Learning When Not to Attend Globally

Anonymous ACL submission

Abstract

When reading books, humans focus primarily on the current page, flipping back to recap prior context only when necessary. Similarly, we demonstrate that Large Language Models (LLMs) can learn to dynamically determine when to attend to global context. We propose All-or-Here Attention (AHA), which utilizes a binary router per attention head to dynamically toggle between full attention and local sliding window attention for each token. Our results indicate that with a window size of 256 tokens, up to 93% of the original full attention operations can be replaced by sliding window attention without performance loss. Furthermore, by evaluating AHA across various window sizes, we identify a long-tail distribution in context dependency, where the necessity for full attention decays rapidly as the local window expands. By decoupling local processing from global access, AHA reveals that full attention is largely redundant, and that efficient inference requires only on-demand access to the global context. Our code and model will be made publicly available.

1 Introduction

The self-attention mechanism (Bahdanau et al., 2014) serves as the cornerstone of the Transformer architecture (Devlin et al., 2019; Brown et al., 2020; Vaswani et al., 2017), driving the success of modern Large Language Models (Ouyang et al., 2022; Touvron et al., 2023; Yang et al., 2025). However, its quadratic complexity with respect to sequence length poses significant computational challenges. To address this, numerous studies have proposed efficiency techniques, including linear attention (Katharopoulos et al., 2020; Gu and Dao, 2024; Yang et al., 2024), sparse attention (Child et al., 2019a; Beltagy et al., 2020a; Zaheer et al., 2020b), and hierarchical attention (Yang et al., 2016; Yuan et al., 2025). While these methods often focus on approximating full attention or compressing state representations, we suggest that the intrinsic redundancy of the attention mechanism enables a far simpler solution.

In this paper, we propose a straightforward paradigm: enabling the Transformer to *learn when not to attend globally*. Our design is motivated by the observation that tokens exhibit heterogeneous context requirements. Intuitively, maintaining syntactic continuity and local coherence should primarily require immediate context, whereas global retrieval is necessary only for a sparse subset of tokens resolving long-range dependencies (Liu et al., 2017; Fan and Jiang, 2023; Futrell et al., 2015; Beltagy et al., 2020a). Based on this perspective, we introduce **All-or-Here Attention (AHA)**. By employing a conditional hard gating mechanism, AHA dynamically toggles between full attention ("All") and local sliding window attention ("Here") for each attention head.

Specifically, we integrate lightweight routers at each Transformer layer to generate scalar importance scores. Based on these scores and a predefined threshold, each attention head determines whether to execute full or sliding window attention. This represents a fundamental departure from methods such as DuoAttention (Xiao et al., 2025) and Prulong (Bhaskar et al., 2025), which rely on a static pre-classification of attention heads into either streaming heads (sliding window attention) or retrieval heads (full attention). While those approaches fix the functional roles of heads regardless of the input, AHA dynamically toggles the attention scope of each head at the token level based on the real-time context. As illustrated in Figure 1, the router automatically calibrates its global access according to the complexity of the information retrieval needed: it learns to use full attention sparingly at 17% for tasks like sequential counting that focus on local patterns, while it automatically increases global access to 39% for tasks that require

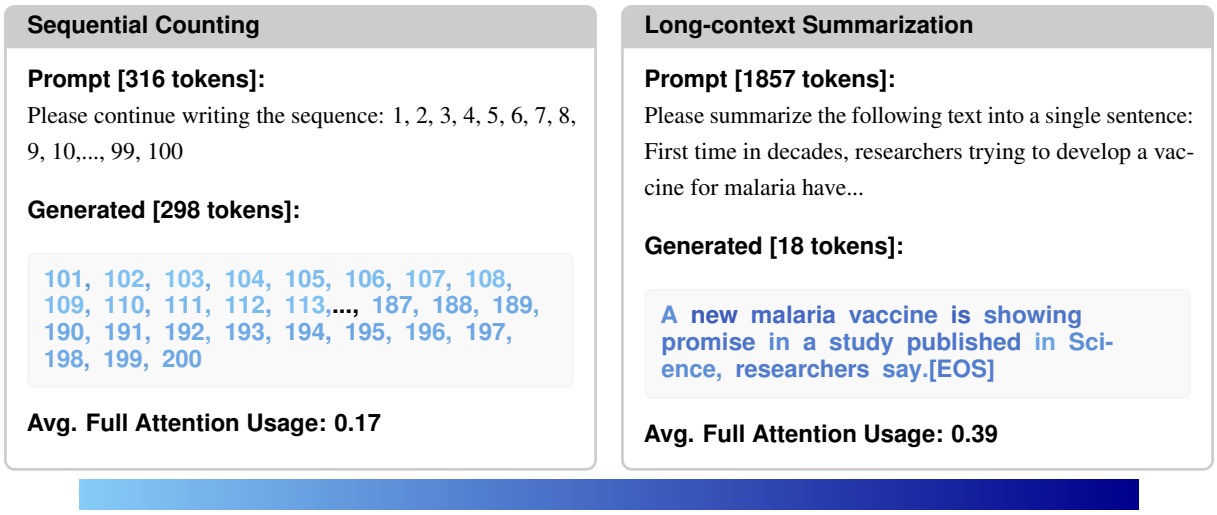

Figure 1: Visualization of average full attention usage across tasks (local attention window size, $w = 128$). The color gradient indicates the ratio of activating full attention, where dark blue represents 1.0 (Full Attention) and light blue represents 0.0 (Local Attention). This shows that reliance on global context varies by task and token.

