Reconstruction

This phase aims to verify that frozen LLMs possess native decoding capabilities for continuous latent vectors. We formulate a controlled reconstruction task to evaluate the contribution of our learned representations.

Models and Data. We select four prominent open-source language models as frozen decoders to ensure generalization across architectures: Mistral-7B-v0.3 (Jiang et al., 2023a), Qwen3-8B (Yang et al., 2025), Llama-3.2-1B-Instruct (Grattafiori et al., 2024), and Qwen3-1.7B (Yang et al., 2025). These models span different parameter scales (1B to 8B) and training paradigms, allowing us to assess whether the observed decoding mechanism is model-agnostic. For both training and evaluation, we utilize the Google Conceptual Captions (GCC) dataset (Sharma et al., 2018), which provides concrete and descriptive image captions. This dataset is particularly suitable for reconstruction tasks as its sentences contain diverse semantic content while maintaining moderate length complexity.

Evaluation. To identify the contribution of $\mathbf{v}$ and mitigate pre-training memorization, we implement a prefix-conditioned completion setup. For each sentence, we provide a reconstruction instruction and fix the first 50% of the target sequence as a forced prefix. The model is then required to complete the remaining 50% of the sentence based on the provided latent representation. This design ensures that any performance gain is attributable to the encoded semantic information rather than the model's prior exposure to the dataset.

We compare three distinct experimental conditions: (1) **Baseline**: only the half-text prefix is provided without any latent vector, testing the model's ability to complete sentences based solely on the prefix; (2) **Latent Vector** $\mathbf{v}$: our learned representation generated by the projection network is prepended to the prefix; and (3) **Full-text Reference**: the original embeddings of the entire sentence are provided, serving as an upper bound for reconstruction performance. We report BLEU (Papineni et al., 2002), Token F1, and Perplexity as evaluation metrics.

3.2 Phase 2: Long-Context QA

In this phase, we evaluate the scalability of our landmark-based compression framework for practical long-context applications. We focus on reading comprehension tasks that require understanding and reasoning over extended passages.

Benchmarks. Following the experimental setup of PCC (Dai et al., 2025), we evaluate on four reading comprehension benchmarks: SQuAD v2 (Rajpurkar et al., 2018), which includes unanswerable questions requiring abstention; HotPotQA

(Yang et al., 2018), which tests multi-hop reasoning across paragraphs; AdversarialQA (Bartolo et al., 2020), which contains adversarially constructed questions; and Natural Questions (NQ) (Kwiatkowski et al., 2019), which features real user queries from Google search. This benchmark suite covers extractive QA, multi-hop reasoning, and robustness to adversarial perturbations.

Training Data. For paraphrase pretraining (Stage 1), we follow xRAG (Cheng et al., 2024) and utilize the December 2021 Wikipedia dump (Izacard et al., 2022). This large-scale corpus enables the encoder to develop robust compression capabilities across diverse domains. For context-aware instruction tuning (Stage 2), we follow PCC (Dai et al., 2025) and use a refined subset of SQuAD that provides high-quality question-answer pairs aligned with the Wikipedia passages.

Baselines. Following the experimental setup established by PCC (Dai et al., 2025), we compare against several competitive context compression baselines organized by technical method:

- **Hard compression**: LLMLingua-2 (Pan et al., 2024) performs task-agnostic discrete token pruning based on information-theoretic criteria, representing the state-of-the-art in interpretable compression.
- **Soft compression**: AutoCompressor (Chevalier et al., 2023) utilizes recursive summary vectors for high-ratio compression. xRAG (Cheng et al., 2024) condenses documents into single latent embeddings via modality fusion techniques. ICAE (Ge et al., 2024) and COCOM-Lite (Rau et al., 2024) learn to map text into continuous memory representations. We also include PCC-Lite (Dai et al., 2025) as a state-of-the-art soft compression baseline that employs segment-based encoding.

For baselines other than PCC-Lite, we report published numbers from the PCC reproduction experiments, which evaluate public checkpoints under their released settings rather than retraining all methods with our frozen decoder. For PCC-Lite, we retrain the model using our pretraining corpus and evaluation pipeline to ensure a fair comparison under identical data conditions.

Implementation Details. Our framework employs Qwen3-Emb-0.6B as the token-level encoder, chosen for its strong semantic representation capabilities at a compact parameter scale. For the frozen decoder, we use Mistral-7B. Bidirectional landmark attention masks ensure that each landmark token aggregates global document context without explicit text chunking.

For training, we apply LoRA (Hu et al., 2022) to the encoder's projection layers to enable efficient adaptation while fully training the projection network and landmark token embeddings. All experiments are conducted on NVIDIA A800 GPUs using DeepSpeed optimization for distributed training. We use a learning rate of 1e-4 for paraphrase pretraining and 5e-5 for QA instruction tuning, with a linear warmup phase. During inference, the projected landmark vectors replace reserved placeholder tokens in the LLM's prompt template.

