Punctuation-aware Hybrid Trainable Sparse Attention for Large Language Models

Junxiang Qiu[1], Shuo Wang[1,*], Zhengsu Chen[2], Hengheng Zhang[2]
Jinda Lu[1], Changcheng Li[1], Xinting Hu[1], Qi Tian[2,*]

[1]University of Science and Technology of China
[2]Huawei Inc.

{qiujx, lujd, lichangcheng}@mail.ustc.edu.cn, shuowang.edu@gmail.com
{chenzhengsu2, zhanghengheng55, tian.qi1}@huawei.com, xinting@ustc.edu.cn

Abstract

Attention serves as the fundamental mechanism for long-context modeling in large language models (LLMs), yet dense attention becomes structurally prohibitive for long sequences due to its quadratic complexity. Consequently, sparse attention has received increasing attention as a scalable alternative. However, sparse attention can miss relevant context, especially when very few tokens are selected. One possible reason is that averaging token representations within a block can dilute signals from important tokens, making relevant context harder to identify. To address this issue, we propose **P**unctuation-aware **H**ybrid **S**parse **A**ttention (**PHSA**), a trainable sparse attention mechanism. PHSA uses punctuation tokens, which often mark the ends of clauses and sentences, as an additional source of contextual information. It combines the average representation of all tokens in each block with a separate average over its punctuation tokens to guide context selection. We further study how training sparsity affects performance when only a small fraction of tokens is selected during inference. Experiments show that PHSA achieves a higher average LongBench score than both dense attention and InfLLM v2, while reducing retrieval errors by 10.8% relative to InfLLM v2 at 97.3% sparsity on 32k-token inputs.

1 Introduction

Recent advances in large language models (LLMs) and foundation models Achiam et al. [2023] have profoundly broadened their application frontiers, spanning multimodal and personalized generation Qiu et al. [2025b], Fang et al. [2026, 2025], Li et al. [2026c], Xu et al. [2025b,a, 2026], rigorous reasoning and capability evaluation Jiang et al. [2026a, 2025a], Wang et al. [2026b,a], as well as trustworthy and safety-aligned deployment Li et al. [2026a], Jiang et al. [2025b], Zhu et al. [2026b,a,c], Wang et al. [2025], Li et al. [2026b]. Across these expanding domains, real-world deployment critically hinges on computational efficiency, spurring active exploration into eliminating computational redundancy and accelerating inference Qiu et al. [2025a,c], Jiang et al. [2026b], Fu et al. [2025], Fan et al. [2026]. Within this landscape, long-context processing Bai et al. [2024], Li et al. [2024c] has emerged as an indispensable capability for complex workflows such as long-document analysis, agent interaction, and repository-level code synthesis. At the architectural core of modern LLMs, however, long-sequence modeling is fundamentally bottlenecked by the attention mechanism Vaswani et al. [2017], whose scalability directly dictates the computational footprint and serving throughput of long sequences.

However, for a text sequence of length L, the dense attention mechanism requires each token to attend to all preceding tokens, resulting in a computational complexity of $O(L^2)$. As shown in the blue bars

*Shuo Wang and Qi Tian are the corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

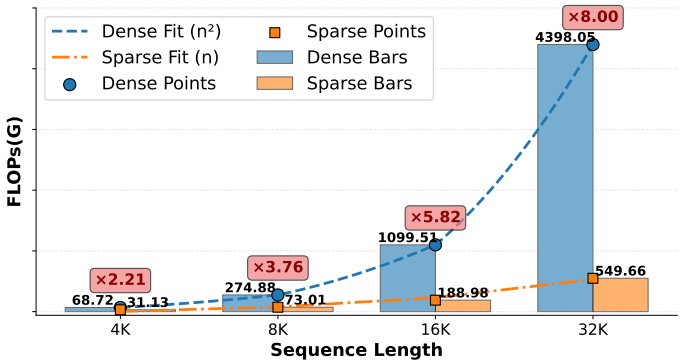


Figure 1: Comparison of the computational complexity of single-layer attention under dense (blue) and sparse (orange) settings for different sequence lengths.

in Figure 1, as L increases, both inference latency and memory footprint grow rapidly, forming the primary hardware bottleneck for long-sequence processing. To overcome this limitation, recent works have focused on sparse attention mechanisms Jiang et al. [2024], Zhang et al. [2023], Yuan et al. [2025], Zhao et al. [2025] that restrict attention computation to a subset of critical query-key pairs, reducing the overall complexity to $O(L)$. By avoiding redundant computations on irrelevant tokens, sparse attention significantly reduces computational and memory overheads, while also enhancing the model's semantic focusing capability through native training. As illustrated in the orange bars in Figure 1, sparse attention exhibits substantially better scalability with respect to sequence length, and its efficiency advantage becomes increasingly pronounced in long-context regimes. Meanwhile, the latest research indicates that sparse attention mechanisms with native training not only avoid inevitable degradation in model performance, but also have the potential to yield performance gains. For example, DeepSeek's NSA Yuan et al. [2025] and MiniCPM's InfLLM v2 Zhao et al. [2025] employ sparse attention training-inference mechanism, and report strong performance on long-context benchmarks.

Despite the aforementioned progress, existing sparse attention methods still face two key limitations that restrict their practical applicability and limit further gains in model capability. First, during the key and value selection stage, existing methods aggregate consecutive tokens into a single representative vector through average pooling to reduce the computational and sorting overhead of query-key relevance estimation. This coarse-grained aggregation inevitably obscures fine-grained semantic distinctions within each block. Semantically critical elements such as key entities or logical hubs may be diluted by irrelevant tokens, leading to the loss of critical semantics and ultimately producing suboptimal selection results. Second, current sparse attention methods exhibit pronounced performance degradation under extremely low selection settings, which hinders the exploration of their full potential in the regime of extreme sparsity. For example, experiments on NIAH Hsieh et al. [2024] show severe information loss when the number of activated tokens is aggressively reduced. For instance, NSA recommends activating 3584 tokens (corresponding to approximately 88.8% sparsity) in 32k long-sequence scenarios, while more aggressive sparsity regimes remain largely unexplored.

Motivated by the observation in SepLLM Chen et al. [2024a] and LLM-Microscope Razzhigaev et al. [2025] that punctuation tokens (*e.g.*, commas, periods, and semicolons) encode inherent semantic boundary information, we leverage their boundary-anchoring property to address the information loss caused by coarse-grained block aggregation. Punctuation tokens naturally divide text into logically coherent semantic segments and mark key semantic turning points across segments, providing a structural cue for preserving fine-grained semantics. Based on this insight, we design a dual-branch representative token aggregation mechanism for block-level token selection, in which a punctuation-aware mechanism is explicitly incorporated. This mechanism preserves critical semantic boundary information during block aggregation while avoiding the dilution of important features, thereby mitigating the semantic degradation introduced by average pooling. We refer to this punctuation-aware aggregation process as the core component of **P**unctuation-aware **H**ybrid **S**parse **A**ttention **(PHSA)**. Meanwhile, beyond improving the accuracy of representative token selection, we further explore the feasibility of training and generation under extreme sparsity, which is a critical factor for achieving highly efficient sparse attention. Building upon the punctuation-aware sparse attention

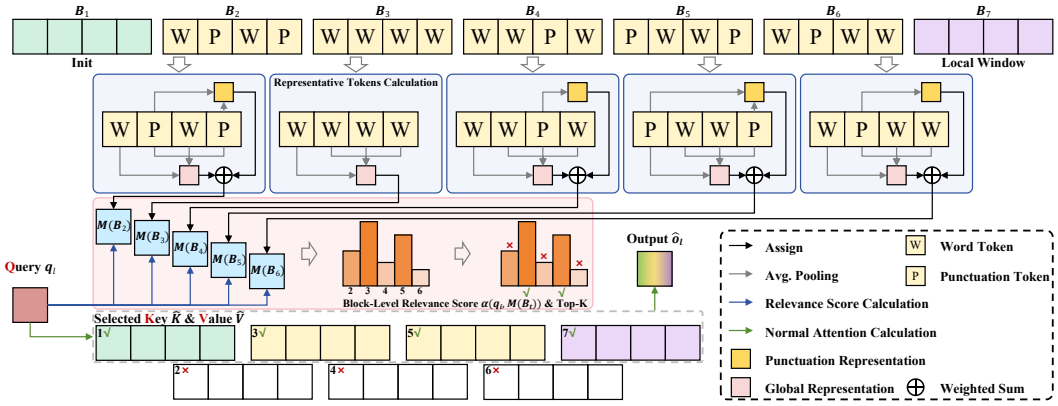


Figure 2: Pipeline of PHSA. By splitting key blocks, we first compute the dual-branch representative tokens for each block, then calculate the relevance score with the query. Finally, we select the Top-K blocks, which, together with the initial block and local window blocks, form the to-be-computed indices for sparse attention.

mechanism, we construct an extreme-sparsity-oriented training and inference framework that enables stable model behavior with a very small number of activated tokens. This framework substantially reduces computational and memory overhead on resource-constrained devices, breaks the efficiency limitations of existing methods, and expands the deployment scope of long-context processing.

