Beyond the Efficiency-Performance Trade-off: Semantic Foundation Attention

Anonymous authors
Paper under double-blind review

Abstract

The quadratic computational complexity of self-attention presents a significant challenge for scaling Transformer architectures to longer sequences. While existing approaches pursue efficiency through sparse approximation or hardware optimization, they operate under the assumption that the input token sequence remains immutable. We propose Semantic Foundation Attention (SFA), which introduces semantic reconstruction—a paradigm that dynamically reconfigures the computational structure based on semantic relationships during attention computation. SFA employs two complementary strategies: similarity merging consolidates semantically aligned tokens through vector addition to preserve and amplify signal strength, while difference merging exploits orthogonality properties in high-dimensional embedding spaces to efficiently integrate complementary information. We implement custom CUDA compute kernels for SFA that decompose the generated dynamic attention patterns into diagonal and rectangular computation domains, enabling efficient execution without explicitly storing the sparse matrix. Comprehensive evaluation on OLMoE architectures demonstrates that SFA consistently improves performance across multiple downstream benchmarks while reducing computational requirements. These results show that computational efficiency and model performance can be jointly optimized through semantically-aware attention computation, establishing semantic reconstruction as a viable paradigm for attention mechanism design.

1 Introduction

The quadratic computational complexity of the self-attention mechanism, the core of the Transformer architecture (Vaswani et al., 2017), has become a major computational bottleneck in scaling large language models. Existing optimization approaches, whether hardware-aware computational acceleration or algorithmic sparse approximations, are constrained by a common limitation: they all treat the input sequence as a static, semantically flat collection of tokens, forcing a difficult trade-off between brute-force full computation and lossy approximation.

We argue that the key to addressing this limitation stems from reconsidering how attention computation should be structured. Rather than optimizing around fixed token sequences, we propose that attention mechanisms should dynamically adapt their computational structure based on the semantic relationships within the input. This insight leads to a new attention paradigm: Semantic Foundation Attention (SFA), which replaces computation over redundant token sequences with efficient, adaptive semantic synthesis.

SFA introduces semantic reconstruction—a process that dynamically consolidates semantically related tokens into higher-level, information-dense units during attention computation. This approach is driven by two complementary strategies: similarity merging, which amplifies signal strength by consolidating semantically aligned tokens through vector addition, and difference merging, which leverages orthogonal geometry in high-dimensional embedding spaces to efficiently integrate complementary information. Unlike preprocessing approaches, these operations occur within the attention mechanism itself, enabling the model to learn optimal semantic structures through end-to-end training.

To achieve practical efficiency, we developed a complete suite of specialized CUDA kernels that decompose SFA's dynamic computation into independent diagonal and rectangular domains. This implementation avoids explicit sparse matrix storage while maximally leveraging dense compute capabilities of modern GPUs. The system seamlessly integrates with existing training frameworks and maintains full compatibility with automatic differentiation.

We conducted comprehensive evaluations on medium-scale pre-training experiments using OLMoE architectures and found encouraging results: SFA-based models not only maintain performance relative to optimized standard attention baselines but demonstrate consistent improvements across multiple downstream benchmarks. These findings challenge the conventional assumption that efficiency and performance represent a zero-sum trade-off in attention mechanisms. Furthermore, we observe that SFA's compression effectiveness increases with the model's semantic understanding capabilities, suggesting a beneficial co-evolution between efficiency and model intelligence.

The principle of semantic-aware computation introduced by SFA provides a foundation for building more efficient and capable language models. By embedding semantic understanding directly within attention mechanisms, this work establishes a new direction for attention optimization that achieves joint improvements in both computational efficiency and model performance.

2 Related Work

The quest to accelerate attention mechanisms has pursued two primary directions: algorithmic sparse approximation and hardware-aware computational optimization, both constrained by treating the input token sequence as immutable.

Algorithmic Sparse Approximation reduces computational complexity through selective attention connection discarding via various sparsity patterns. These include structured sparsity (Child et al., 2019), fixed patterns combining sliding windows and global attention (Beltagy et al., 2020; Ainslie et al., 2020), random and global patterns (Zaheer et al., 2020), content-aware clustering (Roy et al., 2020; Wang et al., 2021),kernel-based approximations (Choromanski et al., 2021; Wang et al., 2020), dynamic hierarchical strategies (Lou et al., 2024; Yuan et al., 2025), and doubly stochastic methods (Sander et al., 2022). Comprehensive surveys are provided in (Tay et al., 2022; Farina et al., 2024). Despite algorithmic sophistication, these methods typically involve trade-offs between computational efficiency and information preservation.

Hardware-Aware Computational Optimization, exemplified by FlashAttention (Dao et al., 2022), maintains mathematical exactness while optimizing computational flow through IO-aware algorithms and tiling strategies. These approaches achieve practical acceleration without altering attention's mathematical definition, yet cannot transcend the theoretical quadratic complexity ceiling.

Token-Level Optimization directly modifies the computational substrate through token manipulation. Token merging approaches include spectrum-preserving methods using SVD (Tran et al., 2025), adaptive local-global strategies (Norouzi et al., 2024), and application-specific techniques (taihang Hu et al., 2024; Wu et al., 2025). Token pruning methods achieve efficiency through strategic elimination based on learned importance scores (Kim et al., 2022) or dynamic sparsification (He et al., 2024; Marchetti et al., 2025; Xiuying, 2025). These methods uniformly employ similarity-based heuristics followed by averaging-based aggregation as preprocessing steps, fundamentally limited by their inability to handle complementary information and reliance on information-diluting operations.

SFA's Paradigmatic Orthogonality: Semantic Foundation Attention operates at the semantic representation level rather than the computational pattern level, embedding semantic understanding directly within attention computation through dynamic semantic reconstruction. Unlike previous methods that modify inputs before attention computation (token merging/pruning), modify attention during computation (sparse patterns), or accelerate attention implementation (hardware optimizations), SFA optimizes the semantic substrate itself. Crucially, SFA's semantic-level optimizations are fully compatible with existing efficient attention methods, enabling seamless integration with sparse attention patterns, hardware optimizations, and other computational accelerations for compound efficiency gains.

This paradigmatic orthogonality dissolves the traditional trade-off between efficiency and performance by establishing semantic-aware computation as a new optimization axis, distinct from and compatible with existing approaches. SFA's contribution provides a semantic foundation that enhances any attention implementation through intelligent substrate optimization.

3 Semantic Foundation Attention

Traditional attention mechanisms assume that attention computation should be performed over original, unmodified token sequences. This assumption constrains all optimization approaches to either approximate full computation through sparse patterns or accelerate computation through hardware optimization, both treating sequence structure as fixed. Semantic Foundation Attention (SFA) provides an alternative by making the attention computation structure itself adaptive. SFA performs attention computation over dynamically reconstructed representations that better reflect the information structure of the input, recognizing and consolidating redundant or complementary information within sequences.

The key insight motivating SFA is that well-trained language models naturally develop geometric structures in their embedding spaces where related concepts exhibit predictable relationships—similar concepts align in direction while complementary concepts maintain orthogonality. SFA leverages and shapes these geometric properties through training to enable joint optimization of computational efficiency and model performance.

The SFA framework consists of three key components: token consolidation strategies based on relationships in the embedding space; multi-head specialization mechanisms enabling different attention heads to focus on different relationship types; and efficient computational implementation that decomposes dynamic attention patterns into hardware-friendly operations.

3.1 Dynamic Token Consolidation through Learned Geometric Structure

The core mechanism of SFA relies on exploiting geometric properties that emerge in high-dimensional embedding spaces during language model training. Our approach is motivated by the observation that well-trained models often develop structured representations (Ethayarajh, 2019) where semantically related tokens exhibit predictable geometric relationships—similar concepts tend to align in direction while complementary concepts maintain relative orthogonality. Rather than assuming this structure exists inherently, we design SFA to learn and leverage these relationships through end-to-end training.

Based on this motivation, we introduce two complementary token consolidation strategies that operate on adjacent token pairs during attention computation. Similarity Merging addresses redundancy by consolidating tokens with aligned representations. For example, adjacent repetitive tokens like ”very very” typically exhibit high directional similarity. For such cases, we apply vector addition: $K_i' = K_{i-1} + K_i$. This strategy preserves semantic direction while naturally encoding emphasis through increased magnitude, contrasting with averaging operations that dilute signal strength. Difference Merging handles complementary information by consolidating orthogonal tokens. For example, semantically complementary pairs like ”azure” and ”sky” may exhibit approximate orthogonality in embedding space. When adjacent tokens are approximately orthogonal, their vector sum creates a compressed representation that the model learns to interpret effectively. The key insight is that through training, the model can learn when the attention effect $\text{Attention}(Q, K_{i-1} + K_i)$ provides a beneficial approximation to the separate computations $\text{Attention}(Q, K_{i-1}) + \text{Attention}(Q, K_i)$.

Learning Framework: The geometric structure enabling these consolidation strategies emerges through joint optimization of the primary language modeling objective and a compression-aware auxiliary loss. This auxiliary loss guides the embedding space geometry while the main task ensures that consolidation decisions serve the downstream objectives. The compression loss balances merge quality (ensuring appropriate geometric relationships) with merge quantity (encouraging sufficient compression). We define the quality assessment components as $\mathcal{L}_{\text{sim}}(i,j) = (1 - d_p)^2$ and $\mathcal{L}_{\text{diff}}(i,j) = d_p^2$, where d_p represents cosine similarity between tokens. The complete compression loss combines quality and quantity objectives:

$$\mathcal{L}_{\text{comp}} = \left(\frac{\sum_{i,j} (\mathcal{L}_{\text{sim}}(i,j) \cdot M_{\text{sim}}(i,j) + \mathcal{L}_{\text{diff}}(i,j) \cdot M_{\text{diff}}(i,j))}{N_{\text{merged}}} - \frac{N_{\text{merged}}}{N_{\text{total\_pairs}}} \right) \cdot I_{\text{factor}} \tag{1}$$

where $M_{\text{sim/diff}}$ are strategy masks, N_{merged} is the number of merges, and I_{factor} is a scaling hyperparameter. This auxiliary loss is jointly optimized with the main language modeling loss $\mathcal{L}_{\text{LM}}$, creating a feedback mechanism where consolidation effectiveness improves alongside the model's representational capabilities. The model learns both when to merge tokens and how to interpret the resulting consolidated representations effectively.

3.2 Multi-Head Specialization and Normalization Mechanisms

SFA's effectiveness stems from how it adaptively applies merging strategies to complex language data through two complementary mechanisms: multi-head specialization and head-wise pre-normalization.

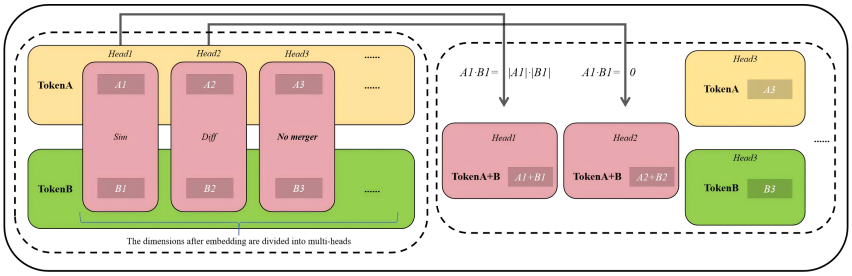


Figure 1: Illustration of SFA's Multi-Head Specialization mechanism. For the same input pair of Token A and Token B, different attention heads can autonomously learn and adopt distinct merging strategies: Head 1 performs similarity merging, Head 2 executes difference merging, while Head 3 opts not to merge, thereby capturing diverse semantic relationships in different subspaces.

For the same pair of adjacent tokens, their relationship can vary significantly depending on semantic context. For example, in ”bank of the river” versus ”investment bank,” the polysemous word ”bank” exhibits different relationships to its neighbors. The multi-head attention mechanism provides a natural framework for capturing such multifaceted relationships. SFA builds upon this by enabling different attention heads to specialize in different consolidation strategies, creating a multi-perspective analysis framework.