information from across the entire document, such as long-context summarization.

Experiments demonstrate that the vast majority of full attention operations in pre-trained LLMs are not needed. Using OLMo-2 (OLMo et al., 2024) as our base model, we show that with a window size of 256, over 93% of full attention operations can be substituted with local sliding window attention while maintaining full performance on standard benchmarks. We further analyze the relationship between window size and attention sparsity, revealing that context dependency follows a long-tail distribution. As the local window expands, the necessity for global access decays rapidly, dropping from 53% at a window size of 16 to under 7% at a window size of 256. These findings validate our hypothesis that full attention is largely unnecessary. By explicitly decoupling local processing from global retrieval, AHA offers new insights for efficient large language models. To facilitate future research, we will release our model weights, training scripts, and evaluation code.

2 Method

The core intuition behind AHA is to assign a binary gate to each attention head, dynamically routing tokens to either a full ('all') or a local sliding window ('here') attention path. Although conditional computation (Bengio et al., 2015; Bapna et al., 2020) and its application to attention mechanisms (Ainslie et al., 2023; Zhang et al., 2022) have been discussed in prior works, here we investigate the implications of enabling such dynamic sparsity within pre-trained decoder-only language models to characterize their intrinsic attention requirements across varying contexts.

2.1 Architecture

Figure 2 illustrates the pipeline of a single All-or-here Attention Block. The core structure is a dynamic routing mechanism that adaptively toggles between full and local attention.

Let the input hidden states be $X \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the model dimension. Each Transformer block comprises m attention heads, denoted by the set $H = \{\text{head}_1, \ldots, \text{head}_m\}$. To achieve input-adaptive routing, we employ a lightweight router $W_{\text{Router}} \in \mathbb{R}^{d \times m}$ that generates importance scores dynamically. The resulting score matrix $S(X) \in \mathbb{R}^{n \times m}$ is conditioned on the input X and computed as:

$$S(X) = \sigma(XW_{\text{Router}}). \quad (1)$$

where the σ represents the sigmoid activation. This formulation ensures that the attention scope for each token is not statically assigned but rather determined by the specific contextual information in X. This represents a major departure from methods like DuoAttention and Prulong, which rely on fixed, pre-defined functional roles for each head.

The element $s_{i,j} \in (0, 1)$ of the matrix $S(X)$ represents the importance score for the i-th token within the j-th attention head. This score $s_{i,j}$ governs the binary routing decision for each specific token-head pair via a thresholding mechanism:

$$g_{i,j} = \mathbb{I}(s_{i,j} > \tau), \quad (2)$$

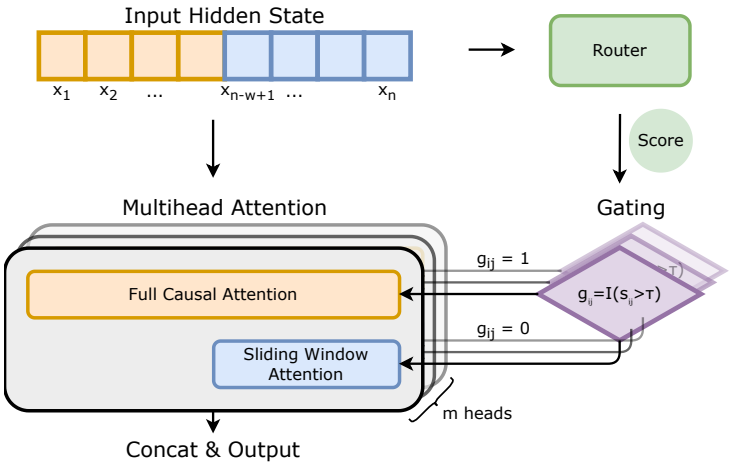

Figure 2: Overview of the All-or-Here Attention (AHA) architecture. A lightweight router computes importance scores for each head, generating binary gates that dynamically toggle between full and local attention.

where $\mathbb{I}(\cdot)$ is the indicator function and τ is a pre-defined threshold. The binary gate $g_{i,j}$ determines the attention granularity: $g_{i,j} = 1$ triggers full causal attention, whereas $g_{i,j} = 0$ restricts the computation to local sliding window attention. To optimize the non-differentiable indicator function, we employ the Straight-Through Estimator (STE) (Bengio et al., 2013). While the discrete gate $g_{i,j}$ determines the attention path during the forward pass, the backward pass approximates the thresholding as an identity function. This bypasses the zero-gradient bottleneck of discrete operations, enabling gradient propagation through the hard gating mechanism.