In summary, the contributions of our **PHSA** are threefold:

- We design a punctuation-aware hybrid aggregation mechanism, which adopts a dual-branch aggregation mechanism to preserve critical semantic boundary information and improve the semantic accuracy of Top-K block selection.
- We propose an adaptable training framework for extreme sparsity, systematically exploring lower activated token ratios, quantitatively analyzing performance boundaries, and providing theoretical support for lightweight deployment.
- PHSA outperforms both dense attention and state-of-the-art baseline in general benchmarks and long-context benchmarks under both training and training-free scenarios.

2 Method

This section elaborates on the structure of PHSA in detail. The subsequent subsections first introduce the preliminaries of our method, followed by a detailed exposition of the overall framework and core algorithms of PHSA.

2.1 Preliminaries

In each dense attention layer, the input sequence $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_L\}$ is first mapped to query vectors $\boldsymbol{Q} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_L\} \in \mathbb{R}^{L \times d_\text{k}}$, key vectors $\boldsymbol{K} = \{\boldsymbol{k}_1, \boldsymbol{k}_2, ..., \boldsymbol{k}_L\} \in \mathbb{R}^{L \times d_\text{k}}$, and value vectors $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_L\} \in \mathbb{R}^{L \times d_\text{v}}$ through linear projection. The core idea of the attention mechanism in LLMs lies in: each query token $\boldsymbol{q}_l \in \mathbb{R}^{d_\text{k}}$ (corresponding to the l-th position in the input sequence) computes semantic relevance scores with all key tokens from the preceding context $\boldsymbol{k}_{:l} = \{\boldsymbol{k}_1, \boldsymbol{k}_2, ..., \boldsymbol{k}_l \in \mathbb{R}^{d_\text{k}}\}$, and then performs a weighted sum of the corresponding value tokens $\boldsymbol{v}_{:l} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_l \in \mathbb{R}^{d_\text{v}}\}$ based on these scores, ultimately yielding an output fused with contextual information. Formally, for an input sequence of length l, the attention output $\boldsymbol{o}_l$ at the l-th position can be defined as:

$$\boldsymbol{o}_l = \sum_{i=1}^{l} \boldsymbol{\alpha}(\boldsymbol{q}_l, \boldsymbol{k}_i) \cdot \boldsymbol{v}_i, \tag{1}$$

in which

$$\boldsymbol{\alpha}(\boldsymbol{q}_l, \boldsymbol{k}_i) = \text{Softmax}(\boldsymbol{q}_l^\top \boldsymbol{k}_i / \sqrt{d_\text{k}}), \tag{2}$$

where $\boldsymbol{\alpha}(\boldsymbol{q}_l, \boldsymbol{k}_i)$ denotes the attention weight between the query vector $\boldsymbol{q}_l$ and the i-th key vector $\boldsymbol{k}_i$, which measures the degree of semantic relevance between them; $\sqrt{d_{\text{k}}}$ is the dot-product normalization factor.

In long context processing scenarios, the computational complexity of the dense attention mechanism grows quadratically with the sequence length ($O(L^2)$, where L denotes the sequence length), gradually becoming the core bottleneck limiting model inference efficiency. By only performing attention computations on a subset of contextual tokens, sparse attention provides a critical solution to break through this bottleneck.

2.2 Punctuation-aware Hybrid Sparse Attention

The pipeline of **P**unctuation-aware **H**ybrid **S**parse **A**ttention (**PHSA**) is shown in Figure 2. First, we partition the key sequence into non-overlapping blocks of equal length. To preserve essential local context, the initial block and a local window around the query are assigned the highest priority and are always retained. The remaining intermediate blocks are treated as candidates for subsequent selection. For each candidate block, **PHSA** constructs a representative token by integrating two complementary semantic views: a global semantic representation and a punctuation-aware semantic representation. The global representation is obtained via average pooling over all tokens within the block, while the punctuation-aware representation is computed by pooling over punctuation tokens only. These two representations are then combined through a weighted aggregation to form a unified representative token for the block. Next, the relevance between the query and each representative token is evaluated, and a ranking-based strategy is employed to select the Top-K most relevant blocks. Attention is finally computed between the query and the keys and values within the selected blocks, producing the final attention output. In the following sections, we provide detailed descriptions of each component of **PHSA** and its implementation.

2.2.1 Block Segmentation

To optimize memory access efficiency, we adopt a block-level key-value caching strategy: Specifically, the key sequence $\boldsymbol{K}$ is partitioned into non-overlapping blocks of equal size, each containing m consecutive tokens. This results in a set of key blocks $\mathcal{B} = \{\boldsymbol{B}_0, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_{T-1}\}$, where $T = \lfloor L/m \rfloor$ and the t-th key block $\boldsymbol{B}_t$ is defined as:

$$\boldsymbol{B}_t = \boldsymbol{K}_{t \cdot m:(t+1) \cdot m}. \tag{3}$$

The corresponding value tokens $\boldsymbol{V}$ are accessed only after block selection and are therefore omitted here for clarity.

2.2.2 Punctuation Token Selection

Punctuation marks (e.g., commas, periods, and semicolons) serve as semantic boundary markers in natural language, effectively dividing sentence- or phrase-level semantic units. To leverage this property, we first automatically extract punctuation-related tokens from the tokenizer vocabulary to construct a punctuation token set $\mathcal{P}$. Given the tokenizer vocabulary $\mathcal{V}$, where each element corresponds to a token ID, the punctuation token set $\mathcal{P}$ is formally defined as:

$$\mathcal{P} = \{p \in \mathcal{V} \mid \boldsymbol{P}(\boldsymbol{Z}(p)) = 1\}, \tag{4}$$

where $\boldsymbol{Z}(p)$ denotes the set of character strings decoded from the token ID p. $\mathcal{P} \subseteq \mathcal{V}$ ensures that only punctuation tokens existing in the vocabulary are included. $\boldsymbol{P}(\cdot)$ is a punctuation judgment function used to filter out pure punctuation tokens:

$$\boldsymbol{P}(\boldsymbol{Z}(p)) = \begin{cases} 1, & \boldsymbol{Z}(p) \in \Gamma, \\ 0, & \boldsymbol{Z}(p) \notin \Gamma, \end{cases} \tag{5}$$

where Γ denotes the set of character strings for all punctuation marks. $\mathcal{P}$ provides fine-grained semantic boundary cues for block-level representation learning and relevance evaluation in the subsequent block selection stage. To illustrate this process, we provide an example to illustrate these operations. Given a tokenized sentence: "*To(1249) be(387) or(476) not(537) to(311) be(387) ,(11) that(429) is(374) the(279) question(3405) .(13)*". In this case, tokens 11 and 13 are identified as punctuation tokens in the punctuation token set $\mathcal{P}$.

2.2.3 Representative Tokens Calculation

To enable accurate and efficient block selection, we construct block-level semantic representations and compute their relevance to the query. The detailed formulation of each component is described as follows.

For each key block $\boldsymbol{B}_t$, we design two mean-pooling-based semantic representations to characterize its internal semantics from complementary perspectives: a global semantic representation and a punctuation semantic representation.

(1) Global Semantic Representation ($\boldsymbol{M}_0$). The global representation captures the overall semantic content of the block and is defined as:

$$\boldsymbol{M}_0(\boldsymbol{B}_t) = \frac{1}{m} \sum_{i=t\cdot m}^{(t+1)\cdot m-1} \boldsymbol{K}_i. \tag{6}$$

(2) Punctuation Semantic Representation ($\boldsymbol{M}_{\text{p}}$). The punctuation-enhanced representation emphasizes semantic boundary information associated with punctuation tokens within the block and is defined as:

$$\boldsymbol{M}_{\text{p}}(\boldsymbol{B}_t) = \frac{\sum_{i=t\cdot m}^{(t+1)\cdot m-1} \boldsymbol{I}(\boldsymbol{x}_i \in \mathcal{P}) \cdot \boldsymbol{K}_i}{\sum_{i=t\cdot m}^{(t+1)\cdot m-1} \boldsymbol{I}(\boldsymbol{x}_i \in \mathcal{P})}, \tag{7}$$

where $\boldsymbol{I}(\cdot)$ is an indicator function (returns 1 if the condition is satisfied, otherwise 0); $\boldsymbol{x}_i$ is the token ID of the i-th token in the input sequence, and $\boldsymbol{I}(\boldsymbol{x}_i \in \mathcal{P})$ indicates whether the token is a punctuation token; $\boldsymbol{K}_i$ is the i-th row of the key vector matrix $\boldsymbol{K}$. If the block does not contain any punctuation tokens, then $\boldsymbol{M}_{\text{p}}(\boldsymbol{B}_t) = \boldsymbol{M}_0(\boldsymbol{B}_t)$.