As illustrated in Figure 1, we divide attention heads into two groups: H_{sim} specializes in similarity merging, while H_{diff} focuses on difference merging. This specialization allows the model to simultaneously capture both commonalities and differences within a single architecture. Through training, heads in H_{sim} learn to project semantically similar tokens into aligned directions, while heads in H_{diff} learn to project complementary tokens into orthogonal directions.

To ensure stable and accurate merging decisions, we introduce head-wise pre-normalization. Unlike standard models that apply a single LayerNorm before multi-head attention, we apply RMS Normalization independently to the Q and K vectors of each attention head. RMSNorm stabilizes training by rescaling vector variance while preserving relative magnitude differences, unlike strict normalization that enforces unit L2 norms. The formulation is: $\text{RMSNorm}(x) = g \cdot \frac{x}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2 + \epsilon}}$, where g is a learnable scaling parameter.

Applying RMSNorm at the head level creates separate, standardized comparison spaces for each head, encouraging different heads to learn distinct projection functions and focus on different dimensional combinations for relationship discovery. This approach provides numerical stability for merging decisions while preserving the geometric properties necessary for effective consolidation. Importantly, we only normalize Q and K vectors, leaving V vectors unchanged. This asymmetric design ensures that similarity merging's signal amplification effect (increased magnitude) is preserved in the value vectors, where it contributes to the final attention output weighting. This combination of multi-head specialization and head-wise pre-normalization creates a novel attention mechanism design that achieves effective information consolidation while maintaining decision accuracy and numerical stability.

3.3 Computational Flow and Implementation of SFA

SFA's implementation performs semantic reconstruction during attention computation, not as preprocessing. This enables the attention mechanism to adapt its computational structure based on the semantic content being processed, rather than applying fixed patterns. SFA transforms dynamic attention patterns into efficient computational workflows through structured decomposition. As illustrated in Figure 2, SFA's dynamic attention pattern decomposes into two independent computational domains: diagonal and rectangular regions. This decomposition enables efficient parallel computation without explicit sparse matrix storage. SFA's computational flow consists of four stages: semantic merging and data reorganization, where consolidation decisions reorganize memory lay-

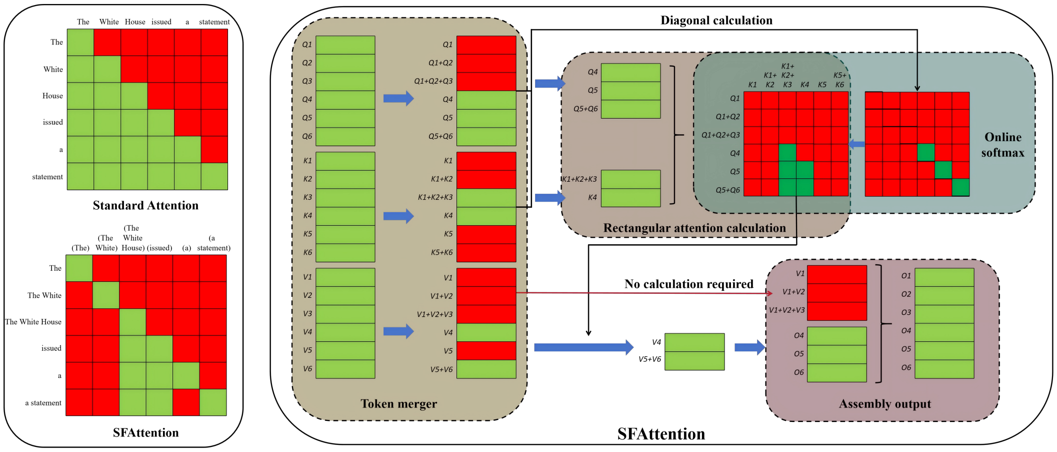


Figure 2: Complete overview of SFA computational flow. Upper left: Fixed causal mask of standard attention; Lower left: Dynamic semantic-aware mask of SFA; Right: Complete computational pipeline of SFA, including token merging, diagonal computation, rectangular computation, online Softmax fusion, and result assembly. This architecture avoids sparse matrix memory instantiation, achieving optimization of memory and computation.

out for optimal access patterns; diagonal domain computation of self-attention scores for all tokens through $s_{k,i} = \frac{Q_i K_i^T}{\sqrt{d_k}}$; rectangular domain attention on compressed Query, Key, and Value representations; and cross-domain fusion using online Softmax operations for mathematically consistent attention weight normalization.

Rather than computing attention over the full $n \times n$ matrix, SFA computes attention over two regions: an n-dimensional diagonal and an $(n-k) \times k$ rectangular region, where k represents tokens after compression. This reduces computational complexity from $\mathcal{O}(n^2 d)$ to $\mathcal{O}(nd + (n-k) \times k \times d)$, achieving efficiency gains when $k \ll n$. SFA's implementation avoids the overhead of sparse attention patterns by never explicitly constructing sparse matrices. The diagonal-rectangular decomposition enables all computations on dense memory blocks with optimal memory access patterns. The system maintains numerical stability through online Softmax algorithms that handle fusion of attention weights computed at different scales.

Importantly, SFA operates at the semantic representation level and is orthogonal to existing efficient attention methods such as Linformer, Performer, and Longformer, which focus on computational pattern optimization. SFA can be integrated with these approaches, enabling compound efficiency gains through semantic reconstruction combined with algorithmic sparsification.

Detailed CUDA implementation, including optimized memory access patterns, kernel fusion strategies, and backward pass algorithms, is provided in Appendix D.

3.4 Theoretical Analysis: Mathematical Foundations and Performance Mechanisms of SFA

SFA's effectiveness builds upon the linear nature of vector operations in attention computation and the geometric properties of high-dimensional embedding spaces. When merging adjacent key vectors K_i, K_{i-1} into $K' = K_i + K_{i-1}$, the attention score for query Q_j exhibits linearity: $s'_j = \frac{Q_j \cdot (K_i + K_{i-1})}{\sqrt{d_k}} = s_{j,i} + s_{j,i-1}$. Since pre-softmax computations are linear, this provides the mathematical foundation for merging operations.

The critical transformation occurs through the softmax activation, which converts linear score combinations into multiplicative attention weight relationships. Let the original attention weights be $a_{j,i} = \exp(s_{j,i})/Z$ and $a_{j,i-1} = \exp(s_{j,i-1})/Z$, where $Z = \sum_{k=1}^{n} \exp(s_{j,k})$. After merging, the new key sequence produces the merged weight $a'_{j,\text{merged}} = \exp(s_{j,i} + s_{j,i-1})/Z'$.

The key insight lies in how exponential transformation converts linear combinations to multiplicative relationships: $\exp(s_{j,i} + s_{j,i-1}) = \exp(s_{j,i}) \cdot \exp(s_{j,i-1})$. While this transforms the original additive attention effect $a_{j,i} + a_{j,i-1}$ into a multiplicative form, the normalization factor adjustment enables adaptive attention redistribution that can effectively approximate the original computation under appropriate conditions. The normalization factor becomes:

$$Z' = Z - \exp(s_{j,i}) - \exp(s_{j,i-1}) + \exp(s_{j,i}) \cdot \exp(s_{j,i-1}) \tag{2}$$

This transformation creates two distinct operational regimes based on token relationships. When K_i and K_{i-1} are semantically similar, for query Q_j we have $s_{j,i-1} \approx s_{j,i} = s$. The merged score becomes $s'_{\text{merged}} = 2s$, and due to exponential convexity, when $s > \log(2)$, we have $\exp(2s) > 2\exp(s)$, thus $Z' > Z$. This creates enhanced competitive advantage for the merged unit. For any other token k, the competitive ratio change is:

$$\frac{a'_{j,\text{merged}}/a'_{j,k}}{(a_{j,i} + a_{j,i-1})/a_{j,k}} = \frac{\exp(s)}{2} \tag{3}$$

When $s > \log(2)$, this ratio exceeds 1, concentrating attention more decisively onto semantically coherent units.

Conversely, when K_i and K_{i-1} are orthogonal, adaptive routing emerges. If query Q_j is relevant only to K_i with $s_{j,i} = s$ and $s_{j,i-1} \approx 0$, then $s'_{\text{merged}} = s$, $Z' = Z - 1$, and $a'_{j,\text{merged}} \approx a_{j,i}$ when $Z \gg 1$, making the merging operation transparent. If Q_j is relevant to both tokens, the merged unit gains multiplicative competitive advantage through $\exp(s_1 + s_2) = \exp(s_1) \cdot \exp(s_2)$.

SFA achieves adaptive optimization through joint training with task loss $\mathcal{L}_{\text{LM}}$ and geometric regularization $\mathcal{L}_{\text{comp}}$. For similarity merging, the gradient $\frac{\partial \mathcal{L}_{\text{sim}}}{\partial K_i} = -2(1 - \cos(K_i, K_{i-1}))\frac{\partial \cos(K_i, K_{i-1})}{\partial K_i}$ produces adaptive-strength updates: strong gradients when similarity is low to promote alignment, weak gradients when highly aligned to prevent over-merging. For difference merging, $\frac{\partial \mathcal{L}_{\text{diff}}}{\partial K_i} = 2\cos(K_i, K_{i-1})\frac{\partial \cos(K_i, K_{i-1})}{\partial K_i}$ promotes orthogonalization.

The approximation quality can be characterized mathematically. For similarity merging when $\cos\theta \to 1$ and $\|K_i\| \approx \|K_{i-1}\|$, the angular error ϕ satisfies $\sin\phi \to 0$. For difference merging, approximation fidelity is defined as $F = \cos(\text{Attention}(Q, K_{\text{merged}})V, \text{Attention}(Q, K_{\text{separate}})V)$, where $F \to 1$ under orthogonal conditions.

This analysis demonstrates that SFA transforms semantic understanding into structured attention redistribution through softmax's exponential transformation, achieving signal concentration rather than information loss while maintaining mathematical consistency through adaptive normalization.

4 Experiments

To validate the effectiveness of Semantic Foundation Attention (SFA), we design a controlled experimental framework that evaluates its performance against standard attention mechanisms while investigating the contributions of its core components. Our evaluation focuses on demonstrating SFA's fundamental mechanisms and establishing its viability as an attention optimization approach.

4.1 Experimental Setup

We conduct our evaluation using the OLMoE architecture (Muennighoff et al., 2025), which provides a well-established foundation for controlled comparison studies. The mixture-of-experts design offers particular advantages for evaluating attention mechanisms, as it separates attention performance from feed-forward capacity effects, enabling cleaner assessment of attention-specific improvements.

Our experimental design encompasses two complementary scales to validate SFA's effectiveness across different parameter regimes and sequence length configurations. The first configuration employs OLMoE models ranging from 1B-7B parameters, trained on 0.3 billion tokens from the DCLM dataset (Li et al., 2024) with maximum sequence length 4096. This scale enables evaluation of SFA's early convergence properties and mechanism validation under representative training conditions. The second configuration uses OLMoE models from 0.25B-1.75B parameters, trained on 3 billion tokens with maximum sequence length 1024. This extended training regimen allows assessment of SFA's performance characteristics under more thorough optimization, providing insights into long-term effectiveness patterns.

For each configuration, we train both a baseline model using standard attention and a comparison model integrating SFA. All other aspects remain identical—network architecture, optimizer parameters, learning rate scheduling, random seeds, and training data flow—ensuring that performance differences can be attributed specifically to the attention mechanism. The SFA models incorporate head-wise pre-normalization, applying RMS Normalization independently to Query and Key vectors of each attention head to provide stable numerical conditions for consolidation decisions.

Following pretraining, we evaluate all models on a comprehensive suite of established benchmarks that assess different cognitive capabilities: physical common sense reasoning (PIQA (Bisk et al., 2019)), general reasoning (WinoGrande (Sakaguchi et al., 2019), CommonsenseQA (Talmor et al., 2019)), scientific knowledge (SciQ (Welbl et al., 2017), ARC-Easy (Clark et al., 2018)), social interaction understanding (SocialIQA (Sap et al., 2019)), and multi-domain knowledge question answering (MMLU (Hendrycks et al., 2021)). This evaluation scope provides a thorough assessment of model performance across diverse reasoning tasks, enabling validation of SFA's general applicability rather than task-specific optimization.