Model	Condition	BLEU	F1	PPL
Qwen3-1.7B	Half-Text	5.90	12.51	135.20
	$\mathbf{v}_c$ + Half	10.76	41.47	12.40
	Full-Text	29.07	61.68	1.07
Llama3.2-1B	Half-Text	5.47	12.28	101.86
	$\mathbf{v}_c$ + Half	27.19	55.28	4.87
	Full-Text	29.66	62.36	1.45
Qwen3-8B	Half-Text	7.01	12.73	79.66
	$\mathbf{v}_c$ + Half	21.87	52.64	6.26
	Full-Text	28.09	62.44	1.03
Mistral-7B	Half-Text	8.39	13.16	49.89
	$\mathbf{v}_c$ + Half	35.06	65.44	3.38
	Full-Text	34.43	67.25	1.23

Table 1: Reconstruction performance on the GCC dataset. $\mathbf{v}_c$ denotes the latent vector generated by our projection network.

4 Results and Analysis

4.1 Sentence-Level Reconstruction

We first evaluate the sentence-level reconstruction task to verify that frozen LLMs possess native decoding capabilities for continuous latent vectors. Table 1 presents the reconstruction performance across four frozen decoders under three conditions: baseline with prefix only, our latent vector $\mathbf{v}_c$, and full-text reference.

The results demonstrate that frozen LLMs can decode substantial semantic information from a single continuous vector. Across all models, incorporating the latent vector $\mathbf{v}_c$ significantly improves performance compared to the baseline. Notably, Mistral-7B achieves 65.44 Token F1 with $\mathbf{v}_c$, recovering over 97% of the full-text reference performance. This establishes the existence of a decodable manifold within the frozen LLM's embedding

Rate	Method	SQuAD		HotPotQA		AdversarialQA		NQ		Average	
		F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Reference	w/o Context	21.23	7.74	29.01	18.21	15.30	8.13	34.59	21.37	25.03	13.86
	w/ Context	73.15	47.90	65.57	52.65	45.08	25.40	56.24	41.78	60.01	41.93
Variable	AutoCompressor	21.46	0.35	16.29	0.29	14.09	2.00	25.57	0.63	19.35	0.82
	xRAG	18.19	3.46	27.51	16.29	13.75	3.47	38.06	20.80	24.38	11.01
	ICAE	45.69	21.63	35.16	26.68	27.98	11.70	59.15	47.35	42.00	26.84
	LLMLingua2	51.20	32.18	**55.72**	**44.18**	35.41	24.80	**68.44**	**55.85**	**53.50**	**39.25**
4x	COCOM-Lite	21.70	9.17	40.07	32.32	19.45	13.90	50.45	41.87	32.92	24.32
	PCC-Lite	46.67	30.32	37.89	29.88	35.02	25.43	63.73	55.01	45.83	35.16
	Ours	**72.96**	**49.37**	36.02	27.70	**43.49**	**26.93**	61.52	49.39	**53.50**	38.35
16x	COCOM-Lite	19.23	8.13	31.94	25.27	19.35	14.73	26.36	20.66	24.22	17.20
	PCC-Lite	45.32	27.60	35.82	27.73	32.04	22.23	61.15	52.97	43.58	32.63
	Ours	57.38	34.36	36.16	27.34	34.88	19.93	52.04	41.25	45.12	30.72

Table 2: Performance comparison on reading comprehension benchmarks. Metrics are F1 and Exact Match (EM). The best and second-best results among compression methods, excluding reference rows, are shown in **bold** and underlined, respectively. Tied best results are both bolded.

Prefix	R1lmk/R1base	RLlmk/RLbase
0.0	0.658 / 0.092	0.549 / 0.082
0.1	0.651 / 0.113	0.545 / 0.098
0.2	0.632 / 0.121	0.540 / 0.106
0.3	0.631 / 0.144	0.563 / 0.132
0.4	0.620 / 0.147	0.557 / 0.136
0.5	0.586 / 0.149	0.538 / 0.138

Table 3: Prefix-conditioned reconstruction on 100 GCC samples. The landmark condition consistently outperforms the prefix-only baseline, indicating that the compressed vector carries semantics beyond local continuation cues.

space.

To further test whether the decoder is merely exploiting prefix continuation, we evaluate a stronger prefix-conditioned probe in which the model receives both the compressed vector and prefixes ranging from 0% to 50% of the original sentence, and we score only the withheld suffix. Table 3 shows that the landmark condition substantially outperforms the prefix-only baseline across all prefix ratios. Even at 0% prefix, a single compressed vector yields 0.658 ROUGE-1, supporting the claim that the vector itself carries substantial semantic content rather than acting as a shortcut for local text continuation.

4.2 Document-Level QA Performance

Scaling to the document level, we evaluate our landmark-based compression framework on four reading comprehension benchmarks. Table 2 presents the comparison with baseline methods at 4x and 16x compression ratios. The training dynamics (Figure 3) show that the mean L2 norm of landmark vectors converges to values substantially higher than standard vocabulary embeddings, with 16x compression converging to approximately twice the norm of 4x, reflecting denser information encoding.