The relevance score between the query token $\boldsymbol{q}_l$ and the key block $\boldsymbol{B}_t$ fuses the global semantic representation and the punctuation-enhanced representation through a gating mechanism, Eq. (2) can be rewritten in the following form:

$$\begin{aligned} \boldsymbol{\alpha}(\boldsymbol{q}_l, \boldsymbol{M}(\boldsymbol{B}_t)) &= \text{Softmax}(\frac{\boldsymbol{q}_l^\top(\boldsymbol{M}(\boldsymbol{B}_t))}{\sqrt{d_{\text{k}}}}), \\ \boldsymbol{M}(\boldsymbol{B}_t) &= \lambda \boldsymbol{M}_0(\boldsymbol{B}_t) + (1-\lambda)\boldsymbol{M}_{\text{p}}(\boldsymbol{B}_t), \end{aligned} \tag{8}$$

where $\boldsymbol{M}(\boldsymbol{B}_t)$ is the representative key of $\boldsymbol{B}_t$, and $\lambda \in [0, 1]$ is a gating parameter that balances the contributions of the two representations.

2.2.4 Block Selection

Based on the block-level relevance score $\boldsymbol{\alpha}(\boldsymbol{q}_l, \boldsymbol{M}(\boldsymbol{B}_t))$, we select the Top-K key blocks with the highest scores. To ensure the preservation of essential contextual information, we introduce a priority mechanism in which the initial block and local window blocks are assigned the highest priority and are always retained in the attention set. This design guarantees that high-contribution blocks are consistently included in attention computation, mitigating the potential loss of critical information caused by sparse selection.

2.2.5 Attention Calculation

The keys and values corresponding to the finally selected blocks are respectively combined into lightweight to-be-computed vectors $\hat{\boldsymbol{K}}$ and $\hat{\boldsymbol{V}}$ via concatenation. Meanwhile, the dense attention output as shown in Eq. (1) and Eq. (2) can be expressed in the following sparse attention output $\hat{\boldsymbol{o}}_l$ form:

$$\hat{\boldsymbol{o}}_l = \sum_{i=1}^{l} \boldsymbol{\alpha}(\boldsymbol{q}_l, \hat{\boldsymbol{k}}_i) \cdot \hat{\boldsymbol{v}}_i, \tag{9}$$

in which

$$\boldsymbol{\alpha}(\boldsymbol{q}_l, \hat{\boldsymbol{k}}_i) = \text{Softmax}(\boldsymbol{q}_l^\top \hat{\boldsymbol{k}}_i / \sqrt{d_{\text{k}}}), \tag{10}$$

where $\hat{\boldsymbol{k}}_i$ and $\hat{\boldsymbol{v}}_i$ are selected for sparse attention calculation, and they belong to $\hat{\boldsymbol{K}}$ and $\hat{\boldsymbol{V}}$.

Table 1: Comparison of NIAH scores across different combinations of training or inference methods (Dense, PHSA, and InfLLM v2) for a text length of 4k tokens. We focus on demonstrating the cross-matching effects between PHSA's training and generation under different Top-K values. And N/A denotes direct adoption of the Qwen3-0.6B-Base model.

Training Top-K	Training Method	Inference Method	Inference Top-K 1	2	4	8	16
N/A		PHSA	**83.5**	**96.4**	**98.4**	**99.8**	99.8
		InfLLM v2 Zhao et al. [2025]	80.6	94.0	97.0	99.4	**100.0**
0	PHSA		0.0	0.0	0.0	0.0	0.0
1			79.6	91.4	97.6	99	99.4
2			**81.8**	**95.0**	**98.8**	**99.8**	99.8
4			77.8	87.2	94.2	98.8	99.4
8			74.6	87.4	94.8	98.6	99.4
16			77.8	89.6	95.2	99.4	**100.0**
2	InfLLM v2		78.6	89.6	97.6	99.6	100.0
Dense		PHSA	**79.8**	86.8	**93.6**	**99.2**	**98.8**
		InfLLM v2	76.4	**88.0**	93.4	98.4	98.4

3 Experiments

In this section, we evaluate the proposed PHSA method and perform a comparative analysis with dense attention and InfLLM v2. Our experiments cover both training-based and training-free paradigms, with comprehensive evaluations conducted on general benchmarks and long-context benchmarks. We aim to address the following research questions (**RQ**):

RQ1: Why prefer lower Top-K?

RQ2: Is PHSA effective on general benchmarks?

RQ3: Is PHSA effective on long-context benchmarks?

RQ4: What is the impact of punctuation marks across different groups?

3.1 Experiment Settings

In this section, we present the complete experimental setup. All training runs were performed on 64 NPUs, each with 64 GB of NPU memory.

3.1.1 Datasets

The training dataset includes open-source datasets: dclm Li et al. [2024b], map-cc Du et al. [2024], ultrachat Ding et al. [2023], tuluv3 Lambert et al. [2024], finemath Liu et al. [2024], megamath Zhou et al. [2025] and high-quality proprietary self-collected datasets related to education. Punctuation tokens account for approximately 10.16% of the training data, while approximately 6.57% of blocks contain no punctuation tokens.

3.1.2 Model Configuration

We include both training and training-free experiments in the experimental section. Training is conducted on sequences with lengths of 4k and 32k tokens, which verifies the effectiveness of PHSA on general benchmarks and long-context benchmarks, respectively. For the 4k sequence length, our initialization block and local window block cover 16 and 128 tokens each. For the 32k sequence length, the corresponding values are 128 and 512 tokens. Each equally partitioned block consists of 16 tokens. Unless otherwise specified, our training is initialized from the Qwen3-0.6B-Base model, and the untrained sparse inference method is consistent with InfLLM v2. For the training-free experiments, we validate the performance of PHSA on three model scales: Qwen3-0.6B, Qwen3-8B Yang et al. [2025a] and MiniCPM4-0.5B Team et al. [2025]. All sparse methods use the same initialization block and local window settings.

3.1.3 Evaluation Metrics

To comprehensively verify the effectiveness of PHSA in general performance, long-context processing capability and information loss control, experiments were implemented on general benchmarks,

Table 2: Scores of Qwen3-0.6B-Base (initial training point) and differently trained models (100B training tokens) on general benchmarks. An inference Top-K value of N/A indicates that dense attention is adopted during inference. Boldface and boxed values denote the best and second-best values.

Training Tokens	100B									
Method	Dense	Dense	PHSA	PHSA	InfLLM v2	InfLLM v2	PHSA	PHSA	InfLLM v2	InfLLM v2
Training Top-K	Dense	Dense	2	2	2	2	4	4	4	4
Inference Top-K	N/A	2	N/A	2	N/A	2	N/A	4	N/A	4
GSM8K	52.16	54.81	56.63	56.50	52.16	52.84	56.25	55.80	**57.32**	56.50
MathQA	41.91	42.24	**42.91**	42.58	41.81	42.31	42.48	42.38	42.24	41.98
MATH	26.48	23.80	23.78	22.50	26.08	22.90	**26.62**	26.48	23.28	23.88
ARC-C	41.30	41.13	41.13	41.04	41.97	**42.15**	41.97	41.30	41.72	41.55
ARC-C-ZH	34.30	34.22	33.02	32.68	**35.67**	35.41	34.39	34.04	34.98	35.07
ARC-E	71.84	71.97	71.13	71.21	72.52	72.31	72.64	**72.69**	72.10	72.18
C-Valid	58.77	58.99	**59.96**	58.99	59.36	58.40	59.14	58.84	58.77	58.25
CMMLU	55.38	55.44	55.16	55.16	**55.90**	55.53	55.23	54.96	55.48	55.51
HellaSwag	**40.78**	40.73	40.71	40.68	40.39	40.48	40.54	40.56	40.61	40.61
HumanEval	36.59	33.54	36.59	33.54	34.15	31.71	**42.07**	37.20	36.59	35.37
LAMBADA	49.52	49.62	**50.13**	50.09	49.84	49.76	49.85	49.74	49.91	49.91
MMLU	**55.53**	55.34	55.26	55.00	54.98	54.79	55.10	55.11	55.33	54.97
PIQA	70.95	70.84	70.62	70.62	71.06	70.89	**71.27**	71.22	69.86	69.86
XStoryCloze	58.90	58.90	58.90	58.97	59.50	59.56	59.36	**59.76**	57.91	58.24
BBH	40.02	33.99	39.56	33.37	39.78	26.26	32.11	32.78	**40.55**	40.02
Avg.	48.96	48.37	49.03	48.20	49.01	47.69	**49.27**	48.86	49.11	48.93

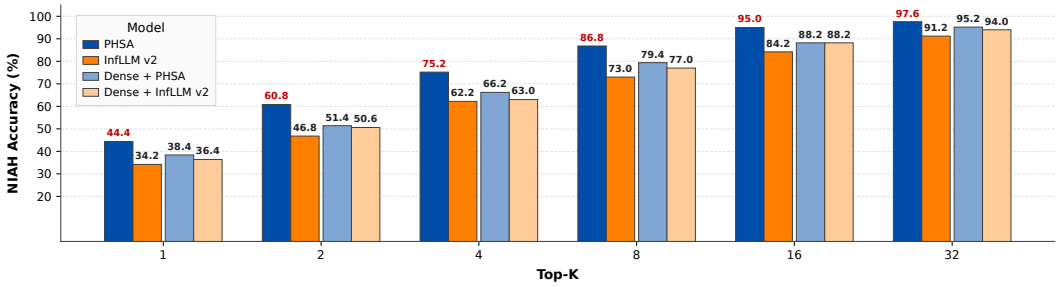


Figure 3: Comparison of NIAH scores for PHSA and InfLLM v2 (training vs. training-free) at 32k sequence length.