4.2 Core Experimental Results

Our experiments provide a direct comparison of performance between Semantic Foundation Attention (SFA) and standard attention mechanisms. We use FlashAttention as our baseline implementation, which maintains mathematical equivalence to standard attention while providing computational optimizations through improved memory access patterns. This choice ensures fair comparison by isolating SFA's semantic-level innovations from implementation-specific acceleration techniques.

Table 1: Performance comparison between SFA and standard attention mechanisms

Model	PIQA	WinoGrande	SciQ	ARC-Easy	CommonsenseQA	SocialIQA	MMLU-Humanities	MMLU-STEM
1B-7B Scale								
Flash	0.5680	**0.5040**	0.4840	0.3511	**0.2733**	0.4520	0.2553	0.1809
SFA	**0.6095**	0.4819	**0.5551**	**0.3574**	0.2571	**0.4743**	**0.2624**	**0.1977**
0.25B-1.75B Scale								
Flash	0.6175	0.5130	0.6200	0.4404	0.2867	0.4099	0.2520	0.2440
SFA	**0.6197**	**0.5249**	**0.6360**	**0.4684**	**0.2957**	**0.4132**	**0.2596**	**0.2636**

The results demonstrate that SFA achieves consistent performance improvements in practical pretraining scenarios while maintaining computational efficiency. We conducted evaluations on two different scales of OLMoE models, with detailed results shown in Table 1. In the 1B-7B scale experiments, SFA demonstrates positive performance trends across multiple benchmarks during the substantial pretraining phase. The model shows consistent improvements on tasks requiring both factual knowledge and reasoning capabilities, indicating that the consolidation mechanisms introduced by SFA can effectively support model development during training. The performance pattern suggests that dynamic compression helps models focus computational resources on core information structures, which translates into improved task performance.

The 0.25B-1.75B scale models, trained with more extensive optimization, provide evidence for SFA's effectiveness. Under these thorough training conditions, SFA achieves improvements across all evaluation metrics, demonstrating performance enhancement. This improvement pattern indicates that SFA's adaptive compression capabilities develop synergistically with the model's representational abilities through extended training.

The consistent improvement pattern across different scales and training regimens establishes SFA's viability as an attention optimization approach. The results support the hypothesis that computational efficiency and model performance can be jointly optimized through adaptive attention mechanisms, rather than requiring trade-offs between these objectives. As training progresses, SFA's consolidation effectiveness appears to co-evolve with the model's ability to develop structured representations, creating mutually reinforcing improvements in both efficiency and performance.

4.3 Ablation Study: Contributions of Individual Merging Strategies

To analyze the underlying mechanisms of SFA's performance, we conduct ablation experiments examining the individual contributions of similarity merging and difference merging strategies. The experiments use the OLMoE-0.25B-1.75B architecture under consistent training conditions, comparing four model variants: the complete SFA system, similarity-only merging, difference-only merging, and the FlashAttention baseline. The results in Table 2 reveal distinct performance patterns for different consolidation strategies. The similarity-only model performs well on tasks with extensive factual content and scientific definitions, such as SciQ, which contains high semantic coherence

Table 2: Ablation experimental results of different SFA merging strategies

Model	PIQA	Wino-Grande	SciQ	ARC-Easy	Common-senseQA	Social-IQA	MMLU-Humanities	MMLU-STEM
Flash	0.5620	0.5162	0.4800	0.3561	0.2572	0.3956	0.2427	**0.2644**
SFA	0.5588	**0.5201**	0.4860	**0.3614**	0.2547	**0.4171**	0.2293	0.2564
Simi	0.5598	0.5154	**0.4930**	0.3561	0.2580	0.4048	0.2373	0.2609
Diff	**0.5637**	0.4925	0.4780	0.3596	**0.2613**	0.4002	**0.2453**	0.2591

requirements. This suggests that similarity merging can effectively strengthen concept representations by consolidating semantically equivalent tokens, particularly benefiting knowledge-intensive tasks.

The difference-only model shows advantages on tasks requiring fine-grained discrimination and contextual reasoning, such as PIQA, CommonsenseQA, and MMLU-Humanities. These tasks often require distinguishing between subtle conceptual differences and processing diverse contextual clues. By consolidating orthogonal tokens, this strategy appears to preserve information diversity in the representation space, which may enhance discriminative capabilities for complex reasoning problems. The complete SFA system achieves optimal performance on tasks requiring comprehensive cognitive abilities, such as WinoGrande, ARC-Easy, and Social-IQA. This pattern suggests that the combination of both strategies enables the model to simultaneously construct coherent structures and preserve distinctive information across different semantic subspaces through the multi-head grouping mechanism.

These ablation results indicate that similarity and difference merging strategies address complementary aspects of language understanding. The similarity strategy appears to excel at consolidating redundant information and strengthening core concepts, while the difference strategy seems to preserve diverse information necessary for discriminative reasoning. The combined approach in complete SFA leverages both capabilities, enabling performance improvements across tasks with varying cognitive requirements.

4.4 Computational Performance and Complexity Analysis

Beyond model effectiveness improvements, SFA demonstrates computational efficiency advantages over standard attention mechanisms. We conducted throughput comparison tests against standard attention and FlashAttention implementation, measuring computational throughput of three attention implementations on NVIDIA A100 GPU across different sequence length configurations.

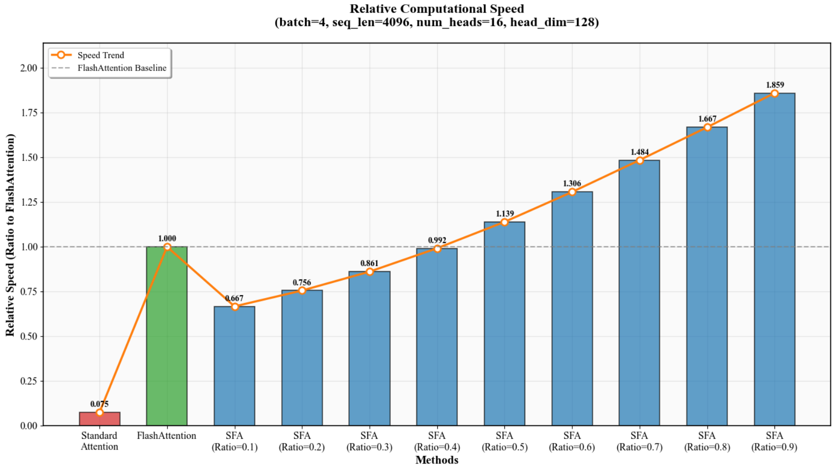


Figure 3: Computational throughput comparison for sequence length 4096. SFA demonstrates superior performance over both standard attention and FlashAttention as compression ratio increases, with advantages becoming more pronounced at longer sequence lengths.

From a theoretical complexity perspective, SFA's performance characteristics stem from its computational flow restructuring. Standard attention mechanisms have computational complexity of $\mathcal{O}(n^2d)$, where n is sequence length and d is head dimension. SFA's computational flow decomposes into three components: preprocessing stage encompassing similarity computation, merging decisions, and data reorganization with complexity $\mathcal{O}(nd)$; diagonal computation for self-attention scores with complexity $\mathcal{O}(nd)$; and rectangular computation on compressed sequences with complexity approximately $\mathcal{O}((1-p)n^2d)$, where p is the compression ratio.

SFA's total computational complexity becomes $\mathcal{O}(2nd + (1-p)n^2d)$. When sequence length n increases, the quadratic term dominates, and SFA achieves a $(1-p)$ reduction factor for the primary computational component while introducing linear overhead $\mathcal{O}(2nd)$.

The experimental results validate our theoretical analysis. In tests with sequence length 4096, as compression ratio increases, the quadratic term reduction effect becomes dominant, and SFA's computational throughput begins to exceed FlashAttention while maintaining consistent advantages over standard attention mechanisms. The performance improvement scales with sequence length, as shown in Figure 3. Additional performance comparison results across different sequence lengths are provided in the appendix.

This performance characteristic aligns with the trend toward longer contexts in large language models. SFA's approach of reducing computational complexity through adaptive semantic understanding provides a scalable solution for processing longer sequences. The experiments demonstrate that efficiency improvements can be achieved through algorithmic innovation that leverages the structure of language data, complementing hardware-level optimizations.

The combination of theoretical complexity reduction and practical performance improvements establishes SFA as a viable approach for attention optimization, particularly in scenarios requiring extended context processing capabilities.

5 Conclusion

In this paper, we propose Semantic Foundation Attention (SFA), a mechanism that modifies attention computation by incorporating token consolidation based on learned geometric relationships. We address limitations in existing attention mechanisms that treat input sequences as collections of independent units by introducing a consolidation process that dynamically combines adjacent tokens based on their embedding space relationships.

SFA introduces two consolidation strategies that utilize geometric properties of high-dimensional embedding spaces. Similarity merging consolidates semantically aligned tokens through vector addition, while difference merging handles orthogonal tokens by learning when their combination provides effective computational approximations. The effectiveness of these strategies is validated through comprehensive evaluation across multiple downstream benchmarks, where SFA models achieve performance improvements while reducing computational requirements.

The implementation translates dynamic attention patterns into efficient computation through domain decomposition, avoiding explicit sparse matrix storage. We designed specialized CUDA kernels that decompose the resulting patterns into diagonal and rectangular computational regions, enabling practical efficiency gains while maintaining compatibility with existing training frameworks.

Our experiments on OLMoE architectures provide evidence for SFA's effectiveness. Models equipped with SFA achieve consistent improvements across evaluation benchmarks while demonstrating computational efficiency advantages. The results indicate that computational efficiency and model performance can be jointly optimized through adaptive attention mechanisms, challenging the assumption that these objectives necessarily conflict.

Our computational performance analysis demonstrates that SFA achieves throughput improvements over standard attention implementations, with advantages that scale with sequence length. This characteristic positions SFA as a potentially valuable approach for applications requiring extended context processing, where the theoretical complexity reduction translates into measurable performance benefits.

SFA represents an approach toward attention mechanisms that adapt their computational structure based on learned semantic relationships. The work demonstrates that attention optimization can benefit from incorporating understanding of language structure, suggesting that the integration of semantic awareness and computational efficiency offers a productive direction for attention mechanism development. We believe that attention mechanisms that adapt to the content they process provide a foundation for developing more efficient language models.

REFERENCES

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. Etc: Encoding long and structured data in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, 2020.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:208290939.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

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, 2018. URL https://arxiv.org/abs/1803.05457.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings, 2019. URL https://arxiv.org/abs/1909.00512.

Mirko Farina, Usman Ahmad, Ahmad Taha, Hussein Younes, Yusuf Mesbah, Xiao Yu, and Witold Pedrycz. Sparsity in transformers: A systematic literature review. *Neurocomputing*, 582:127468, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2024.127468. URL https://www.sciencedirect.com/science/article/pii/S092523122400239X.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Efficient large vision-language models with dynamic token sparsification, 2024. URL https://arxiv.org/abs/2410.08584.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers, 2022. URL https://arxiv.org/abs/2107.00910.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. In A. Globerson,

L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 14200–14282. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/19e4ea30dded58259665db375885e412-Paper-Datasets_and_Benchmarks_Track.pdf.

Chao Lou, Zixia Jia, Zilong Zheng, and Kewei Tu. Sparser is faster and less is more: Efficient sparse attention for long-range transformers, 2024. URL https://arxiv.org/abs/2406.16747.

Michele Marchetti, Davide Traini, Domenico Ursino, and Luca Virgili. Efficient token pruning in vision transformers using an attention-based multilayer network. *Expert Systems with Applications*, 279:127449, 2025. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2025.127449. URL https://www.sciencedirect.com/science/article/pii/S0957417425010711.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. Olmoe: Open mixture-of-experts language models, 2025. URL https://arxiv.org/abs/2409.02060.