Our method demonstrates strong but mixed performance across benchmarks. At 4× compression, it is strongest on SQuAD and AdversarialQA and reaches the highest average F1 among soft-compression methods, but PCC-Lite remains stronger on HotPotQA and NQ. At 16× compression, our method substantially outperforms COCOM-Lite and other global soft-compression baselines, while remaining below PCC-Lite overall. All QA scores are from a single trained checkpoint evaluated with deterministic greedy decoding, so test-time outputs are fixed while training-run variance remains unmeasured.

4.3 The High-Norm Phenomenon

A consistent pattern across all successful reconstruction experiments is the emergence of excep-

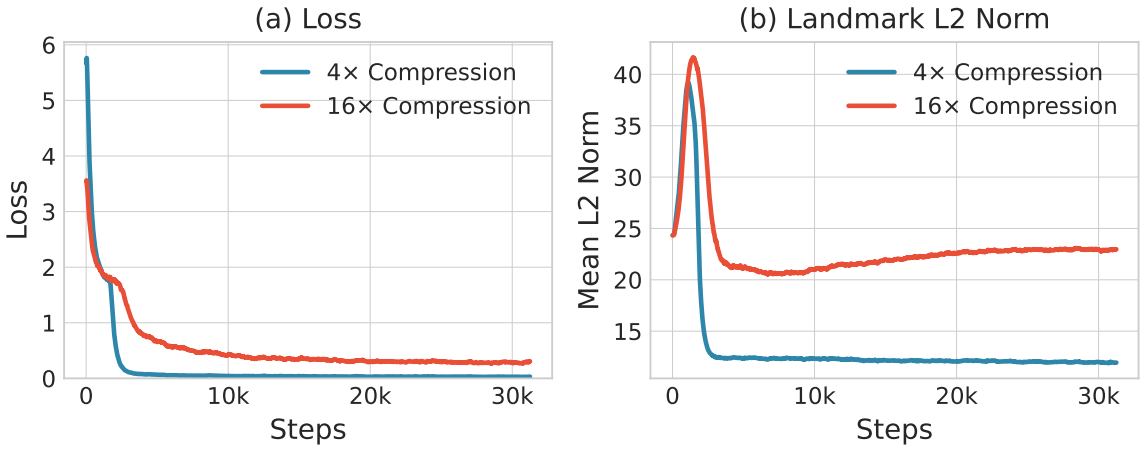

Figure 3: Training dynamics of landmark-based compression at 4× and 16× compression ratios. (a) Training loss curves showing convergence patterns. (b) Mean L2 norm of landmark vectors, which rises initially before decreasing and converging. The 16× ratio converges to a higher norm than 4×, reflecting denser information encoding per vector.

Model	Standard Token	Latent Vector
Qwen3-1.7B	1.54	299.08
Llama3.2-1B	0.93	199.03
Qwen3-8B	1.38	278.71
Mistral-7B	0.17	235.56

Table 4: Mean L2 norms of standard vocabulary token embeddings and learned latent vectors across models.

tionally high L2 norms in the projected vectors. Table 4 quantifies this phenomenon: the learned latent vectors exhibit norms that are two orders of magnitude higher than standard vocabulary embeddings across all tested models. For example, Qwen3-1.7B produces latent vectors with mean L2 norm of 299.08, compared to 1.54 for standard token embeddings—a ratio of nearly 200x.

We hypothesize that this high-norm property serves as a geometric signaling mechanism, allowing the frozen LLM to distinguish information-dense vectors from standard embeddings during self-attention computation. In this revision, we frame this as an enabling factor rather than a fully isolated mechanism.

Intervention via Norm Scaling. To probe the role of high L2 norm, we perform an inference-time intervention. We scale the magnitude of the landmark vectors while preserving their direction, sweeping the scale factor from 0.1 to 1.0. Figure 4 shows the results on QA performance at 4x compression.

The results reveal a clear intervention effect between norm magnitude and decoding performance.

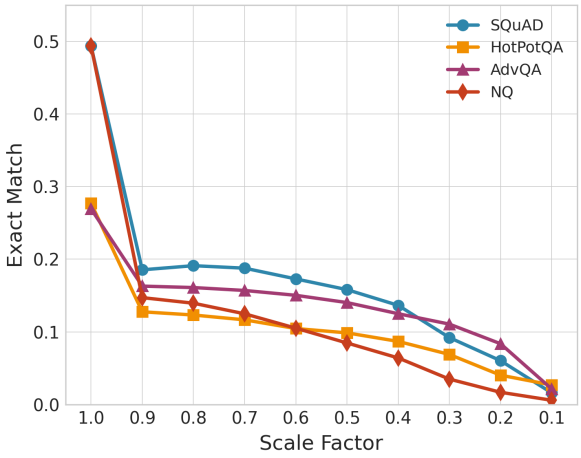

Figure 4: Causal effect of the landmark vector's L2 norm on QA performance (4x compression). The x-axis shows the scaling factor applied to the vector's original norm, where 1.0 represents the unmodified vector.

All four datasets show consistent degradation as the scale factor decreases from 1.0 to 0.1, with SQuAD and NQ exhibiting particularly steep drops. At scale factor 0.1, performance collapses to near-zero levels across all benchmarks. We therefore view high magnitude as strong evidence for an enabling condition of successful decoding, while leaving open whether the underlying mechanism is attention dominance, low-norm suppression, or both.