LongBench Bai et al. [2024] and Needle-in-a-Haystack (NIAH) Hsieh et al. [2024] respectively. The general benchmarks cover 15 typical task categories, including mathematical reasoning (GSM8K Cobbe et al. [2021], MathQA Amini et al. [2019], MATH Hendrycks et al. [2021]), commonsense question answering (ARC-C, ARC-C-ZH (Chinese translated version of ARC-C), ARC-E Clark et al. [2018]), comprehensive cognitive reasoning (MMLU Hendrycks et al. [2020], CMMLU Li et al. [2024a], BBH Suzgun et al. [2023]), text coherence assessment (LAMBADA Paperno et al. [2016], XStoryCloze Lin et al. [2022]), code generation (HumanEval Chen [2021]), and domain-specific tasks (C-Valid Huang et al. [2023], HellaSwag Zellers et al. [2019], PIQA Bisk et al. [2020]), which span both English and Chinese bilingual scenarios. LongBench consists of four task modules, namely single-turn question answering (Single, including MQE, MQZ, NQA, QAS), few-shot learning (Fewshot, including LST, SSM, TRC, TQA), text summarization (Summarization, including GRP, MTN, QSM, VSM), and code-related tasks (Code, including LCC, RBP). NIAH primarily assesses the performance of sparse attention mechanisms in the preservation and retrieval of long-distance information, and it can most intuitively reflect the boundary of information loss under sparse attention.

Table 3: Comparison of NIAH scores of PHSA and InfLLM v2 inference methods for Qwen3-8B and MiniCPM4-0.5B.

Seq_len	Method	Thinking	Top-K (Qwen3-8B) 1	2	4	Top-K (MiniCPM4-0.5B) 8
16k	**InfLLM v2 Zhao et al. [2025]**	w/o	62.0	87.0	97.8	93.4
	PHSA		**69.0**	**89.2**	**98.6**	**95.2**
16k	**InfLLM v2**	w/	59.4	74.2	90.4	N/A
	PHSA		**64.2**	**81.0**	**94.8**	
32k	**InfLLM v2**	w/o	64.6	85.8	98.0	71.0
	PHSA		**68.0**	**88.8**	**99.0**	**73.5**

Table 4: LongBench scores for models trained with Dense, InfLLM v2, PHSA, PHSA_less and PHSA_en+zh.

Method	Single				Fewshot				Summarization				Code		Avg.
	MQE	MQZ	NQA	QAS	LST	SSM	TRC	TQA	GRP	MTN	QSM	VSM	LCC	RBP	
Dense	34.53	22.13	7.16	20.47	25.67	**28.12**	**58.5**	74.90	13.11	**15.37**	15.33	12.23	30.47	32.05	27.86
InfLLM v2	31.02	21.85	10.18	20.98	28.00	26.36	52.0	77.71	13.19	15.30	15.54	12.65	26.98	31.80	27.40
PHSA	33.95	20.63	8.11	**22.46**	26.75	26.70	55.5	76.94	13.17	15.06	**15.55**	12.75	**31.08**	**35.14**	28.13
PHSA_less	30.98	20.94	7.95	19.04	28.5	27.07	49	73.49	13.09	14.83	15.56	12.45	29.25	32.48	26.76
PHSA_en+zh	**35.96**	**22.22**	**11.04**	20.67	**30.50**	26.02	52.5	**79.85**	**13.45**	15.07	15.54	**14.17**	25.95	31.20	**28.15**

3.2 Effect of PHSA

3.2.1 Low Top-K Preference (RQ1)

We use a fixed global value of $1 - \lambda = 0.1$ (i.e., $\lambda = 0.9$) in all experiments. The sensitivity results across different $1 - \lambda$ values and Top-K settings are reported in Appendix E.

A large number of attention computations are performed on irrelevant tokens during model training. Sparse attention enables the model to filter out the most relevant tokens in the training process, thereby accelerating training and facilitating faster convergence simultaneously. However, an excessively low Top-K value often prevents the model from capturing all relevant tokens, which consequently leads to model performance collapse.

As shown in Table 1, in the scenario with training-inference consistency, training with a relatively low training Top-K can reduce information loss at low inference Top-K, while an excessively small training Top-K degrades model performance. Specifically, training Top-K=2 serves as an effective configuration, yielding an NIAH score of 95 at inference Top-K=2, compared with 91.4 when training Top-K=1. Performance drops when training Top-K≥ 4. Under the same training Top-K setting, PHSA achieves better performance; for instance, with training Top-K=2, PHSA obtains a score of 81.8 at inference Top-K=1, outperforming InfLLM v2 at 78.6. Thus, training Top-K=2 or 4 is a locally effective sparsity level for short-text training. PHSA also achieves higher accuracy than InfLLM v2 when applied to dense-trained models and directly to Qwen3-0.6B-Base.

3.2.2 Effect on general benchmarks (RQ2)

General benchmarks serve as an effective means to evaluate the fundamental capabilities of models. We present the dense inference scores and scores under training-inference consistent settings of PHSA, InfLLM v2 and dense attention with 20B (Appendix D) and 100B training tokens.

As shown in Table 2, when trained with 100B tokens, PHSA maintains competitive performance against InfLLM v2 under Top-K=2 and Top-K=4, and achieves an average score of 49.27 at Top-K=4, exceeding the dense attention score of 48.96. Compared with the shorter 20B training regime, a moderately higher Top-K can reduce model error and improve performance in the longer training regime.

Experiments in RQ2 indicate that PHSA has the potential to enhance model general capabilities under low Top-K settings compared with dense attention.

Table 5: NIAH scores of punctuation-control variants on Qwen3-0.6B at 16K context length. Random Prompts replaces punctuation marks in the prompts with other punctuation tokens. Random Tokens and High-Frequency replace the punctuation branch with random tokens and high-frequency words, respectively.

Method	**Top-K=2**	**Top-K=4**
InfLLM v2	65.4	81.0
PHSA	73.6	84.2
InfLLM v2 (Random Prompts)	68.4	76.0
PHSA (Random Prompts)	73.4	81.2
PHSA (Random Tokens)	68.0	80.0
PHSA (High-Frequency)	64.6	79.8

Table 6: Measured computational cost of PHSA and InfLLM v2 at different sequence lengths.

Length	**Method**	**Time (ms)**	**Peak Memory (MiB)**	**Throughput (M tok/s)**
4K	InfLLM v2	0.660134	56.501	6.21
4K	PHSA	0.671898	56.501	6.10
16K	InfLLM v2	6.124999	706.001	2.67
16K	PHSA	6.125182	706.001	2.66
32K	InfLLM v2	22.470382	2692.001	1.46
32K	PHSA	22.474310	2692.001	1.46

3.2.3 Effect on long-context benchmarks (RQ3)

Long-context benchmarks are important metrics for evaluating the performance of sparse attention methods. Sparse attention yields more significant efficiency gains in long-context scenarios, while sparser token selection is accompanied by a higher probability of information loss.

Table 3 shows that for the 8B-parameter model, PHSA consistently outperforms InfLLM v2 on the NIAH dataset across both 16k and 32k sequence lengths, regardless of whether the Thinking step is performed before inference. This result also validates the effectiveness of the training-free paradigm in long-context scenarios. To verify the effectiveness of PHSA across different tokenizers, we further conduct experiments on MiniCPM4-0.5B Team et al. [2025]; PHSA also exhibits better performance in the corresponding settings.

We also conduct extensive training with Top-K set to 16. As evidenced by the NIAH metrics illustrated in Figure 3, training can further amplify the gains of PHSA. Specifically, PHSA reduces the loss of critical information by approximately 10% compared with InfLLM v2.

For LongBench, Table 4 reports the performance of dense attention, InfLLM v2 and PHSA. PHSA achieves competitive or superior performance over the two baselines on most subtasks, with an overall average score of 28.13, compared with 27.86 for dense attention and 27.40 for InfLLM v2. These results show that PHSA improves the overall long-context modeling performance on LongBench.

To examine whether the gains depend on punctuation-specific signals, we compare PHSA with variants that replace punctuation in NIAH prompts or replace the punctuation branch with other token subsets. Table 5 reports the results on Qwen3-0.6B at 16K context length. PHSA retains its advantage when punctuation marks in the prompts are replaced. Replacing the punctuation branch with random tokens or high-frequency words does not reproduce the same gains, suggesting that the observed improvements are not explained by an arbitrary second token subset.