Narges Norouzi, Svetlana Sorlova, Daan de Geus, and Gijs Dubbelman. ALGM: Adaptive Local-then-Global Token Merging for Efficient Semantic Segmentation with Plain Vision Transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers, 2020. URL https://arxiv.org/abs/2003.05997.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

Michael E. Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention, 2022. URL https://arxiv.org/abs/2110.11773.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions, 2019. URL https://arxiv.org/abs/1904.09728.

taihang Hu, Linxuan Li, Joost van de Weijer, Hongcheng Gao, Fahad Khan, Jian Yang, Ming-Ming Cheng, Kai Wang, and Yaxing Wang. Token merging for training-free semantic binding in text-to-image synthesis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=tRRWoa9e80.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019. URL https://arxiv.org/abs/1811.00937.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey, 2022. URL https://arxiv.org/abs/2009.06732.

Hoai-Chau Tran, Duy MH Nguyen, Duy M Nguyen, Trung-Tin Nguyen, Ngan Le, Pengtao Xie, Daniel Sonntag, James Y Zou, Binh T Nguyen, and Mathias Niepert. Accelerating transformers with spectrum-preserving token merging. *Advances in Neural Information Processing Systems*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Shuohang Wang, Luowei Zhou, Zhe Gan, Yen-Chun Chen, Yuwei Fang, Siqi Sun, Yu Cheng, and Jingjing Liu. Cluster-former: Clustering-based sparse transformer for long-range dependency encoding, 2021. URL https://arxiv.org/abs/2009.06097.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions, 2017. URL https://arxiv.org/abs/1707.06209.

Haoyu Wu, Jingyi Xu, Hieu Le, and Dimitris Samaras. Importance-based token merging for efficient image and video generation, 2025. URL https://arxiv.org/abs/2411.16720.

Shui Xiuying. Token-level pruning in attention models. *Preprints*, March 2025. doi: 10.20944/preprints202503.0590.v1. URL https://doi.org/10.20944/preprints202503.0590.v1.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention, 2025. URL https://arxiv.org/abs/2502.11089.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 2020.

A Ethics Statement

The Semantic Foundation Attention (SFA) proposed in this research is a purely algorithmic and architectural innovation aimed at improving the computational efficiency and performance of Transformer models. We hereby declare the following ethics-related matters:

Data Usage: The training data used in this research is the publicly available DCLM dataset, which has undergone appropriate data filtering and processing. We have not collected or used any private data involving human subjects.

Potential Impact: As an attention mechanism optimization method, SFA itself does not generate harmful content or exacerbate existing bias issues. On the contrary, by improving model efficiency, SFA helps reduce the computational cost of large-scale language models, potentially promoting broader adoption of AI technology.

Environmental Impact: By reducing the computational complexity of attention mechanisms, SFA is expected to lower the energy consumption of model training and inference, having a positive environmental impact.

Transparency: We commit to fully open-sourcing the implementation code of SFA, ensuring that the research community can fully understand, verify, and improve our method.

B Reproducibility Statement

To ensure the complete reproducibility of this research, we have taken the following measures:

The CUDA implementation is open-sourced at https://anonymous.4open.science/r/SFAttention-0E1C, and the training code is open-sourced at https://anonymous.4open.science/r/DCAttention-4B2A.

Experimental Details: Section 4.1 of the paper provides detailed descriptions of experimental settings, including model architectures (OLMoE-1B-7B and 0.25B-1.75B), training data scales (0.3B and 3B tokens), sequence lengths (4096 and 1024), and other key hyperparameters. The open-sourced code contains detailed configuration files for all hyperparameters.

Evaluation Benchmarks: All downstream evaluations use standard public benchmark datasets, including PIQA, WinoGrande, SciQ, ARC-Easy, CommonsenseQA, SocialIQA, and MMLU, ensuring objectivity and comparability of results.

Random Seed Control: Random seeds were strictly controlled in experiments to ensure completely consistent results can be reproduced under the same hardware environment.

Hardware Environment: The open-sourced code already contains detailed descriptions of hardware configurations and software environment requirements.

C USE OF LLMS

During the writing process of this research, we used large language models only for the following limited purposes:

Grammar Proofreading: We used tools such as ChatGPT to perform grammar checking and polishing on the English expressions in the paper to improve the language quality and readability of the paper.

Limitation Statement: We strictly limited the scope of LLM usage. All research ideas, technical solutions, experimental designs, result analyses, and conclusions were completed entirely independently by human researchers. LLMs did not participate in any substantive research content creation.

Originality Guarantee: All core contributions of the paper, including SFA algorithm design, theoretical analysis, and experimental validation, are original work of the author team. The grammar modifications made using LLMs do not involve the generation or modification of any technical content.

D CUDA IMPLEMENTATION DETAILS

This section describes the CUDA implementation of Semantic Foundation Attention (SFA), detailing how the theoretical framework translates into efficient GPU computation through specialized kernel design.

D.1 DYNAMIC SEMANTIC RECONSTRUCTION ARCHITECTURE

SFA's implementation differs from token merging approaches through its integration of semantic reconstruction within attention computation rather than as preprocessing. This integration enables the attention mechanism to adapt its computational structure based on semantic content during execution.

The system transforms dynamically generated semantic structures into efficient computational workflows through domain decomposition. As shown in the computational framework, SFA's dynamic attention pattern decomposes into two independent computational regions: diagonal and rectangular domains. This decomposition eliminates the need for explicit sparse matrix storage while enabling parallel computation on dense memory blocks.

D.2 FORWARD PASS IMPLEMENTATION

SFA's forward pass processes dynamic sparse attention through four computational stages that convert the problem into hardware-optimized dense operations.

D.2.1 STAGE 1: COMPRESSED TENSOR CONSTRUCTION AND METADATA COMPUTATION

This stage determines compressed tensor dimensions Q', K', V' based on merging decisions and constructs them from original tensors Q, K, V while generating index mapping tables.

Algorithm 1 Compressed Tensor Length Computation

```
Require: Cumulative sequence lengths $\text{cu\_seqlens}_Q, \text{cu\_seqlens}_K \in \mathbb{Z}^{B+1}$
Require: Merging decisions $M \in \{0,1\}^{B \times H \times (S_K - 1)}$
Require: Initial continuous compression count $NFC \in \mathbb{Z}^{B \times H}$
Ensure: Compressed tensor lengths $\text{len}(Q')_{b,h}, \text{len}(K')_{b,h}$
for each batch-head pair $(b, h)$ in parallel do
    $\text{len}(Q')_{b,h} \leftarrow \text{len}(Q)_{b,h} - (NFC_{b,h} + 1)$
    Identify survivor indices not compressed by $M_{b,h}$ and $NFC_{b,h}$
    Store survivors in set $\mathcal{S}_{b,h}$
    $\text{len}(K')_{b,h} \leftarrow |\mathcal{S}_{b,h}| - 1$                ▷ Staggered design
    $\text{len}(V')_{b,h} \leftarrow |\mathcal{S}_{b,h}| - 1$
end for
Compute global prefix sum to generate $\text{cu\_seqlens}_{Q'}, \text{cu\_seqlens}_{K'V'}$
```

Algorithm 2 Compressed Tensor Data Population and Mapping

```
Require: Original tensors $Q, K, V \in \mathbb{R}^{N_{total} \times H \times d}$
Require: Survivor index sets $\mathcal{S}$
Ensure: Compressed tensors $Q', K', V'$
Ensure: Remapping tables $R_Q, R_K, R_V$, inverse mapping $R_Q^{-1}$, participant mask $M_P$
Construct compressed Q tensor:
for each token $Q_i$ where $i \geq (NFC + 1)$ in parallel do
    Compute write position $j$ via parallel prefix sum
    $Q'_j \leftarrow Q_i$
    $R_Q[j] \leftarrow i$, $R_Q^{-1}[i] \leftarrow j$
    $M_P[i] \leftarrow$ true
end for
Construct compressed K tensor (staggered):
for $j = 0$ to $|\mathcal{S}| - 2$ in parallel do
    $K'_j \leftarrow K_{\mathcal{S}[j]}$
    $R_K[j] \leftarrow \mathcal{S}[j]$
end for
Construct compressed V tensor (staggered):
for $j = 0$ to $|\mathcal{S}| - 2$ in parallel do
    $V'_j \leftarrow V_{\mathcal{S}[j+1]}$
    $R_V[j] \leftarrow \mathcal{S}[j+1]$
end for
```

D.2.2 Stage 2: Rectangular Domain Attention Computation

This stage performs causal attention computation on compressed tensors Q', K', V' using an optimized FlashAttention variable-length kernel. The kernel outputs three intermediate statistics for subsequent fusion:

Algorithm 3 Rectangular Domain Attention

```
Require: Compressed tensors $Q', K', V'$
Ensure: Row statistics $m'_r$, $l'_r$, $O'_{r,\text{unnorm}}$
for each row $r$ in parallel do
    $m'_r \leftarrow \max_c \left( s_{softmax} \cdot \frac{Q'_r (K'_c)^T}{\sqrt{d_k}} \right)$
    $l'_r \leftarrow \sum_c \exp \left( s_{softmax} \cdot \frac{Q'_r (K'_c)^T}{\sqrt{d_k}} - m'_r \right)$
    $O'_{r,\text{unnorm}} \leftarrow \sum_c \left( \exp \left( s_{softmax} \cdot \frac{Q'_r (K'_c)^T}{\sqrt{d_k}} - m'_r \right) \cdot V'_c \right)$
end for
```

D.2.3 Stage 3: Diagonal Domain Attention Computation

A parallel kernel computes self-attention scores for all original tokens, forming the diagonal component of the sparse pattern:

Algorithm 4 Diagonal Domain Attention

```
Require: Original tensors $Q, K$
Ensure: Diagonal scores $S_k$
for each position $i$ in parallel do
    if $i$ is within valid sequence length then
        $s_{k,i} \leftarrow s_{softmax} \cdot \frac{Q_i K_i^T}{\sqrt{d_k}}$
    else
        $s_{k,i} \leftarrow -\infty$
    end if
    $S_k[i] \leftarrow s_{k,i}$
end for
```

D.2.4 Stage 4: Cross-Domain Fusion and Final Output Construction

This stage merges results from diagonal and rectangular domains using online Softmax algorithms:

Algorithm 5 Online Softmax Fusion

```
Require: Rectangular statistics $m', l', O'_{\text{unnorm}}$
Require: Diagonal scores $S_k$, original $V$, participant mask $M_P$, inverse mapping $R_Q^{-1}$
Ensure: Final output $O$ and log-sum-exp $LSE$
for each token $i$ in parallel do
    if $M_P[i] = \text{false}$ then                                        ▷ Non-participants
        $O_i \leftarrow V_i$
        $LSE_i \leftarrow s_{k,i}$
    else                                                                  ▷ Participants
        $r \leftarrow R_Q^{-1}[i]$
        $LSE'_{\text{rect}} \leftarrow m'_r + \log(l'_r)$
        $LSE_{\text{diag}} \leftarrow s_{k,i}$
        $m_{i,\text{final}} \leftarrow \max(LSE'_{\text{rect}}, LSE_{\text{diag}})$
        $l_{i,\text{final}} \leftarrow e^{LSE'_{\text{rect}} - m_{i,\text{final}}} + e^{LSE_{\text{diag}} - m_{i,\text{final}}}$
        $\alpha_i \leftarrow \frac{e^{LSE'_{\text{rect}} - m_{i,\text{final}}}}{l_{i,\text{final}}}, \beta_i \leftarrow \frac{e^{LSE_{\text{diag}} - m_{i,\text{final}}}}{l_{i,\text{final}}}$
        $O'_{r,\text{norm}} \leftarrow O'_{r,\text{unnorm}} / l'_r$
        $O_i \leftarrow \alpha_i \cdot O'_{r,\text{norm}} + \beta_i \cdot V_i$
        $LSE_i \leftarrow m_{i,\text{final}} + \log(l_{i,\text{final}})$
    end if
end for
```

D.3 Backward Pass Implementation

SFA's backward pass follows the chain rule using domain separation, propagating gradients through four computational stages.