Attention Pattern Analysis. To understand how the frozen LLM processes high-norm vectors internally, we analyze attention patterns across decoder layers. Figure 5 visualizes the attention weights and activation norms for representative ini-

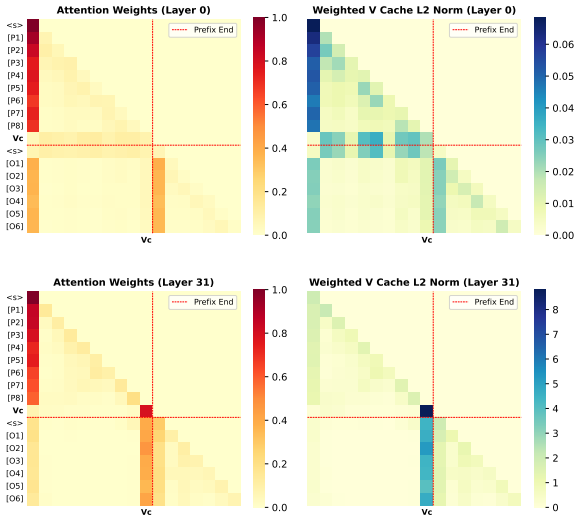

Figure 5: Attention and activation analysis across decoder layers. The sequence is partitioned into the prompt ([P]), the latent vector ($\mathbf{v}_c$), and the output ([O]).

Ratio	Attention	Avg. F1	Avg. EM
4x	Causal	32.57	21.90
	Bidirectional	**53.50**	**38.35**
16x	Causal	31.61	20.86
	Bidirectional	**45.12**	**30.72**

Table 5: Ablation study on landmark attention mask. Bidirectional attention enables landmarks to capture global context.

tial (Layer 0) and final (Layer 31) layers.

The attention patterns confirm that $\mathbf{v}_c$ serves as the primary information source during generation. Output tokens consistently assign high attention weights to the latent vector, with this focus becoming more pronounced in deeper layers. The weighted V-cache norms reveal that the influence of $\mathbf{v}_c$ is amplified in final layers. This evidence is consistent with a norm-induced prioritization effect, although it does not by itself rule out decoder insensitivity to lower-norm vectors.

4.4 Ablation of Bidirectional Attention

The strong performance of our method in long-context scenarios stems from the integration of the high-norm mechanism with bidirectional landmark attention. To isolate the impact of global contextual dependencies, we conduct an ablation study on the landmark attention mask.

Table 5 shows that landmark vectors with bidirectional visibility significantly outperform those constrained by causal masks. At 4x compression, bidirectional attention improves average F1 by over 20 absolute points. This validates that landmarks function as global semantic anchors aggregating document-wide information, rather than local summaries.

5 Related Work

Non-Vocabulary Vectors in LLMs Recent research demonstrates that LLMs can process information encoded in continuous, non-vocabulary vectors. Parameter-efficient fine-tuning methods like Prefix-Tuning (Li and Liang, 2021) and prompt tuning (Lester et al., 2021) steer frozen LLMs via learnable "soft prompts." Models such as CLIP (Radford et al., 2021) show that visual inputs can be aligned with language representations, motivating later work that connects non-text embeddings to language-model interfaces.

Further studies expand this understanding: Task Arithmetic (Ilharco et al., 2023) manipulates model behaviors via arithmetic operations on task vectors, while SelfIE (Chen et al., 2024) explores self-interpretability of LLM embeddings. These works suggest LLMs possess a latent capacity to process continuous information beyond standard tokens.

Context Compression Processing lengthy token sequences is computationally expensive, motivating methods that compress text into compact vector representations. Early work learns to summarize long contexts into memory vectors, such as AutoCompressors (Chevalier et al., 2023). Subsequent methods push the "vector-in, text-out" paradigm further: ICAE compresses contexts into continuous slots (Ge et al., 2024), while xRAG injects each document as approximately a single token (Cheng et al., 2024). Recent works systematize this compressor–LLM interface. PCC proposes a decoupled compressor with practical compression ratios (4×–16×) (Dai et al., 2025). Hybrid methods combine soft global compression with hard token selection (Liao et al., 2025), while REFRAG targets the decoding stack by replacing retrieved spans with chunk embeddings (Lin et al., 2025).

Geometric Properties of Text Embeddings A parallel stream of research analyzes the geometry of embedding spaces, offering theoretical grounding. Prior work links the norm of static embeddings to information gain (Oyama et al., 2023) and the norm–variance trade-off in contextualized embeddings to semantic dispersion (Yamagiwa and Shimodaira, 2025). While these studies focus on

embeddings representing words or short spans, we extend the inquiry to complex, sentence-level vectors produced by a compressor. Rather than only describing existing embedding geometry, we show how a model can learn to manipulate geometric properties as an active mechanism that makes compressed vectors decodable by a frozen LLM.