Table 6 compares the measured time, peak memory, and throughput of PHSA and InfLLM v2 at different sequence lengths. The measured time and peak memory of PHSA are close to those of InfLLM v2 across the evaluated sequence lengths.

3.3 Additional Discussion

3.3.1 Multi Punctuation Impact (RQ4)

We note that PHSA delivers suboptimal performance on two Chinese-related metrics, namely ARC-C-ZH in Table 2 and MQZ in Table 4. We attribute this deficiency to the exclusive use of English punctuation marks during the training phase. To verify this conjecture and further demonstrate the effectiveness of PHSA across different languages, we incorporate Chinese punctuation marks for training and obtain the PHSA_en+zh model presented in Table 4. The MQZ metric improves from 20.63 to 22.22, indicating that the additional punctuation marks contribute to performance on the corresponding language task.

Punctuation tokens do not serve identical functions in practice. We categorize them into commonly used sentence delimiters (e.g., enumeration commas, commas, periods, and semicolons) and general-purpose symbols that appear in specific contexts such as code snippets and tabular data (e.g., parentheses, square brackets, curly braces, and percent signs). As shown in Table 4, PHSA_less denotes the LongBench scores obtained by training and inference using only sentence delimiters. The average score drops from 28.13 to 26.76 when general-purpose symbols are excluded. In the Single category, the score on multifieldqa_en decreases from 33.95 to 30.98. The Code metrics, which involve many general-purpose symbols, also show a decline.

Overall, these results support incorporating punctuation marks from different languages and functional categories into PHSA's dual-branch design.

4 Conclusion

In this paper, we propose a Punctuation-aware Hybrid Sparse Attention (PHSA) mechanism, which takes punctuation tokens as natural semantic boundary anchors and designs a dual-branch aggregation mechanism integrating global semantic representations and punctuation-enhanced boundary features, effectively preserving core semantic information without additional computational overhead. Extensive experimental results validate that the proposed PHSA outperforms dense attention and the state-of-the-art baseline InfLLM v2 across both general benchmarks and long-context evaluation tasks under both training and training-free paradigms. It can be seen that PHSA effectively raises the sparsity upper bound of sparse attention in large language models. In addition, we conducted extensive ablation studies on the selection of sparsity levels, different tokenizers, varying model scales, and punctuation token categories, which demonstrate both the generalizability of PHSA's effectiveness and its performance boundaries. PHSA not only provides a new effective solution for improving the efficiency and performance of long-context processing in large language models but also lays a theoretical and experimental foundation for the lightweight deployment of large language models on resource-constrained devices.

Although PHSA can effectively reduce the frequency of information loss in sparse attention, we argue that the wholesale introduction of punctuation marks would introduce noise. We believe that a screened subset of punctuation marks has the potential to raise the upper bound of PHSA performance; however, due to the constraints of paper length and computational resources, we will conduct an in-depth investigation into this direction in future work.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No. 62576330), National Natural Science Foundation of Anhui (No.2508085MF143).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based

formalisms. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2357–2367, 2019.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 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)*, pages 3119–3137, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439, 2020.

Guoxuan Chen, Han Shi, Jiawei Li, Yihang Gao, Xiaozhe Ren, Yimeng Chen, Xin Jiang, Zhenguo Li, Weiyang Liu, and Chao Huang. Sepllm: Accelerate large language models by compressing one segment into one separator. *arXiv preprint arXiv:2412.12094*, 2024a.

Mark Chen. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, et al. Magicpig: Lsh sampling for efficient llm generation. *arXiv preprint arXiv:2410.16179*, 2024b.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Aditya Desai, Shuo Yang, Alejandro Cuadron, Matei Zaharia, Joseph E Gonzalez, and Ion Stoica. Hashattention: Semantic sparsity for faster inference. *arXiv preprint arXiv:2412.14468*, 2024.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, 2023.

Xinrun Du, Zhouliang Yu, Songyang Gao, Ding Pan, Yuyang Cheng, Ziyang Ma, Ruibin Yuan, Xingwei Qu, Jiaheng Liu, Tianyu Zheng, et al. Chinese tiny llm: Pretraining a chinese-centric large language model. *arXiv preprint arXiv:2404.04167*, 2024.

Jiaming Fan, CAO DAMING, Xiangzhong Luo, Jiale Fu, Chonghan Liu, et al. Flatter tokens are more valuable for speculative draft model training. In *International Conference on Learning Representations*, volume 2026, pages 94224–94248, 2026.

Zixun Fang, Zhiheng Liu, Kai Zhu, Yu Liu, Ka Leong Cheng, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Vangogh: A unified multimodal diffusion-based framework for video colorization. *arXiv preprint arXiv:2501.09499*, 2025.

Zixun Fang, Kai Zhu, Zhiheng Liu, Yu Liu, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Panoramic video generation with pretrained diffusion models. *Advances in Neural Information Processing Systems*, 38:12486–12508, 2026.

Jiale Fu, Yuchu Jiang, Junkai Chen, Jiaming Fan, Xin Geng, and Xu Yang. Fast large language model collaborative decoding via speculation. *arXiv preprint arXiv:2502.01662*, 2025.

Yizhao Gao, Jianyu Wei, Qihao Zhang, Yu Cheng, Shimao Chen, Zhengju Tang, Zihan Jiang, Yifan Song, Hailin Zhang, Liang Zhao, et al. Hysparse: A hybrid sparse attention architecture with oracle token selection and kv cache sharing. *arXiv preprint arXiv:2602.03560*, 2026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36: 62991–63010, 2023.

Houcheng Jiang, Junfeng Fang, Jiaxin Wu, Tianyu Zhang, Chen Gao, Xiang Wang, Xiangnan He, and Yang Deng. Contrastive weak-to-strong generalization. *arXiv preprint arXiv:2510.07884*, 2025a.

Houcheng Jiang, Jiajun Fu, Junfeng Fang, Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. Univlr: Unifying text and vision in visual latent reasoning for multimodal llms. *arXiv preprint arXiv:2605.11856*, 2026a.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515, 2024.

Yuchu Jiang, Yue Cai, Xiangzhong Luo, Jiale Fu, Jiarui Wang, Chonghan Liu, et al. dΘ2 cache: Accelerating diffusion-based llms via dual adaptive caching. In *International Conference on Learning Representations*, volume 2026, pages 137511–137531, 2026b.

Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25004–25014. IEEE, 2025b.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Changcheng Li, Jiancan Wu, Hengheng Zhang, Zhengsu Chen, Guo An, Junxiang Qiu, Xiang Wang, and Qi Tian. Confidence before answering: A paradigm shift for efficient llm uncertainty estimation. *arXiv preprint arXiv:2603.05881*, 2026a.

Chenxu Li, Zhicai Wang, Yuan Sheng, Xingyu Zhu, Yanbin Hao, and Xiang Wang. Res-bench: Benchmarking the robustness of multimodal large language models to dynamic resolution input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31545–31553, 2026b.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, 2024a.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024b.

Ouxiang Li, Yuan Wang, Xinting Hu, Huijuan Huang, Rui Chen, Jiarong Ou, Xin Tao, Pengfei Wan, Xiaojuan Qi, and Fuli Feng. Easier painting than thinking: Can text-to-image models set the stage, but not direct the play? In *International Conference on Learning Representations*, volume 2026, pages 86729–86758, 2026c.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024c.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 9019–9052, 2022.

Guangda Liu, Chengwei Li, Jieru Zhao, Chenqi Zhang, and Minyi Guo. Clusterkv: Manipulating llm kv cache in semantic space for recallable compression. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pages 1–7. IEEE, 2025.

Yan Liu, Renren Jin, Ling Shi, Zheng Yao, and Deyi Xiong. Finemath: A fine-grained mathematical evaluation benchmark for chinese large language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 2024.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1412–1421, 2015.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1525–1534, 2016.

Junxiang Qiu, Lin Liu, Shuo Wang, Jinda Lu, Kezhou Chen, and Yanbin Hao. Accelerating diffusion transformer via gradient-optimized cache. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17608–17617. IEEE, 2025a.

Junxiang Qiu, Jinda Lu, and Shuo Wang. Multimodal generation with consistency transferring. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 504–513, 2025b.

Junxiang Qiu, Shuo Wang, Jinda Lu, Lin Liu, Houcheng Jiang, Xingyu Zhu, and Yanbin Hao. Accelerating diffusion transformer via error-optimized cache. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 9588–9597, 2025c.

Anton Razzhigaev, Matvey Mikhalchuk, Temurbek Rahmatullaev, Elizaveta Goncharova, Polina Druzhinina, Ivan Oseledets, and Andrey Kuznetsov. Llm-microscope: Uncovering the hidden role of punctuation in context memory of transformers. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7757–7764, 2025.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, 2023.