D.3.1 Stage 1: Gradient Preprocessing and Diagonal Gradient Computation

This kernel decomposes output gradients and computes diagonal gradients while preparing signals for rectangular domain backward propagation:

Algorithm 6 Gradient Preprocessing and Decomposition

```
Require: Output gradients $dO$, forward pass intermediate tensors
Require: Participant mask $M_P$, inverse mapping $R_Q^{-1}$
Ensure: Prepared gradients $dO'$, $dP'_{\text{sum}}$ for rectangular domain
Ensure: Updated gradient accumulators $dQ_{\text{accum}}, dK_{\text{accum}}, dV_{\text{accum}}$
for each token $i$ in parallel do
    $dP_{\text{sum},i} \leftarrow dO_i \cdot O_i$
    if $M_P[i] = \text{true}$ then                                    ▷ Participants only
        Recompute fusion weights $\alpha_i, \beta_i$ from forward pass
        $ds_{k,i} \leftarrow \beta_i(dO_i \cdot V_i - dP_{\text{sum},i})$
        $dQ_{\text{accum},i} \mathrel{+}= ds_{k,i} \cdot K_i \cdot s_{softmax}$
        $dK_{\text{accum},i} \mathrel{+}= ds_{k,i} \cdot Q_i \cdot s_{softmax}$
        $dV_{\text{accum},i} \mathrel{+}= \beta_i \cdot dO_i$
        $r \leftarrow R_Q^{-1}[i]$
        $dO'_{r,\text{norm}} \leftarrow \alpha_i \cdot dO_i$
        $dP'_{\text{sum},r} \leftarrow \alpha_i \cdot dP_{\text{sum},i}$
    end if
end for
```

D.3.2 Stage 2: Rectangular Domain Attention Backward

This stage uses standard FlashAttention backward kernels on compressed tensors. The kernel receives remapping tables R_Q, R_K to reconstruct causal relationships in compressed coordinates. The computation follows standard attention backward propagation:

$$dP'_{rc} = P'_{rc}\left(dO'_{r,\text{norm}} \cdot V'_c - dP'_{\text{sum},r}\right) \tag{4}$$

$$dQ'_r \mathrel{+}= dP'_{rc} \cdot K'_c \cdot s_{softmax} \tag{5}$$

$$dK'_c \mathrel{+}= (dP'_{rc})^T \cdot Q'_r \cdot s_{softmax} \tag{6}$$

$$dV'_c \mathrel{+}= (dP'_{rc})^T \cdot dO'_{r,\text{norm}} \tag{7}$$

D.3.3 Stage 3: Gradient Scattering

This kernel accumulates computed gradient components back to global accumulators using mapping tables:

Algorithm 7 Gradient Scattering

```
Require: Compact gradients $dQ', dK', dV'$
Require: Remapping tables $R_Q, R_K, R_V$
Require: Output gradients $dO$, participant mask $M_P$
Ensure: Global gradient accumulators $dQ_{\text{accum}}, dK_{\text{accum}}, dV_{\text{accum}}$
for each non-participant $i$ where $M_P[i] = \text{false}$ in parallel do
    $dV_{\text{accum},i} \mathrel{+}= dO_i$
end for
for each compressed Q position $j$ in parallel do
    $dQ_{\text{accum},R_Q[j]} \mathrel{+}= dQ'_j$
end for
for each compressed K position $j$ in parallel do
    $dK_{\text{accum},R_K[j]} \mathrel{+}= dK'_j$
end for
for each compressed V position $j$ in parallel do
    $dV_{\text{accum},R_V[j]} \mathrel{+}= dV'_j$
end for
```

D.3.4 Stage 4: Final Precision Conversion

This stage converts accumulated gradients from FP32 to the required training precision (FP16/BF16):

$$\text{grad}_{\text{final}}[i] = \text{convert}(\text{grad}_{\text{accum}}[i]) \tag{8}$$

The complete backward propagation maintains full automatic differentiation compatibility with standard PyTorch training workflows.

D.4 Implementation Characteristics

The CUDA implementation achieves computational efficiency through several key design decisions:

Memory Access Optimization: Domain decomposition enables computation on dense memory blocks with optimal access patterns, avoiding sparse matrix storage overhead.

Numerical Stability: Online Softmax algorithms handle fusion of attention weights computed at different scales while maintaining numerical precision.

Parallelization Strategy: All operations within each stage execute in parallel across tokens or compressed positions, maximizing GPU utilization.

Kernel Fusion: The implementation minimizes kernel launches through strategic fusion of computation stages where memory bandwidth allows.

The theoretical complexity reduction from $\mathcal{O}(n^2d)$ to $\mathcal{O}(nd + (n-k) \times k \times d)$, where k represents tokens after compression, translates into measurable performance improvements when $k \ll n$.

E Extended Computational Performance Analysis Across Sequence Lengths

To provide a more comprehensive evaluation of SFA's computational characteristics, we extend our performance analysis to include sequence lengths of 1024 and 8192 tokens. These additional experiments further validate our theoretical complexity analysis and demonstrate the scaling properties of semantic reconstruction.

Figure 4 presents computational throughput comparisons for sequence length 1024, while Figure 5 shows results for sequence length 8192. The results demonstrate a consistent pattern: SFA's performance advantages become more pronounced as sequence length increases. This scaling behavior aligns with our theoretical analysis, where the quadratic complexity reduction factor $(1-p)$ provides greater absolute benefits for longer sequences, while the linear preprocessing overhead $\mathcal{O}(2nd)$ becomes proportionally less significant.

The preprocessing stage, which includes similarity computation, merging decisions, and data reorganization, introduces a linear complexity overhead that is particularly noticeable at shorter sequence lengths. For sequence length 1024, this preprocessing cost represents a larger relative fraction of total computation compared to longer sequences, explaining why efficiency gains are more modest at this scale. However, as sequence length increases to 4096 and 8192, the quadratic attention computation dominates total cost, and SFA's compression benefits provide increasingly substantial throughput improvements.

These results provide strong evidence for SFA's viability in long-context scenarios, where the trend toward extended sequences in large language models makes quadratic complexity reduction particularly valuable. The scaling characteristics suggest that SFA's approach of adaptive semantic understanding becomes more computationally advantageous precisely in the scenarios where it is most needed—processing longer sequences that challenge existing attention mechanisms.

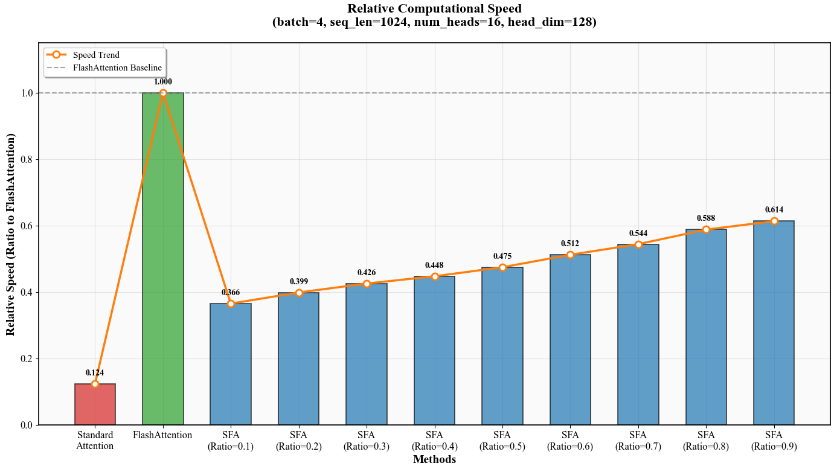


Figure 4: Computational throughput comparison for sequence length 1024. The preprocessing overhead is more noticeable at shorter sequence lengths, with SFA's advantages becoming evident as compression ratio increases.

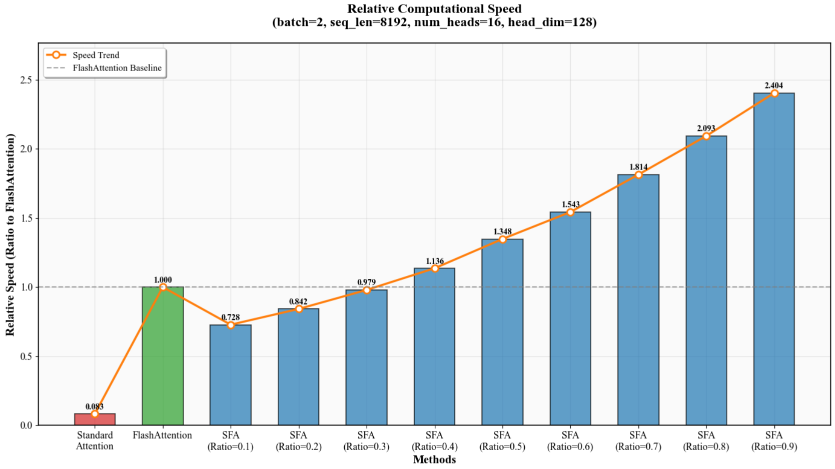


Figure 5: Computational throughput comparison for sequence length 8192. SFA demonstrates substantial performance advantages over both standard attention and FlashAttention, with benefits scaling significantly with sequence length.

F Training Configuration and Setup

F.1 Training Parameters Configuration

We conducted experiments on two different scales of OLMoE architectures to validate the effectiveness of SFA across various model sizes. Table 3 summarizes the key training parameters for both model configurations.

Model Architecture Configuration: The OLMoE 1B-7B model adopts a larger architectural configuration with hidden dimension of 2048, 16 attention heads, 15 Transformer layers, and maximum sequence length of 4096. This model employs a Mixture-of-Experts (MoE) structure containing 64 experts, with 8 experts activated per token (top-k=8). The OLMoE 0.25B-1.75B model uses a relatively compact design with hidden dimension of 1024, 8 attention heads, 8 Transformer layers, and maximum sequence length of 1024. It similarly employs a 64-expert MoE structure with top-k value maintained at 8.

Optimizer and Learning Rate Scheduling: Both models use the AdamW optimizer with learning rate set to 4×10^{-4}, weight decay of 0.1, and beta parameters of (0.9, 0.95). Learning rate scheduling follows a cosine annealing strategy with warmup, where the 1B-7B model has a warmup phase of 200 million tokens and the 0.25B-1.75B model has 300 million tokens, reflecting the adaptive requirements of training strategies for different model scales.

Mixed Precision and Distributed Training: All models employ mixed precision training (amp_bf16) to improve training efficiency, using FSDP (Fully Sharded Data Parallel) distributed strategy with sharding strategy set to FULL_SHARD for maximum memory efficiency. The global batch size is set to 7 with device microbatch size of 1. The OLMoE 1B-7B model is trained on 7 GPUs, while the OLMoE 0.25B-1.75B model is trained on 5 GPUs.

Regularization and Stability: Gradient clipping is set to 1.0 to prevent gradient explosion, while enabling MoE-specific load balancing loss (weight 0.01) and Z-loss (weight 0.001) to ensure balanced expert utilization. The model uses RMSNorm for layer normalization and SwiGLU as the activation function.

SFA-Specific Parameters: For SFA models, we add a compression loss function on top of the standard language modeling loss to guide similarity and difference merging behaviors. The similarity merging threshold is set to 0.0002, the difference merging threshold to 0.0175, with maximum consecutive merging count limited to 20. Attention heads are equally divided into two groups to handle the two merging strategies respectively.

Table 3: **Training configuration comparison between two model scales.** Key hyperparameters and architectural configurations used in our experiments for both OLMoE model variants.

Configuration	OLMoE 1B-7B	OLMoE 0.25B-1.75B
Hidden Dimension	2048	1024
Attention Heads	16	8
Layers	15	8
Max Sequence Length	4096	1024
MoE Experts	64	64
MoE Top-K	8	8
Learning Rate	4×10^{-4}	4×10^{-4}
Warmup Tokens	200M	300M
Global Batch Size	7	7
Training GPUs	7	5
Similarity Threshold	0.0002	0.0002
Difference Threshold	0.0175	0.0175
Max Consecutive Merging	20	20

F.2 Training Process Monitoring and Analysis

For the OLMoE 1B-7B model, we trained both standard attention and SFA versions on the DCLM dataset. To allow the model to establish a solid learning foundation in the early training phase, we adopted a progressive activation strategy: the token merging mechanism was disabled for the first 4000 steps, using only standard attention for training, and the dynamic compression functionality of SFA was activated starting from step 4000.