6 Conclusion

This work investigates the capacity of frozen LLMs to decode continuous semantic representations. Through controlled experiments, we identify high L2 norm as a critical geometric signal associated with successful decoding of information-dense vectors, with intervention-based evidence that it is an enabling factor for frozen decoders. Building on this finding, we propose a landmark-based, chunking-free compression method that leverages bidirectional attention to aggregate global document context into compact representations. Experiments on text reconstruction and four reading comprehension benchmarks demonstrate strong but mixed performance at 4x and 16x compression ratios, with the clearest gains over global soft-compression baselines and on selected QA datasets. These results support both the empirical insight and the practical utility of our method for long-context compression.

Limitations

Despite the empirical and mechanistic insights provided by this work, several limitations remain that suggest directions for future inquiry:

- Model Parameter Rigidity: A primary constraint of our framework is the exclusive use of frozen LLM parameters. While this design choice was essential to isolate and analyze the model's native decoding capabilities, it potentially overlooks the performance gains that could be achieved through synergistic optimization. Future research may explore the trade-offs between parameter-efficient probing and joint fine-tuning of the encoder and decoder to enhance reconstruction precision.

- Mechanistic Ambiguity: Our norm-scaling intervention shows that reducing vector magnitude sharply degrades decoding quality, but it does not fully separate whether the dominant effect is increased attention mass, reduced sensitivity to low-norm vectors, or a combination of the two.

- Uniform Compression Granularity: Our current landmark-based strategy employs a static compression rate across the entire document. However, textual information is rarely distributed uniformly, and the fixed-stride method may be suboptimal for segments with varying information entropy. Transitioning from a static stride to an adaptive sampling mechanism that allocates latent capacity based on local semantic density represents a promising avenue for improving compression fidelity.

- Breadth of Task Generalization: Our evaluation is primarily focused on verbatim semantic reconstruction and extractive question answering, and the QA results come from a single training run per setting. While these tasks effectively probe the information density of the compressed representations, the utility of high-norm latent vectors in scenarios requiring complex logical synthesis, long-form creative generation, or symbolic reasoning remains to be fully characterized.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2024YFF0907404.

References

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. 2024. SelfIE: Self-interpretation of large language model embeddings. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 7373–7388. PMLR.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. In *Advances in Neural Information Processing Systems*, volume 37, pages 109487–109516. Curran Associates, Inc.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore. Association for Computational Linguistics.

Yuhong Dai, Jianxun Lian, Yitian Huang, Wei Zhang, Mingyang Zhou, Mingqi Wu, Xing Xie, and Hao Liao. 2025. Pretraining context compressor for large language models with embedding-based memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28715–28732, Vienna, Austria. Association for Computational Linguistics.

Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models. *Preprint*, arXiv:2208.03299.

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. 2023a. Mistral 7b. *Preprint*, arXiv:2310.06825.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.

Huanxuan Liao, Wen Hu, Yao Xu, Shizhu He, Jun Zhao, and Kang Liu. 2025. Beyond hard and soft: Hybrid context compression for balancing local and global information retention. *Preprint*, arXiv:2505.15774.

Xiaoqiang Lin, Aritra Ghosh, Bryan Kian Hsiang Low, Anshumali Shrivastava, and Vijai Mohan. 2025. REFRAG: Rethinking RAG based decoding. *Preprint*, arXiv:2509.01092.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Momose Oyama, Sho Yokoi, and Hidetoshi Shimodaira. 2023. Norm of word embedding encodes information gain. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2108–2130, Singapore. Association for Computational Linguistics.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

David Rau, Shuai Wang, Hervé Déjean, and Stéphane Clinchant. 2024. Context embeddings for efficient answer generation in rag. *Preprint*, arXiv:2407.09252.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hiroaki Yamagiwa and Hidetoshi Shimodaira. 2025. Norm of mean contextualized embeddings determines their variance. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7778–7808, Abu Dhabi, UAE. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 Technical Report. *Preprint*, arXiv:2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

A Dataset Statistics

Table 6 summarizes the datasets used in our experiments. For sentence-level reconstruction, we use the Google Conceptual Captions (GCC) dataset. For document-level QA, pretraining is conducted on the English Wikipedia dump from December 2021, and finetuning uses the SQuAD training set.

Task	Dataset	Train	Test
Sentence-Level Reconstruction			
Reconstruction	GCC	3,318,333	15,840
Document-Level QA			
Pretraining	enwiki-dec2021	3,176,581	–
Finetuning	SQuAD	86,821	5,928

Table 6: Dataset statistics for sentence-level reconstruction and document-level QA experiments.

B Full Attention Visualization

To supplement the analysis in the main body, this section provides the complete visualizations for attention weights (Figure 6) and attention-weighted V-cache L2 norms (Figure 7) across all 32 decoder layers. These figures illustrate the consistent behavior of the model: the complex semantic embedding ($\mathbf{v}_c$) is established as the primary information source from the earliest layers and its influence is maintained or amplified throughout the network, confirming that the phenomena observed in Layers 0 and 31 are representative of the model's general processing dynamic.