MiniCPM Team, Chaojun Xiao, Yuxuan Li, Xu Han, Yuzhuo Bai, Jie Cai, Haotian Chen, Wentong Chen, Xin Cong, Ganqu Cui, et al. Minicpm4: Ultra-efficient llms on end devices. *arXiv preprint arXiv:2506.07900*, 2025.

MiniCPM Team, Wenhao An, Yingfa Chen, Yewei Fang, Jiayi Li, Xin Li, Yaohui Li, Yishan Li, Yuxuan Li, Biyuan Lin, et al. Minicpm-sala: Hybridizing sparse and linear attention for efficient long-context modeling. *arXiv preprint arXiv:2602.11761*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yuan Wang, Ouxiang Li, Tingting Mu, Yanbin Hao, Kuien Liu, Xiang Wang, and Xiangnan He. Precise, fast, and low-cost concept erasure in value space: Orthogonal complement matters. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28759–28768. IEEE, 2025.

Yuan Wang, Ouxiang Li, Yulong Xu, Borui Liao, Jiajun Liang, Jinghan Li, Meng Wang, Xintao Wang, Pengfei Wan, Kuien Liu, et al. Think, then score: Decoupled reasoning and scoring for video reward modeling. *arXiv preprint arXiv:2605.05922*, 2026a.

Yuan Wang, Borui Liao, Huijuan Huang, Jinda Lu, Ouxiang Li, Kuien Liu, Meng Wang, and Xiang Wang. Thinking with frames: Generative video distortion evaluation via frame reward model. *arXiv preprint arXiv:2601.04033*, 2026b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yiyan Xu, Wenjie Wang, Yang Zhang, Biao Tang, Peng Yan, Fuli Feng, and Xiangnan He. Personalized image generation with large multimodal models. In *Proceedings of the ACM on Web Conference 2025*, pages 264–274, 2025a.

Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. Personalized generation in large model era: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24607–24649, 2025b.

Yiyan Xu, Ruoxuan Xia, Wuqiang Zheng, Fengbin Zhu, Wenjie Wang, and Fuli Feng. Nextads: Towards next-generation personalized video advertising. *arXiv preprint arXiv:2603.02137*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Lijie Yang, Zhihao Zhang, Arti Jain, Shijie Cao, Baihong Yuan, Yiwei Chen, Zhihao Jia, and Ravi Netravali. Less is more: Training-free sparse attention with global locality for efficient reasoning. *arXiv preprint arXiv:2508.07101*, 2025b.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23078–23097, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. *arXiv e-prints*, pages arXiv–2502, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

Weilin Zhao, Zihan Zhou, Zhou Su, Chaojun Xiao, Yuxuan Li, Yanghao Li, Yudi Zhang, Weilun Zhao, Zhen Li, Yuxiang Huang, et al. Infllm-v2: Dense-sparse switchable attention for seamless short-to-long adaptation. *arXiv preprint arXiv:2509.24663*, 2025.

Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P Xing. Megamath: Pushing the limits of open math corpora. *arXiv preprint arXiv:2504.02807*, 2025.

Xingyu Zhu, Junfeng Fang, Shuo Wang, Beier Zhu, Zhicai Wang, Yonghui Yang, and Xiangnan He. Mitigating hallucinations in large vision-language models without performance degradation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1995–2009, 2026a.

Xingyu Zhu, Kesen Zhao, Liang Yi, Shuo Wang, Zhicai Wang, Beier Zhu, Hanwang Zhang, and Xiangnan He. Look carefully: Adaptive visual reinforcements in multimodal large language models for hallucination mitigation. In *International Conference on Learning Representations*, volume 2026, pages 81235–81256, 2026b.

Xingyu Zhu, Beier Zhu, Junfeng Fang, Shuo Wang, Yin Zhang, Xiang Wang, and Xiangnan He. Guardalign: Test-time safety alignment in multimodal large language models. *arXiv preprint arXiv:2602.24027*, 2026c.

Appendix

A Related Work

In this section, we introduce current related sparse attention methods in terms of whether training is required.

A.1 Training-free Sparse Attention

Training-free sparse attention methods aim to reduce inference latency by introducing sparsity into the attention mechanism without modifying the pre-trained dense attention backbone. These methods have made remarkable attempts in optimizing Key-Value (KV) cache usage and reducing computational overhead, but they still face significant challenges in translating theoretical advantages into practical efficiency improvements. One major challenge is phase-limited sparsity. Methods such as H2O Zhang et al. [2023] apply sparsity during autoregressive decoding but require dense preprocessing during prefill. In contrast, methods like MInference Jiang et al. [2024] focus on prefill sparsity. Neither of these approaches achieves acceleration across all inference phases, as the computational cost of at least one phase remains comparable to that of dense attention. This phase specialization impairs their acceleration capability in workloads dominated by prefill (*e.g.*, book summarization and code completion) or decoding (*e.g.*, long chain-of-thought Wei et al. [2022] reasoning). Recent works including LessIsMore Yang et al. [2025b] and SpargeAttn Zhang et al. [2025] have carried out targeted optimization for this limitation, trying to break the single-phase barrier and realize unified acceleration of prefill and decoding stages under the training-free paradigm. But their acceleration gains still suffer from severe degradation in ultra-long context and heterogeneous workload scenarios, and the acceleration bottleneck and model performance degradation induced by train-inference mismatch remain an inevitable problem.

A.2 Trainable Sparse Attention

Trainable sparse attention methods enable models to learn optimal sparse patterns during training, addressing performance degradation issues of post-hoc sparsity. But research on training-inference consistent sparse attention requires substantial computing resources for end-to-end training and long-context validation, which leads to a slow pace of technological iteration. Representative works include ClusterKV Liu et al. [2025] with k-means clustering for token grouping to reduce attention overhead, and MagicPIG Chen et al. [2024b] using SimHash-based selection to filter out uninformative tokens, though both contain untrainable discrete operations in their core sparse pipelines. The non-differentiable nature of their hard clustering and hash matching steps means these core sparse pattern generation processes cannot be optimized via end-to-end gradient backpropagation, limiting the model's ability to learn task-optimal sparse structures. HashAttention Desai et al. [2024] adopts finer-grained token-granularity selection to enhance trainability, but suffers from inefficient backpropagation due to its complex sparse mapping logic, with poor parallelism on mainstream hardware that hinders its scalability to ultra-long sequences. Recent advances focus on overcoming these limitations to build fully differentiable, efficient end-to-end trainable sparse attention paradigms. NSA Yuan et al. [2025] designs dynamic hierarchical sparse strategies for end-to-end training, enabling adaptive sparse pattern learning across different semantic levels. InfLLM v2 Zhao et al. [2025], integrated into MiniCPM4 Team et al. [2025], proposes a more streamlined fine-grained trainable sparse attention paradigm, which retains the key operations of NSA while simplifying redundant modules to achieve superior long-context performance. HySparse Gao et al. [2026] introduces oracle token selection for end-to-end trainable sparse attention without extra proxy modules, reducing additional training overhead while avoiding objective mismatch between proxy tasks and downstream tasks. SALA Team et al. [2026] unifies sparse and linear attention to support million-level context modeling on edge devices, expanding the application boundaries of trainable sparse attention. These works strive to balance training efficiency, inference speed, and model performance, though the inherent trade-off between these three core objectives remains a widely recognized open problem in the research community.

B Algorithm of PHSA

The pseudocode for the attention component of PHSA during training is presented below.

Algorithm 1: Punctuation-aware Hybrid Sparse Attention (PHSA)

Input : Query $\boldsymbol{Q}$, Key $\boldsymbol{K} \in \mathbb{R}^{L \times d_{\text{k}}}$, Value $\boldsymbol{V} \in \mathbb{R}^{L \times d_{\text{v}}}$, Input Sequence $\boldsymbol{X}$, Punctuation Set $\mathcal{P}$, Block size m, Top-K count k, Window size w (Local window), Init count n_{init} (Init block), Gate param λ

Output : Sparse Attention Output $\hat{\boldsymbol{O}}$

// 1. Block Segmentation

Divide $\boldsymbol{K}$ into blocks $\mathcal{B} = \{\boldsymbol{B}_0, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_{T-1}\}$, where $T = \lfloor L/m \rfloor$;

// 2. Representative Tokens Calculation

Initialize block representations $\mathcal{M} = \emptyset$;

for $t \leftarrow 0$ **to** $T-1$ **do**

  Calculate Global Rep: $\boldsymbol{M}_0(\boldsymbol{B}_t) = \frac{1}{m} \sum_{l \in \text{indices}(\boldsymbol{B}_t)} \boldsymbol{K}_l$;

  Identify punctuation: $\text{mask}_p = \{\boldsymbol{I}(\boldsymbol{x}_l \in \mathcal{P}) \mid l \in \text{indices}(\boldsymbol{B}_t)\}$;

  if $\sum \textit{mask}_p > 0$ **then**

    Calculate Punctuation Rep $\boldsymbol{M}_{\text{p}}(\boldsymbol{B}_t)$;