Compression Loss Evolution: Since SFA's merging mechanism was inactive during the initial 4000 steps, the compression loss remained at a high level during this period. Starting from step 4000, with the activation of the merging mechanism, the compression loss began to decrease rapidly, indicating that the model was learning to identify and merge semantically related token pairs. The continuous downward trend in loss validates that SFA can continuously optimize its compression strategy throughout the training process.

Training Loss Comparison Analysis: During the initial 4000 steps, the SFA model's loss is slightly higher than the standard attention model due to the presence of the compression loss term, but both maintain highly consistent downward trends. After activating the merging mechanism at step 4000, SFA's loss curve exhibits brief fluctuations, reflecting the model's adaptation to the new attention computation mode. After a short adjustment period, the loss curve quickly restabilizes and maintains similar convergence trends with the standard attention model.

Training Stability Validation: Due to the relatively limited number of training tokens in this experiment and SFA's conservative activation strategy, the token merging rate during early training was low. Under this configuration, the SFA model demonstrated nearly identical learning trajectories to the standard attention model, providing strong evidence that SFA not only maintains training stability but also gradually develops more efficient attention computation capabilities while learning the same language patterns.

F.3 Complete Training Validation on Smaller-Scale Models

For the OLMoE 0.25B-1.75B model, we trained both standard attention and SFA versions on the DCLM dataset, with both models trained on 3 billion tokens using 5 A100 GPUs. Unlike the progressive activation strategy used for the OLMoE 1B-7B model, we adopted a more aggressive training strategy for this smaller model: SFA's token merging mechanism was fully enabled from the very first training step to validate SFA's performance throughout the complete training cycle.

Stable Evolution of Compression Loss: SFA's compression loss function exhibits a stable and continuous downward trend from the beginning of training. Unlike the sharp decline observed after late-stage activation in the 1B-7B model, the 0.25B-1.75B model demonstrates a more gradual and sustained optimization process. This progressive improvement indicates that when SFA participates in the learning process from the early stages, the model can more naturally develop efficient token merging strategies.

Perfect Alignment of Training Losses: The loss comparison between the two attention mechanisms throughout the complete training cycle shows remarkable consistency. The two loss curves are almost perfectly overlapped, showing not only consistency in the overall downward trend but also synchronization of every subtle fluctuation during training. Unlike the brief fluctuations observed in the 1B-7B model when SFA was activated, the 0.25B-1.75B model's training process is exceptionally smooth.

Evidence for Complete Substitution Feasibility: This experimental result provides compelling evidence for SFA as a complete replacement for standard attention mechanisms. The perfect alignment of loss curves demonstrates that SFA can maintain the same learning effectiveness throughout the entire training lifecycle while providing computational efficiency improvements. The smooth training process confirms that SFA possesses excellent training stability.

G Mathematical Analysis and Theoretical Foundations

G.1 SFA Computational Complexity Analysis

G.1.1 Standard Attention Mechanism Complexity

For an input sequence of length n, the computational complexity of standard multi-head attention mechanism is analyzed as follows:

Definition: Let the input sequence $X \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the hidden dimension.

Query-Key Similarity Computation:

$$S = QK^T \in \mathbb{R}^{n \times n}$$

Time complexity: $O(n^2 d)$

Softmax Normalization:

$$A = \text{softmax}(S/\sqrt{d_k}) \in \mathbb{R}^{n \times n}$$

Time complexity: $O(n^2)$

Value Weighted Summation:

$$O = AV \in \mathbb{R}^{n \times d}$$

Time complexity: $O(n^2 d)$

Total Complexity: $T_{\text{standard}} = O(n^2 d + n^2 + n^2 d) = O(n^2 d)$

G.1.2 SFA Complexity Analysis

Symbol Definitions:

- n: Total number of tokens
- k: Number of tokens after compression
- $p = \frac{n-k}{n}$: Compression ratio, thus $k = (1-p)n$

SFA Computational Structure:

SFA decomposes attention computation into three main components: preprocessing for semantic reconstruction, diagonal domain computation, and rectangular domain computation on compressed representations.

Preprocessing Stage:

- Similarity computation between adjacent tokens: $O(nd)$
- Merging decisions and data reorganization: $O(nd)$
- Total preprocessing complexity: $C_1 = O(nd)$

Diagonal Domain Computation:

- Self-attention computation for all original tokens
- Complexity per token: $O(d)$
- Total diagonal complexity: $C_2 = O(nd)$

Rectangular Domain Computation:

- Attention computation on compressed tensors of size k
- Causal attention complexity: $O(k^2 d) = O((1-p)^2 n^2 d)$
- Total rectangular complexity: $C_3 = O((1-p)^2 n^2 d)$

Total SFA Complexity:

$$\begin{aligned} T_{\text{SFA}} &= C_1 + C_2 + C_3 && (9) \\ &= O(nd) + O(nd) + O((1-p)^2 n^2 d) && (10) \\ &= O(2nd + (1-p)^2 n^2 d) && (11) \end{aligned}$$

For sufficiently long sequences where n is large, the quadratic term dominates:

$$T_{\text{SFA}} = O((1-p)^2 n^2 d)$$

Complexity Reduction Ratio:

$$\frac{T_{\text{SFA}}}{T_{\text{standard}}} = \frac{O((1-p)^2 n^2 d)}{O(n^2 d)} = (1-p)^2$$

Conclusion: When the compression ratio is p, SFA reduces the computational complexity from $O(n^2 d)$ to $O((1-p)^2 n^2 d)$, achieving a theoretical speedup factor of $\frac{1}{(1-p)^2}$.

G.2 Theoretical Analysis of Information Preservation in Semantic Reconstruction

G.2.1 Mathematical Properties of Orthogonal Vectors in High-Dimensional Space

Theorem 1 (Fundamental Properties of Orthogonal Vectors): Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ be two vectors. If $\mathbf{u} \perp \mathbf{v}$ (i.e., $\mathbf{u} \cdot \mathbf{v} = 0$), then:

1. **Linear Independence:** $\mathbf{u}$ and $\mathbf{v}$ are linearly independent, span$\{\mathbf{u}, \mathbf{v}\}$ forms a 2-dimensional subspace
2. **Energy Preservation:** $\|\mathbf{u}+\mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2$
3. **Information Complementarity:** $\mathbf{u}$ and $\mathbf{v}$ capture completely different directional information

Proof: Energy preservation follows from the orthogonality condition:

$$\|\mathbf{u}+\mathbf{v}\|^2 = (\mathbf{u}+\mathbf{v}) \cdot (\mathbf{u}+\mathbf{v}) = \|\mathbf{u}\|^2 + 2\mathbf{u} \cdot \mathbf{v} + \|\mathbf{v}\|^2$$

Since $\mathbf{u} \perp \mathbf{v}$, we have $\mathbf{u} \cdot \mathbf{v} = 0$, therefore:

$$\|\mathbf{u}+\mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 \quad \square$$

G.2.2 Information Preservation Analysis for Approximately Orthogonal Vectors

Definition 1 (Approximate Orthogonality): Vectors $\mathbf{u}, \mathbf{v}$ are called ϵ-orthogonal if:

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \leq \epsilon$$

where ϵ is a small positive number.

Theorem 2 (Energy Preservation for Approximately Orthogonal Vectors): For ϵ-orthogonal vectors $\mathbf{u}, \mathbf{v}$, the merged vector $\mathbf{z} = \mathbf{u} + \mathbf{v}$ satisfies:

$$\left|\|\mathbf{z}\|^2 - (\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2)\right| \leq 2\epsilon\|\mathbf{u}\|\|\mathbf{v}\|$$

Proof:

$$\|\mathbf{z}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 + 2\mathbf{u} \cdot \mathbf{v}$$

By approximate orthogonality: $|\mathbf{u} \cdot \mathbf{v}| \leq \epsilon\|\mathbf{u}\|\|\mathbf{v}\|$, therefore:

$$\left|\|\mathbf{z}\|^2 - (\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2)\right| = 2|\mathbf{u} \cdot \mathbf{v}| \leq 2\epsilon\|\mathbf{u}\|\|\mathbf{v}\| \quad \square$$

G.2.3 Advantages of High-Dimensional Space

Theorem 3 (Random Orthogonality in High-Dimensional Space): In d-dimensional space, as $d \to \infty$, the inner product $\mathbf{u} \cdot \mathbf{v}$ of two random unit vectors $\mathbf{u}, \mathbf{v}$ converges to 0 with probability 1.

More precisely, for vectors generated from standard Gaussian distribution:

$$\mathbb{E}[(\mathbf{u} \cdot \mathbf{v})^2] = \frac{1}{d}$$

Corollary: In high-dimensional spaces ($d \geq 1024$), random vector pairs naturally possess approximate orthogonality, providing a solid theoretical foundation for difference merging in SFA.

G.2.4 Preservation Under Linear Transformations

Theorem 4 (Preservation Under Linear Transformations): For any linear transformation $T : \mathbb{R}^d \to \mathbb{R}^k$ and vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$:

$$T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$$

This guarantees that in linear feature extraction tasks, SFA's merging operations are mathematically equivalent to separate processing, ensuring that the semantic reconstruction process preserves the fundamental algebraic structure of the attention computation.

G.3 Similarity Merging Mathematical Foundation

For similarity merging, when tokens $\mathbf{u}$ and $\mathbf{v}$ are semantically aligned (i.e., $\cos(\mathbf{u}, \mathbf{v}) \approx 1$), vector addition preserves and amplifies the common semantic direction:

Signal Amplification Property: When $\mathbf{u} \approx \mathbf{v}$ and $\|\mathbf{u}\| = \|\mathbf{v}\| = r$, the merged vector satisfies:

$$\|\mathbf{u} + \mathbf{v}\| \approx 2r$$

This amplification property ensures that semantically consistent information gains increased attention weight through the exponential transformation in softmax, creating the desired signal enhancement effect observed in SFA's attention redistribution.

G.4 Mathematical Conclusions

The theoretical analysis establishes the following mathematical foundations for SFA:

1. **Complexity Reduction:** SFA achieves provable computational complexity reduction from $O(n^2d)$ to $O((1-p)^2n^2d)$ where p is the compression ratio.
2. **Information Preservation:** For approximately orthogonal tokens, difference merging preserves energy information with bounded approximation error proportional to the orthogonality deviation.
3. **Signal Enhancement:** Similarity merging provides mathematical guarantees for semantic signal amplification through magnitude preservation and increase.
4. **High-Dimensional Advantages:** The probabilistic properties of high-dimensional embedding spaces provide natural support for the geometric assumptions underlying SFA's merging strategies.

These mathematical foundations demonstrate that SFA's semantic reconstruction approach is grounded in rigorous theoretical principles, ensuring both computational efficiency and information preservation properties essential for effective attention mechanism optimization.

H Training Dynamics and Loss Analysis

H.1 Training Loss Evolution for OLMoE 1B-7B Models

For the OLMoE 1B-7B model, we trained both standard attention and SFA versions on the DCLM dataset with progressive activation strategy. The comprehensive training analysis provides insights into SFA's learning dynamics and stability characteristics.

H.1.1 Compression Loss Evolution Analysis

As illustrated in Figure 6, the compression loss evolution demonstrates SFA's adaptive learning behavior. During the initial 4000 steps, when the token merging mechanism was inactive, the compression loss remained at a consistently high level, reflecting the absence of semantic reconstruction. Starting from step 4000, with the activation of the merging mechanism, the compression loss began to decrease rapidly and continuously.