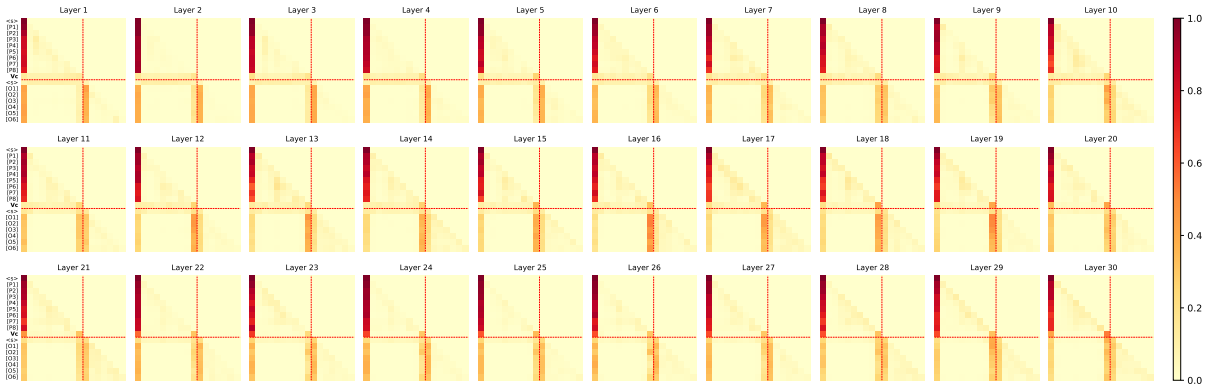

Figure 6: Attention visualization for the complex embedding ($\mathbf{v}_c$) across all decoder layers.

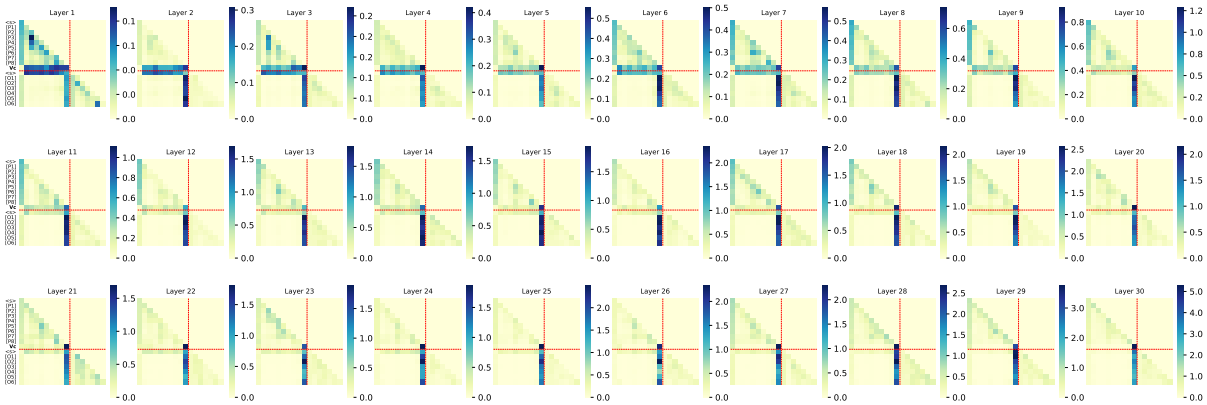

Figure 7: Attention-weighted V-cache norm visualization for the complex embedding ($\mathbf{v}_c$) across all decoder layers.

C Scaling Examples

This section presents qualitative examples for the inference-time norm scaling experiment. Table 7 shows the reconstructed text generated from a complex vector whose norm was scaled by different factors (α). These examples correspond to the quantitative results shown in Figure 4 and illustrate how reconstruction fidelity degrades as the norm is artificially reduced.

The target text for this example is: "marking our way through the dunes as we neared the coast".

α	**Generated Text**
0.2	the dunes leading us to the coast as we made our way back from the markers
0.4 - 0.6	meandering our way down the dunes towards the markers on the other side of the bay
0.8 - 1.6	meandering our way down the dunes towards the markers on the other side
1.8 - 2.0	meandering our way down the dunes towards the coast as the sun set

Table 7: Reconstructed text from a complex vector whose norm was scaled by different factors (α).

D Interpolation Examples

This section provides the qualitative results for the latent space interpolation experiment discussed in the main analysis. Table 8 shows the text generated by the frozen LLM when decoding vectors that are linearly interpolated between the complex embeddings of two source sentences, D_1 and D_2. As the interpolation factor λ shifts from 0.0 to 1.0, the output text exhibits a smooth and semantically coherent transition from the content of Sentence 1 to that of Sentence 2.

λ	**Generated Text**
0.0	flying over the mountains and lake
0.1	flying over the mountains and lake
0.2	flying over the mountains and lake
0.3	flying over the mountains and lake
0.4	flying over the mountains and vineyards
0.5	flying over the wine tasting vineyards
0.6	wine tasting in the mountains
0.7	wine tasting in the beautiful surroundings
0.8	wine tasting in the sunny weather
0.9	wine tasting in the sunny weather
1.0	wine tasting in the sunshine

Table 8: Generated text at different interpolation factors (λ) between two sentences. At $\lambda = 0$, the output corresponds to "flying over the mountains and lake"; at $\lambda = 1$, it corresponds to "wine tasting in the sunshine".