    $\boldsymbol{M}(\boldsymbol{B}_t) = \lambda \cdot \boldsymbol{M}_0(\boldsymbol{B}_t) + (1-\lambda) \cdot \boldsymbol{M}_{\text{p}}(\boldsymbol{B}_t)$;

  else

    $\boldsymbol{M}(\boldsymbol{B}_t) = \boldsymbol{M}_0(\boldsymbol{B}_t)$;

  end

  $\mathcal{M} \leftarrow \mathcal{M} \cup \{\boldsymbol{M}(\boldsymbol{B}_t)\}$;

end

Stack $\mathcal{M}$ to form compressed keys $\boldsymbol{K}_{\text{comp}} \in \mathbb{R}^{T \times d_{\text{k}}}$;

// 3. Relevance Scoring

Compute scores $\boldsymbol{S} \in \mathbb{R}^{L \times T}$ where $\boldsymbol{S}_{i,t} = \boldsymbol{q}_l^{\top} \boldsymbol{M}(\boldsymbol{B}_t) / \sqrt{d_{\text{k}}}$;

Initialize selected block indices $\mathcal{I}_{\text{total}} = \emptyset$;

foreach *query token* $\boldsymbol{q}_l$ *(or parallelized)* **do**

  $\mathcal{I}_{\text{init}} \leftarrow \{0, \ldots, n_{\text{init}} - 1\}$; $\mathcal{I}_{\text{local}} \leftarrow \{\max(0, \lfloor l/m \rfloor - w/m), \ldots, \lfloor l/m \rfloor\}$

  Apply causal mask to $\boldsymbol{S}_{i,:}$ (mask future blocks);

  Mask $\mathcal{I}_{\text{init}}$ and $\mathcal{I}_{\text{local}}$ positions in $\boldsymbol{S}_{i,:}$ with $-\infty$;

  $\mathcal{I}_{\text{Top-K}} \leftarrow$ indices of Top-k values in $\boldsymbol{S}_{i,:}$

  $\mathcal{I}_{\text{final}} \leftarrow \texttt{Unique}(\mathcal{I}_{\text{init}} \cup \mathcal{I}_{\text{local}} \cup \mathcal{I}_{\text{Top-K}})$;

  Store $\mathcal{I}_{\text{final}}$ for reconstruction;

end

// 4. Attention Calculation

Map block indices $\mathcal{I}_{\text{final}}$ to token indices $\mathcal{I}_{\text{tok}}$;

Gather sparse KV: $\hat{\boldsymbol{K}} \leftarrow \boldsymbol{K}[\mathcal{I}_{\text{tok}}], \hat{\boldsymbol{V}} \leftarrow \boldsymbol{V}[\mathcal{I}_{\text{tok}}]$;

Compute output $\hat{\boldsymbol{o}}_l = \text{Softmax}(\boldsymbol{q}_l \hat{\boldsymbol{K}}^{\top} / \sqrt{d_{\text{k}}}) \hat{\boldsymbol{V}}$;

return $\hat{\boldsymbol{O}} = \{\hat{\boldsymbol{o}}_1, \ldots, \hat{\boldsymbol{o}}_L\}$

C Why Punctuation as Semantic Anchors

Punctuation marks serve as natural semantic boundary anchors in natural language. They inherently divide text into logically coherent units and mark semantic turning points, providing noise-robust segmentation cues without the need for additional learning. The inherent "punctuation-semantics" correlation is already implicitly encoded in LLMs during pre-training, which can be formally interpreted via the canonical self-attention formulation Eq. (1) and Eq. (2) that underpins the Transformer architecture.

As validated through multi-layer attention visualization in SepLLMChen et al. [2024a], punctuation tokens receive significantly higher attention weights than general semantic tokens across all attention heads and layers. In addition, directly removing punctuation tokens from the training data during model training will significantly impair the model's performance. Formally, for any punctuation token

p and non-punctuation content token c in the same context, the expected attention weight satisfies $E_{i\in[1,L]}[\boldsymbol{\alpha}_{i,p}] \gg E_{i\in[1,L]}[\boldsymbol{\alpha}_{i,c}]$, which essentially indicates that punctuation tokens have condensed the core semantic information of their corresponding segments. Punctuation tokens only account for 5% to 10% of the total tokens, yet carry high-entropy structural information. From the perspective of information theory, punctuation pooling achieves the optimal trade-off between efficiency and performance, while circumventing the inherent limitations of the standard average pooling strategy. Unlike tokens of nouns, adverbs, adjectives and other lexical categories, punctuation tokens can serve as core retained semantic anchors, and are explicitly integrated into the lightweight dual-branch computation of representative tokens for sparse attention. This requires no additional design of complex token importance scoring mechanisms, nor does it disrupt the GPU communication and computation pipeline.

Therefore, to address the core challenge of semantic dilution caused by coarse-grained aggregation in sparse attention, the dual-branch design of PHSA enables block representations to capture both global semantics and fine-grained structural information with almost no additional computational overhead, thereby improving the accuracy of sparse selection.

D Extra Experiment of Effect on general benchmarks

Table 7: Scores of differently trained models (20B training tokens) on general benchmarks.

Training Tokens	**N/A**		**20B**					
Method	**Qwen-0.6B-Base**		**Dense**	**Dense**	**PHSA**	**PHSA**	**InfLLM v2**	**InfLLM v2**
Training Top-K	**N/A**		**Dense**	**Dense**	**2**	**2**	**2**	**2**
Inference Top-K	**N/A**	**2**	**N/A**	**2**	**N/A**	**2**	**N/A**	**2**
GSM8K	56.18	49.05	54.97	54.13	**56.56**	54.81	52.54	51.33
MathQA	36.72	36.68	40.10	40.03	40.07	40.07	40.77	**40.94**
MATH	25.02	24.56	**24.74**	23.30	24.60	19.12	21.80	21.80
ARC-C	33.45	33.28	40.44	40.36	40.87	**41.13**	39.33	39.51
ARC-C-ZH	30.97	31.14	33.87	**33.96**	33.70	33.36	33.53	33.36
ARC-E	65.66	65.74	**72.31**	72.22	72.20	71.97	71.89	71.51
C-Valid	54.90	54.61	55.87	56.02	**57.80**	56.76	56.69	55.79
CMMLU	53.38	52.97	**53.45**	53.00	52.96	53.18	52.61	52.93
HellaSwag	40.99	40.99	39.75	39.78	**39.87**	39.82	39.72	39.64
HumanEval	28.05	28.05	**37.80**	35.98	35.37	32.32	32.93	34.76
LAMBADA	53.56	53.79	50.16	50.09	**50.30**	50.22	49.84	49.70
MMLU	52.69	51.91	53.42	53.09	**53.43**	53.17	52.95	52.68
PIQA	69.64	69.75	69.75	69.48	**70.13**	69.86	70.08	70.02
XStoryCloze	59.23	59.43	58.84	**58.97**	58.50	58.44	57.97	57.97
BBH	40.65	36.25	**38.81**	32.13	37.83	34.71	38.07	34.39
Avg.	46.74	45.88	**48.29**	47.50	48.28	47.26	47.38	47.09

Table 7 shows that PHSA outperforms InfLLM v2 (47.38) with an average score of 48.28 under both dense inference and training-inference consistent (Top-K=2) settings at 20B tokens, achieving performance comparable to dense attention (48.29). Specifically, PHSA achieves the best or second-best performance on the vast majority of benchmarks.

E Sensitivity to λ

We evaluate different values of $1-\lambda$ on Qwen3-0.6B at 16K context length in Table 8. The value $1-\lambda=0$ corresponds to average pooling, equivalent to InfLLM v2, while $1-\lambda=1$ uses only the punctuation representation.

Table 8: NIAH scores for different values of $1-\lambda$ on Qwen3-0.6B at 16K context length.

Top-K	1.0	0.5	0.2	0.1	0.05	0 (InfLLM v2)
16	85.0	90.2	90.6	92.0	91.4	92.0
8	82.2	81.2	89.0	88.8	88.8	88.8
4	73.0	77.4	84.0	84.2	82.8	81.0
2	60.6	68.8	73.0	73.6	72.2	65.4

F Theoretical Analysis of Semantic Fidelity in PHSA

F.1 Notations

First, we formalize the key notations used in our theoretical analysis, which are fully consistent with the main text:

- For the t-th token block $\boldsymbol{B}_t = \{\boldsymbol{k}_{t\cdot m}, \boldsymbol{k}_{t\cdot m+1}, \ldots, \boldsymbol{k}_{(t+1)\cdot m-1}\}$ with block size m, let S_{left} and S_{right} denote the two adjacent semantic segments partitioned by punctuation tokens within $\boldsymbol{B}_t$.
- $\boldsymbol{M}_0(\boldsymbol{B}_t) = \frac{1}{m}\sum_{i\in\boldsymbol{B}_t}\boldsymbol{k}_i$: Global semantic representation via average pooling, consistent with Eq. (6) in the main text.
- $\boldsymbol{M}_p(\boldsymbol{B}_t)$: Punctuation-aware semantic representation via pooling over punctuation tokens $\mathcal{P}\cap\boldsymbol{B}_t$, consistent with Eq. (7) in the main text.
- $I(X;Y)$: Mutual information between random variables X and Y, measuring the amount of information obtained about one variable through observing the other.
- $\alpha_{l,p}$: Attention weight between the l-th query token $\boldsymbol{q}_l$ and punctuation token p, defined as $\alpha_{l,p} = \text{Softmax}(\boldsymbol{q}_l^\top \boldsymbol{k}_p/\sqrt{d_k})$, consistent with the attention formulation in Eq. (2) of the main text.
- $\text{Var}(\cdot)$ and $\text{Cov}(\cdot)$: Variance and covariance of random variables, used for optimal parameter derivation.