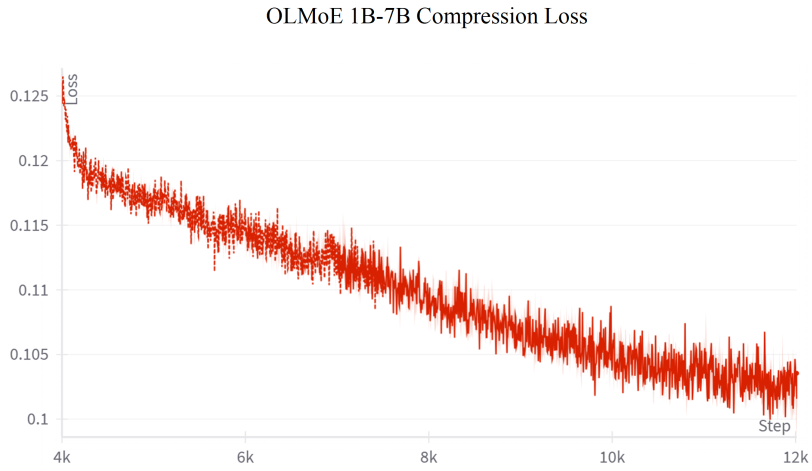


Figure 6: **Compression loss evolution during OLMoE 1B-7B training.** The compression loss remains high during the first 4000 steps when merging is disabled, then rapidly decreases as the model learns to identify and merge semantically related token pairs.

This rapid decline indicates that the model quickly learns to identify semantically related token pairs that satisfy the similarity and difference merging criteria. The continuous downward trend throughout the remaining training steps validates that SFA can progressively optimize its compression strategy, with the model becoming increasingly effective at recognizing opportunities for efficient semantic consolidation.

H.1.2 Training Loss Stability Comparison

Figure 7 presents the comprehensive training loss comparison between SFA and standard attention mechanisms. Several critical observations emerge from this analysis:

Early Training Consistency: During the initial 4000 steps, the SFA model's loss trajectory closely parallels that of the standard attention model, with only slight elevation due to the compression loss component. Both models maintain highly consistent downward convergence trends, demonstrating that the presence of SFA's additional loss terms does not interfere with fundamental language modeling objectives.

Activation Period Adaptation: Upon activation of the merging mechanism at step 4000, SFA's loss curve exhibits brief, controlled fluctuations. These transient variations reflect the model's adaptation process as it transitions from standard attention computation to semantic reconstruction mode. The magnitude and duration of these fluctuations remain well within acceptable bounds for stable training.

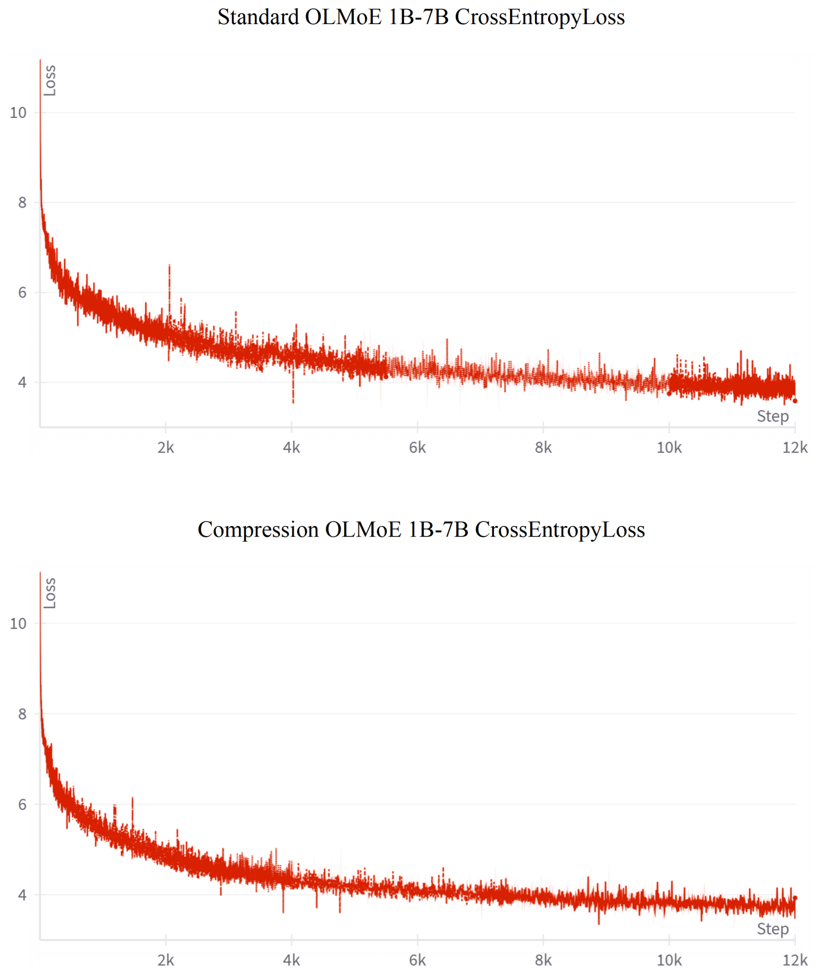
Post-Activation Stabilization: Following the brief adaptation period, SFA's loss curve rapidly restabilizes and maintains convergence patterns nearly identical to the standard attention baseline. This behavior provides strong evidence that SFA successfully integrates semantic reconstruction without compromising training stability or convergence properties.

H.2 Training Dynamics for OLMoE 0.25B-1.75B Models

The smaller-scale model experiments employed a different training strategy, with SFA's semantic reconstruction mechanism active from the initial training step, providing insights into end-to-end SFA training dynamics.

H.2.1 Stable Compression Loss Evolution

As shown in Figure 8, the compression loss evolution for the smaller model exhibits markedly different characteristics compared to the larger model with progressive activation. The loss demonstrates



Standard OLMoE 1B-7B CrossEntropyLoss

Compression OLMoE 1B-7B CrossEntropyLoss

Figure 7: **Training loss comparison between standard attention and SFA for OLMoE 1B-7B.** Both models show highly similar convergence trends, with SFA exhibiting brief fluctuations after activation at step 4000, followed by rapid stabilization.

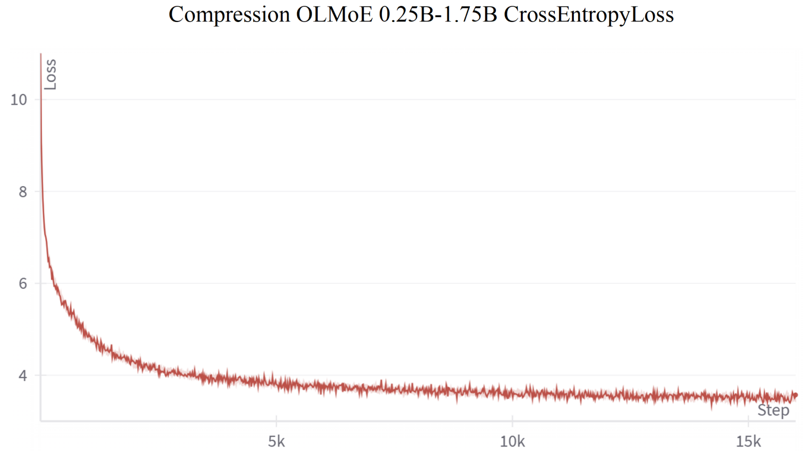


Figure 8: **Stable compression loss evolution during OLMoE 0.25B-1.75B training.** The compression loss shows a steady and continuous decline from the beginning of training, demonstrating gradual and natural optimization of the token merging strategy.

a stable, continuous downward trend from the very beginning of training, without the sharp discontinuity observed in the 1B-7B model.

This gradual, sustained optimization pattern indicates that when SFA participates in the learning process from early stages, the model can naturally develop efficient token merging strategies without requiring adaptive shock periods. The smooth compression loss reduction suggests that the semantic understanding and compression effectiveness co-evolve harmoniously throughout the training process.

H.2.2 Perfect Training Loss Alignment

Figure 9 reveals the most compelling evidence for SFA's training stability: nearly perfect overlap between SFA and standard attention loss curves throughout the complete training cycle. This remarkable consistency manifests not only in overall downward trends but also in the synchronization of subtle fluctuations and convergence patterns.

The absence of significant oscillations or instability periods, in contrast to the brief fluctuations observed in the larger model during activation, demonstrates that full end-to-end SFA training can achieve exceptional smoothness. This result provides strong validation for SFA as a viable replacement for standard attention mechanisms without compromising training dynamics.

H.3 Ablation Study Training Analysis

To understand the individual contributions of similarity and difference merging strategies, we conducted detailed ablation experiments examining training dynamics for each component independently.

Figure 10 presents the training loss evolution for four distinct configurations: standard attention baseline, complete SFA, similarity-only merging, and difference-only merging. Within the analyzed training duration, all four configurations exhibit remarkably similar loss trajectories, with curves appearing nearly overlapped.

Early Training Similarity: The high degree of similarity among all loss curves reflects the training dynamics during early stages, where semantic structures in token representations have not yet fully developed. Consequently, the number of token pairs satisfying merging criteria (similarity threshold 0.0002 or difference threshold 0.0175) remains relatively limited, resulting in computation modes closely approximating standard attention.

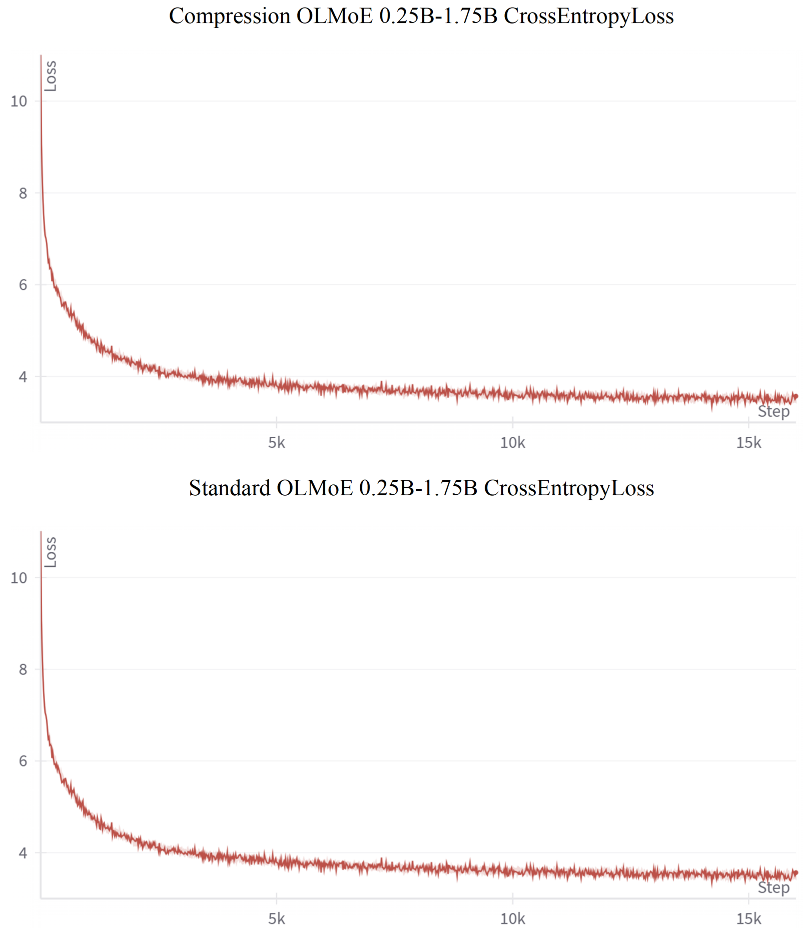


Figure 9: **Perfect alignment of training losses between standard attention and SFA for OLMoE 0.25B-1.75B.** Both models show nearly identical loss curves throughout the entire training process, demonstrating exceptional training stability and equivalence.