E Pretraining Reconstruction Example

This section presents qualitative examples from the document-level pretraining stage. Given a Wikipedia passage about the Society of Jesus (Jesuits) in the United States, the model compresses it into landmark tokens at different compression ratios and attempts to reconstruct the original document. Words in red indicate deviations from the ground truth, identified using word-level edit distance alignment.

The comparison reveals a clear quality-compression trade-off. At 4$\times$ compression ratio with 41 landmark tokens, the reconstruction achieves perfect fidelity with no errors. At 16$\times$ compression ratio

with only 11 tokens, the model preserves the overall topic and many key entities but introduces factual and temporal errors, such as: (1) "geographic" → "geographical"; (2) "provincial superior" → "superior general"; (3) "early 21st century" → "early 2021"; and (4) reordering or renaming province names.

F Single-Token Compression Probe

Table 10 provides a finer-grained view of the decoder dynamics under a single-token compression bottleneck. Across layers, the compressed landmark maintains a substantially larger hidden-state norm than ordinary text positions, with the gap being most pronounced in early layers. At the same time, the Value norms remain broadly comparable between landmark and text tokens.

The table also shows a clear difference in how the decoder allocates attention and contribution magnitude. From middle layers onward, the landmark receives substantially more attention mass than ordinary text positions, and its attention-weighted Value norm is correspondingly larger across layers 4–31. Taken together, these measurements show that the landmark becomes a prominent retrieval position during decoding while preserving Value vectors of ordinary scale.

G Full Efficiency Tables

Tables 11–13 provide the full latency accounting for the compression pipeline. They make explicit that the reported 4×/16× ratios are decoder-side compression ratios: the encoder still processes the full document, while the decoder consumes a reduced landmark sequence. This accounting clarifies the relationship between encoder tokens, landmark tokens, decoder tokens, and end-to-end latency.

Across buckets, decoder prefill decreases as the compressed representation becomes shorter, while the one-time compression step stays relatively stable. For the [512,800) bucket, full-context TTFT is 66.2, compared with 63.0 at 4× and 48.9 at 16×. If compression is precomputed offline, the corresponding 16× prefill cost is 24.3. For shorter contexts, the compression step constitutes a larger share of the total latency budget.

H Document-Level QA Hyperparameters

Table 9 lists the hyperparameter settings used for the document-level QA experiments. The pretraining and finetuning stages share the same encoder length, LoRA configuration, landmark token, and projector architecture, while using different learning rates, batch sizes, gradient accumulation settings, and numbers of epochs to match their distinct training objectives.

Category	Hyperparameter	Pretraining	Finetuning
Data	Max Training Samples	2,000,000	–
	Max Sequence Length	512	512
	Encoder Max Length	512	512
LoRA	Rank	64	64
	Alpha	32	32
	Dropout	0.1	0.1
Architecture	Landmark Token	`[control_128]`	`[control_128]`
	Projector Depth	2	2
	Projector Hidden Size	4096	4096
Training	Optimizer	AdamW	AdamW
	LR Scheduler	Linear	Linear
	Learning Rate	1e-4	5e-5
	Warmup Ratio	0.03	0.03
	Batch Size (per device)	2	4
	Gradient Accumulation	8	4
	Training Epochs	1	3
	Gradient Clipping	1.0	1.0

Table 9: Hyperparameter configurations for the pretraining and finetuning stages of document-level QA experiments using Mistral-7B as the frozen decoder.

Figure 8 provides the full qualitative example for Appendix E, including the prompt format, target passage, 4× reconstruction, and 16× reconstruction with word-level deviations highlighted.

Prompt (4×): `[INST] Use these compressed markers as hints: [control_128] × 41 Please recover the original document faithfully. [/INST]`

Prompt (16×): `[INST] Use these compressed markers as hints: [control_128] × 11 Please recover the original document faithfully. [/INST]`

Target:

In the United States, the order is best known for its missions to the Native Americans in the early 17th century, its network of colleges and universities, and (in Europe before 1773) its politically conservative role in the Catholic Counter Reformation. The Society of Jesus, in the United States, is organized into geographic provinces, each of which being headed by a provincial superior. Today, there are five Jesuit provinces operating in the United States: the USA East, USA Central and Southern, USA Midwest, and USA West Provinces. At their height, there were ten provinces. Though there had been mergers in the past, a major reorganization of the provinces began in early 21st century, with the aim of consolidating into four provinces by 2020.

Reconstructed (4×):

In the United States, the order is best known for its missions to the Native Americans in the early 17th century, its network of colleges and universities, and (in Europe before 1773) its politically conservative role in the Catholic Counter Reformation. The Society of Jesus, in the United States, is organized into geographic provinces, each of which being headed by a provincial superior. Today, there are five Jesuit provinces operating in the United States: the USA East, USA Central and Southern, USA Midwest, and USA West Provinces. At their height, there were ten provinces. Though there had been mergers in the past, a major reorganization of the provinces began in early 21st century, with the aim of consolidating into four provinces by 2020.