F.2 Information Loss Lower Bound of Average Pooling

The global representation $\boldsymbol{M}_0(\boldsymbol{B}_t)$ obtained via average pooling uniformly averages all tokens in the block, leading to a lower bound on the mutual information with the block's semantic structure:

$$I(\boldsymbol{M}_0(\boldsymbol{B}_t);\boldsymbol{B}_t) \le \frac{1}{m}\sum_{i\in\boldsymbol{B}_t} I(\boldsymbol{k}_i;\boldsymbol{B}_t). \tag{F.1}$$

Moreover, the mutual information between $\boldsymbol{M}_0(\boldsymbol{B}_t)$ and the semantic boundary $(S_{\text{left}}, S_{\text{right}})$ is diluted due to uniform averaging:

$$I(\boldsymbol{M}_0(\boldsymbol{B}_t);S_{\text{left}},S_{\text{right}}) \le \frac{1}{m}\sum_{i\in\boldsymbol{B}_t} I(\boldsymbol{k}_i;S_{\text{left}},S_{\text{right}}). \tag{F.2}$$

Proof. By the data processing inequality, applying a deterministic function (average pooling) to the token set $\{\boldsymbol{k}_i\}_{i\in\boldsymbol{B}_t}$ cannot increase mutual information:

$$I(\boldsymbol{M}_0(\boldsymbol{B}_t);\boldsymbol{B}_t) \le I(\{\boldsymbol{k}_i\}_{i\in\boldsymbol{B}_t};\boldsymbol{B}_t). \tag{F.3}$$

Since $\boldsymbol{M}_0(\boldsymbol{B}_t)$ is a linear convex combination of $\{\boldsymbol{k}_i\}$, by the convexity of mutual information in the mixture distribution, we have:

$$I(\boldsymbol{M}_0(\boldsymbol{B}_t);\boldsymbol{B}_t) \le \frac{1}{m}\sum_{i\in\boldsymbol{B}_t} I(\boldsymbol{k}_i;\boldsymbol{B}_t). \tag{F.4}$$

The Eq. (F.2) follows analogously by replacing the full block $\boldsymbol{B}_t$ with the semantic boundary pair $(S_{\text{left}}, S_{\text{right}})$. □

F.3 Information Fidelity Upper Bound of Punctuation Anchoring

For any punctuation token $p \in \mathcal{P} \cap \boldsymbol{B}_t$ and non-punctuation content token $c \in \boldsymbol{B}_t \setminus \mathcal{P}$, the mutual information between p and the semantic boundary is strictly greater than that of c:

$$I(p; S_{\text{left}}, S_{\text{right}}) \geq I(c; S_{\text{left}}, S_{\text{right}}). \tag{F.5}$$

This is because punctuation tokens receive significantly higher attention weights in pre-trained LLMs, and attention weights serve as a valid upper bound on token-wise mutual information.

Proof. Prior work Chen et al. [2024a], Razzhigaev et al. [2025] has empirically verified that punctuation tokens have significantly higher expected attention weights than non-punctuation content tokens across all attention layers and heads:

$$\mathbb{E}_{l\in[1,L]}[\alpha_{l,p}] \gg \mathbb{E}_{l\in[1,L]}[\alpha_{l,c}]. \tag{F.6}$$

Theoretically, the attention weight $\alpha_{l,p}$ reflects the semantic relevance between query $\boldsymbol{q}_l$ and key $\boldsymbol{k}_p$, which can be viewed as an upper bound on the pointwise mutual information $I(\boldsymbol{q}_l; \boldsymbol{k}_p)$ Luong et al. [2015]. Averaging over all query positions, we get:

$$\mathbb{E}_l[I(\boldsymbol{q}_l; \boldsymbol{k}_p)] \geq \mathbb{E}_l[I(\boldsymbol{q}_l; \boldsymbol{k}_c)]. \tag{F.7}$$

Since S_{left} and S_{right} are exactly the adjacent semantic segments partitioned by punctuation token p, $\boldsymbol{k}_p$ inherently encodes the core boundary information between them, which directly leads to:

$$I(\boldsymbol{k}_p; S_{\text{left}}, S_{\text{right}}) \geq I(\boldsymbol{k}_c; S_{\text{left}}, S_{\text{right}}). \tag{F.8}$$

□

F.4 Optimality of Dual-Branch Fusion in PHSA

The fused block representation $\boldsymbol{M}(\boldsymbol{B}_t) = \lambda \boldsymbol{M}_0(\boldsymbol{B}_t) + (1-\lambda)\boldsymbol{M}_p(\boldsymbol{B}_t)$ with $\lambda \in [0, 1]$ (consistent with Eq. (8) in the main text) achieves strictly higher semantic fidelity than pure average pooling:

$$I(\boldsymbol{M}(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}}) \geq I(\boldsymbol{M}_0(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}}). \tag{F.9}$$

Moreover, the theoretically optimal value of the gating parameter λ is derived as $\lambda^* = 1 - r$, where $r = |\mathcal{P} \cap \boldsymbol{B}_t|/m$ denotes the ratio of punctuation tokens in the block.

Proof. By the data processing inequality and the convexity of mutual information, since $\boldsymbol{M}(\boldsymbol{B}_t)$ is a convex combination of $\boldsymbol{M}_0(\boldsymbol{B}_t)$ and $\boldsymbol{M}_p(\boldsymbol{B}_t)$, we have:

$$I(\boldsymbol{M}(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}}) \geq \lambda I(\boldsymbol{M}_0(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}}) + (1-\lambda) I(\boldsymbol{M}_p(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}}). \tag{F.10}$$

From Section F.3, we have $I(\boldsymbol{M}_p(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}}) \geq I(\boldsymbol{M}_0(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}})$. Substituting this into the above inequality, we directly get:

$$I(\boldsymbol{M}(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}}) \geq I(\boldsymbol{M}_0(\boldsymbol{B}_t); S_{\text{left}}, S_{\text{right}}). \tag{F.11}$$

To derive the optimal λ, we minimize the variance of the fused representation (a widely used proxy for information loss in representation learning). The variance of $\boldsymbol{M}(\boldsymbol{B}_t)$ is formulated as:

$$\text{Var}(\boldsymbol{M}(\boldsymbol{B}_t)) = \lambda^2 \text{Var}(\boldsymbol{M}_0) + (1-\lambda)^2 \text{Var}(\boldsymbol{M}_p) + 2\lambda(1-\lambda)\text{Cov}(\boldsymbol{M}_0, \boldsymbol{M}_p). \tag{F.12}$$

Taking the first-order derivative with respect to λ and setting it to zero (the minimum variance condition), we get the closed-form solution:

$$\lambda^* = \frac{\text{Var}(\boldsymbol{M}_p) - \text{Cov}(\boldsymbol{M}_0, \boldsymbol{M}_p)}{\text{Var}(\boldsymbol{M}_0) + \text{Var}(\boldsymbol{M}_p) - 2\text{Cov}(\boldsymbol{M}_0, \boldsymbol{M}_p)}. \tag{F.13}$$

Empirically, since punctuation tokens are a sparse subset of the block, by the law of large numbers for small sample sizes, we have $\text{Var}(\boldsymbol{M}_p) \approx \text{Var}(\boldsymbol{M}_0)/r$. Meanwhile, the global representation and punctuation representation are highly correlated, so $\text{Cov}(\boldsymbol{M}_0, \boldsymbol{M}_p) \approx \text{Var}(\boldsymbol{M}_0)$. Substituting these two conditions, we simplify the optimal solution to $\lambda^* \approx 1 - r$. □

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract, Section 1 (Introduction)

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4 (Conclusion)

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix E (Theoretical Analysis of Semantic Fidelity in PHSA)

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 (Experiments, including 3.1 Experiment Settings, 3.2 Effect of PHSA, 3.3 Additional Discussion), Appendix D (Extra Experiment of Effect on general benchmarks)

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Section 3.1 (Experiment Settings), References

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3.1 (Experiment Settings), Section 3.2 (Effect of PHSA)

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 3.2 (Effect of PHSA), Section 3.3 (Additional Discussion), Appendix D (Extra Experiment of Effect on general benchmarks)

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 3.1 (Experiment Settings)

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Section 4 (Conclusion), References

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Section 3.1 (Experiment Settings), References

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.