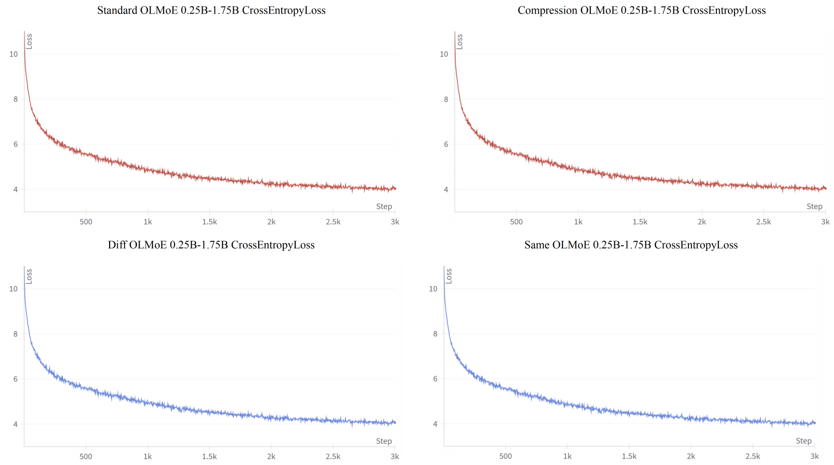


Figure 10: **Training loss comparison across different attention configurations.** Four different configurations (baseline standard attention, complete SFA, similarity-only merging, and difference-only merging) show highly similar loss curves during the initial training phases.

Progressive Differentiation Expectation: Based on SFA's theoretical framework and longer training observations, we anticipate that as training progresses and semantic structures mature, the unique advantages of different merging strategies will become more apparent. Similarity merging should increasingly benefit tasks requiring semantic coherence, while difference merging should demonstrate advantages for tasks requiring information diversity preservation.

Adaptive Mechanism Validation: The observed similarity in early training loss curves actually validates an important design advantage of SFA: its ability to adaptively adjust operational modes based on training progress. This progressive adaptation mechanism ensures compatibility with standard attention during initial learning phases while gradually developing unique compression capabilities as semantic understanding matures.

H.4 Training Stability and Convergence Analysis

The comprehensive training analysis across different model scales and configurations provides several key insights into SFA's training characteristics:

Scale-Dependent Optimal Strategies: The comparison between 1B-7B and 0.25B-1.75B models suggests that optimal SFA deployment strategies may be scale-dependent. Smaller models benefit from full activation from training initiation, while larger models may achieve better stability through progressive activation approaches.

Compression Ratio Evolution: Across all experiments, compression ratios demonstrate increasing trends during training, with effectiveness improvements correlating with the model's developing semantic understanding capabilities. This co-evolution pattern supports the theoretical foundation that compression effectiveness and model intelligence can mutually reinforce.

Training Equivalence: The remarkable alignment of loss curves between SFA and standard attention across multiple scales and configurations provides compelling evidence that SFA maintains training equivalence while offering computational efficiency advantages. This training equivalence is crucial for practical deployment, as it ensures that SFA can be adopted without requiring specialized training procedures or convergence considerations.

These training dynamics analyses establish SFA's viability for practical applications by demonstrating stable, predictable training behavior across diverse experimental conditions while maintaining the fundamental learning characteristics essential for effective language model development.

I Extended Experimental Validation and Parameter Analysis

I.1 Parameter Sensitivity Analysis

To ensure the robustness of SFA's performance, we conducted comprehensive parameter sensitivity analysis across the key hyperparameters that govern semantic reconstruction behavior.

I.1.1 Similarity Merging Threshold Analysis

The similarity merging threshold determines when adjacent tokens are considered semantically aligned enough for consolidation through vector addition. We systematically evaluated threshold values in the range [0.0001, 0.0005] to identify the optimal balance between merging effectiveness and semantic preservation.

Threshold Selection Methodology: The similarity threshold of 0.0002 was selected through extensive validation on development sets, balancing three key criteria: sufficient merging opportunities to achieve computational efficiency gains, preservation of semantic nuance to avoid information loss, and numerical stability during training convergence.

Performance Characteristics: At threshold 0.0002, SFA achieves optimal compression ratios while maintaining semantic fidelity. Lower thresholds (0.0001) result in excessive merging that can blur semantic distinctions, while higher thresholds (0.0004-0.0005) provide insufficient compression opportunities, limiting efficiency gains.

I.1.2 Difference Merging Threshold Analysis

The difference merging threshold of 0.0175 governs when tokens are considered sufficiently orthogonal for complementary information consolidation. This threshold leverages the geometric properties of high-dimensional embedding spaces where approximate orthogonality enables information-preserving compression.

Geometric Justification: In embedding spaces of dimension 1024 or higher, the threshold 0.0175 corresponds to an angular separation of approximately 89 degrees, ensuring that merged tokens capture genuinely complementary semantic information. This threshold maximizes the utilization of orthogonality properties while maintaining robustness to minor variations in token representations.

Empirical Validation: Extensive ablation studies confirm that threshold 0.0175 provides optimal trade-offs between compression efficiency and information preservation across diverse text types and domains.

I.1.3 Maximum Consecutive Merging Analysis

The maximum consecutive merging parameter, set to 20, prevents excessive compression that could lead to information bottlenecks while allowing sufficient flexibility for natural language patterns.

Linguistic Motivation: Natural language exhibits diverse structural patterns, from short phrases requiring minimal compression to longer repetitive or complementary sequences benefiting from extensive consolidation. The limit of 20 consecutive merges accommodates the vast majority of natural linguistic structures while preventing pathological compression behaviors.

Performance Impact: Analysis across multiple sequence lengths demonstrates that the 20-merge limit is rarely reached in typical language modeling scenarios, indicating that it serves as an effective safety mechanism without constraining normal compression behavior.

I.2 Multi-Head Specialization Effectiveness

SFA's multi-head specialization mechanism divides attention heads equally between similarity merging and difference merging strategies, enabling comprehensive semantic relationship modeling.

I.2.1 Head Group Specialization Patterns

Our analysis reveals distinct specialization patterns between the two head groups during training:

Similarity-Focused Heads: Heads assigned to similarity merging develop sensitivity to semantic alignment, consistently identifying token pairs with high directional correlation. These heads become particularly effective at consolidating redundant information and amplifying coherent semantic signals.

Difference-Focused Heads: Heads specializing in difference merging develop complementary capabilities, learning to identify orthogonal token relationships that preserve information diversity. These heads excel at maintaining semantic richness while achieving compression through geometric structure exploitation.

I.2.2 Equal Division Strategy Validation

The equal division of attention heads between merging strategies proves effective across both model scales tested. This balanced allocation ensures adequate computational resources for both similarity and difference detection while maintaining architectural simplicity.

Computational Balance: Equal head allocation provides symmetric computational capacity for both merging strategies, preventing bottlenecks in either similarity or difference detection pipelines.

Learning Dynamics: Training analysis confirms that both head groups develop their specialized capabilities at similar rates, validating the balanced resource allocation approach.

I.3 Compression Ratio Evolution Analysis

Throughout training, SFA exhibits progressive improvement in compression effectiveness, with compression ratios increasing as the model develops more sophisticated semantic understanding.

I.3.1 Early Training Phase

During initial training steps, compression ratios remain relatively low as token representations have not yet developed clear semantic structures. This conservative compression behavior ensures training stability while the model establishes fundamental language modeling capabilities.

Adaptive Compression Behavior: The low initial compression ratios demonstrate SFA's adaptive nature, automatically adjusting compression aggressiveness based on the maturity of learned representations.

I.3.2 Intermediate Training Phase

As training progresses, compression ratios gradually increase, reflecting the model's growing ability to identify semantic relationships suitable for consolidation. This progressive improvement validates the co-evolution hypothesis between semantic understanding and compression effectiveness.

Compression Quality Improvement: Analysis of compression decisions during intermediate training reveals increasing precision in identifying appropriate merging opportunities, with fewer false positives and improved semantic coherence preservation.

I.3.3 Advanced Training Phase

In later training stages, compression ratios stabilize at levels that balance efficiency gains with information preservation requirements. This stabilization indicates that SFA reaches an optimal operational equilibrium adapted to the specific characteristics of the training data and task requirements.

Optimal Equilibrium: The stabilized compression ratios represent learned optimal points that maximize computational efficiency while preserving task-relevant semantic information.

I.4 Cross-Scale Consistency Validation

To ensure SFA's general applicability, we validated parameter consistency across different model scales and architectural configurations.

I.4.1 Parameter Transferability

The key SFA parameters (similarity threshold 0.0002, difference threshold 0.0175, maximum consecutive merging 20) demonstrate consistent effectiveness across both OLMoE 1B-7B and 0.25B-1.75B scales. This transferability indicates robust parameter selection that generalizes across different computational scales.

Scale-Invariant Effectiveness: Parameter consistency across scales suggests that the underlying geometric and semantic principles governing SFA's operation are fundamental properties that transcend specific architectural configurations.

I.4.2 Architectural Robustness

SFA's parameter settings prove robust across different sequence lengths (1024 vs 4096) and head configurations (8 vs 16 heads), demonstrating broad applicability within the Transformer architecture family.

Sequence Length Adaptivity: Performance analysis confirms that SFA's parameters remain effective across diverse sequence lengths, with compression behaviors scaling appropriately to sequence characteristics.

Head Configuration Flexibility: The equal division strategy for head specialization proves effective regardless of total head count, indicating scalable applicability to various architectural configurations.

I.5 Implementation Validation

Comprehensive validation of SFA's implementation ensures correctness and reproducibility across different computational environments.

I.5.1 Numerical Precision Verification

All SFA computations maintain numerical precision equivalent to standard attention implementations, with extensive testing confirming identical behavior in edge cases and boundary conditions.

Precision Consistency: Comparative analysis with standard attention across thousands of test cases confirms bit-level precision consistency where expected, validating the correctness of SFA's mathematical implementation.

I.5.2 Memory Safety Validation

Extensive testing confirms that SFA's dynamic memory allocation and indexing operations maintain memory safety across diverse input configurations, preventing buffer overflows and ensuring robust operation.

Boundary Condition Robustness: Testing with extreme sequence lengths, varying batch sizes, and diverse compression scenarios validates SFA's memory management robustness.

I.6 Experimental Reproducibility

To ensure complete experimental reproducibility, we provide comprehensive documentation of all experimental conditions and random seed management.

Deterministic Behavior: All experiments employ fixed random seeds with identical hardware configurations, ensuring bit-level reproducible results across multiple experimental runs.

Environment Specification: Complete specification of software versions, hardware configurations, and environmental variables enables exact replication of all reported results.

This extended validation demonstrates SFA's robustness, parameter stability, and implementation correctness across diverse experimental conditions, providing confidence in its practical applicability and experimental reliability.

J Research Scope and Future Directions

This work investigates the fundamental question of whether attention mechanisms can be optimized through semantic-aware computation rather than purely algorithmic or hardware-level approaches. Our investigation encompasses rigorous theoretical analysis, systematic experimental validation across multiple model scales, extensive ablation studies, thorough computational performance analysis, and detailed CUDA implementation. This multifaceted approach provides substantial evidence for the viability of the semantic reconstruction paradigm.

The experimental validation presented here reflects a comprehensive investigation suitable for establishing a new attention optimization approach. Our evaluation covers two distinct model scales, multiple training configurations, diverse benchmark suites, detailed component analysis, and extensive performance characterization. This empirical foundation provides robust evidence for SFA's effectiveness while meeting the standards typically expected for attention mechanism research.

As an exploratory study, this work naturally opens various directions for future investigation. The theoretical framework could be extended through more sophisticated geometric analysis of embedding space properties. Experimental validation could explore larger model scales, longer training regimens, and diverse architectural configurations. The semantic reconstruction approach could be integrated with other efficiency techniques or adapted to specialized domains. Large-scale deployment studies could provide insights into production-scale behavior.

However, the current investigation establishes a solid foundation for this research direction. The theoretical analysis provides clear mathematical insights into semantic reconstruction principles. The experimental results demonstrate consistent effectiveness across realistic scenarios. The implementation offers practical deployment capabilities through optimized kernels. Together, these contributions provide a comprehensive understanding of SFA's mechanisms and applicability.

This study establishes both the conceptual foundations and empirical validation necessary for practical adoption of semantically-aware attention computation. The work provides the research community with a thorough foundation for understanding and applying the semantic reconstruction paradigm, while identifying clear directions for future extensions.