Reconstructed (16×) (errors in red)**:**

In the United States, the order is best known for its missions to the Native Americans in the early 17th century, its network of colleges and universities, and (in Europe before 1773) its politically conservative role in the Catholic Counter Reformation. The Society of Jesus, in the United States, is organized into a series of geographical provinces, each headed by a superior general. There are five Jesuit provinces operating today in the USA: the East USA, the South USA, Central USA, Midwest USA and Western USA Provinces. At their height, there were ten provinces. Though there had been mergers in the past, a major reorganization of the provinces began in early 2021, with the aim of consolidating into four provinces by 2020.

Figure 8: Document reconstruction examples at 4× and 16× compression ratios. The 4× reconstruction (41 landmark tokens) achieves perfect fidelity, while the 16× reconstruction (11 tokens) preserves the overall topic but introduces factual and temporal errors highlighted in red.

Table 10 provides the detailed measurements for the single-token compression probe discussed above, including hidden-state norms, projected Q/K/V norms, attention mass, and attention-weighted Value norms.

Layer	HS_{lmk}	HS_{text}	Q_{lmk}	Q_{text}	K_{lmk}	K_{text}	V_{lmk}	V_{text}	$Attn_{lmk}$	$Attn_{text}$	wV_{lmk}	wV_{text}
0	32.6	0.1	2.9	11.7	3.4	12.2	0.2	0.2	0.040	0.038	0.009	0.008
4	32.6	13.6	4.6	11.3	6.7	15.7	1.9	2.3	0.068	0.008	0.132	0.007
8	32.9	14.7	5.6	11.2	8.8	17.5	2.8	3.0	0.191	0.007	0.520	0.016
12	33.1	16.2	7.5	11.1	11.4	16.8	3.8	3.6	0.284	0.006	1.026	0.019
16	33.3	18.4	8.0	10.8	12.9	16.7	4.4	4.1	0.460	0.007	2.024	0.028
20	34.8	23.7	8.8	11.8	12.9	16.4	4.9	5.8	0.351	0.008	1.731	0.033
24	35.5	28.5	10.1	11.7	13.2	15.5	5.9	6.6	0.214	0.009	1.187	0.025
28	36.7	32.8	10.8	11.8	13.4	15.0	7.3	8.1	0.243	0.011	1.775	0.051
31	39.1	34.5	11.9	13.3	13.7	14.5	9.4	10.2	0.385	0.010	3.602	0.082

Table 10: Single-token compression probe on 50 GCC samples with the frozen Mistral-7B decoder. HS/Q/K/V denote hidden-state and projected Q/K/V L2 norms. Attn is the attention mass assigned to the compressed landmark versus ordinary text positions, and wV is the attention-weighted Value norm averaged across 8 KV heads.

Tables 11–13 provide the complete latency measurements for the full-context baseline and the two compression settings.

Bucket	N	Enc Tokens	Dec Tokens	Compress	Prefill	TTFT
[0,64)	70	48.5	71.7	–	22.2	22.2
[64,128)	1545	109.6	138.8	–	25.3	25.3
[128,192)	2581	155.7	187.9	–	27.1	27.1
[192,256)	1060	216.3	253.0	–	31.2	31.2
[256,384)	557	298.6	343.1	–	36.3	36.3
[384,512)	76	437.7	486.8	–	45.7	45.7
[512,800)	39	638.0	710.0	–	66.2	66.2

Table 11: Full-context latency accounting on the SQuAD test set grouped by context length. Measurements were collected on a single NVIDIA A800-80GB GPU with FP16 precision.

Bucket	N	Enc Tokens	Lmk Tokens	Dec Tokens	Compress	Prefill	TTFT
[0,64)	70	48.5	11.7	68.4	24.0	22.1	46.1
[64,128)	1545	109.6	27.0	98.9	24.2	22.6	46.8
[128,192)	2581	155.7	38.6	121.9	24.3	23.5	47.8
[192,256)	1060	216.3	53.7	152.3	24.3	26.1	50.4
[256,384)	557	298.6	74.3	193.4	25.4	26.8	52.2
[384,512)	76	437.7	109.1	265.5	27.1	32.4	59.5
[512,800)	39	638.0	159.0	368.0	25.1	37.9	63.0

Table 12: Deployment-style latency accounting for our $4\times$ compression setting. The encoder still reads the full document, while the decoder consumes only the landmark sequence.

Bucket	N	Enc Tokens	Lmk Tokens	Dec Tokens	Compress	Prefill	TTFT
[0,64)	70	48.5	2.4	49.9	24.2	22.1	46.3
[64,128)	1545	109.6	6.4	57.6	24.2	22.0	46.2
[128,192)	2581	155.7	9.3	63.3	24.2	22.1	46.4
[192,256)	1060	216.3	13.1	71.0	24.3	22.2	46.5
[256,384)	557	298.6	18.2	81.3	25.3	22.8	48.1
[384,512)	76	437.7	26.9	101.0	26.9	23.4	50.2
[512,800)	39	638.0	40.0	129.0	24.7	24.3	48.9

Table 13: Deployment-style latency accounting for our $16\times$ compression setting. For long contexts, the prefill reduction outweighs the one-time compression overhead.