With the binary gate $g_{i,j}$ determined, the model conditionally executes the attention mechanism. Let w denote the pre-defined size of the local sliding window. When the gate is activated ($g_{i,j} = 1$), the attention output $\mathbf{a}_{i,j}$ for the j-th head at token step i is computed using the entire prefix (Full Attention):

$$\mathbf{a}_{i,j} = g_{i,j} \cdot \text{Attn}(\mathbf{q}_{i,j}, \mathbf{K}_{:i,j}, \mathbf{V}_{:i,j}). \quad (3)$$

Conversely, when $g_{i,j} = 0$, the model restricts its context to a local sliding window of the most recent w tokens:

$$\mathbf{a}_{i,j} = (1 - g_{i,j}) \cdot \text{Attn}(\mathbf{q}_{i,j}, \mathbf{K}_{i-w:i,j}, \mathbf{V}_{i-w:i,j}), \quad (4)$$

where $\mathbf{q}_{i,j}$ is the query vector for the current token. The notation $\mathbf{K}_{:i,j}$ represents the key states for the entire prefix, whereas $\mathbf{K}_{i-w:i,j}$ denotes the truncated key states restricted to the local window.

Following the conditional computation, we aggregate the results to form the final block output. For the current token i, the outputs from all m heads are concatenated and projected via the output weight matrix $\mathbf{W}_O \in \mathbb{R}^{d \times d}$:

$$\mathbf{O}_i = \text{Concat}(\mathbf{a}_{i,1}, \ldots, \mathbf{a}_{i,m})\mathbf{W}_O. \quad (5)$$

Beyond the specific modifications to the attention mechanism, our architecture strictly adheres to the standard decoder-only Transformer design (Brown et al., 2020; Touvron et al., 2023). Crucially, our method does not require pre-training from scratch. A standard, fully pre-trained Transformer can be seamlessly adapted by replacing its full attention modules with AHA, followed by continued Supervised Fine-Tuning (SFT). This capability allows AHA to serve as an drop-in enhancement for existing Large Language Models without incurring the prohibitive costs of full-scale pre-training.

2.2 Loss Function

To balance language modeling performance with attention sparsity, we optimize the model using a joint objective function:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \mathcal{L}_{\text{reg}}, \quad (6)$$

where $\mathcal{L}_{\text{LM}}$ is the standard next-token cross-entropy loss and λ controls the sparsity trade-off. Specifically, $\mathcal{L}_{\text{reg}}$ applies an L_1 penalty to the router scores across all L layers, n tokens, and m heads:

$$\mathcal{L}_{\text{reg}} = \frac{1}{L \cdot n \cdot m} \sum_{k=1}^{L} \sum_{i=1}^{n} \sum_{j=1}^{m} s_{i,j}^{(k)}. \quad (7)$$

Minimizing $\mathcal{L}_{\text{reg}}$ drives the scores $s_{i,j}^{(k)}$ toward zero, thereby lowering the frequency of triggering full

Method	MMLU	HellaSwag	CSQA	GSM8K	MBPP	News	Retain %
Vanilla	0.4056	0.6487	0.5307	0.3874	0.1560	0.2074	100.0%
$w = 16$	0.3676	0.6400	0.4578	0.3980	0.1260	0.2014	92.7%
$w = 32$	0.3847	0.6410	0.4586	0.4102	0.1240	0.1814	92.2%
$w = 64$	0.3963	0.6431	0.4980	0.4121	0.1200	0.1983	94.9%
$w = 128$	0.4087	0.6445	0.4996	0.4291	0.1580	0.2053	100.9%
$w = 256$	0.4134	0.6520	0.5053	0.4632	0.1600	0.1975	102.5%

Table 1: Performance comparison of the proposed All-or-Here Attention across various window sizes. "Vanilla" denotes the OLMo-2-0425-1B-SFT model using full attention.

Method	MMLU	HellaSwag	CSQA	GSM8K	MBPP	News	Avg %
Vanilla	100.0%	100.0%	100%	100%	100%	100%	100.0%
$w = 16$	66.0%	71.8%	59.5%	41.5%	41.3%	36.0%	52.7%
$w = 32$	52.1%	58.0%	43.7%	35.2%	32.0%	27.1%	41.4%
$w = 64$	34.4%	38.3%	24.9%	24.0%	24.6%	22.5%	28.1%
$w = 128$	11.2%	11.8%	7.4%	11.3%	13.6%	14.4%	11.6%
$w = 256$	5.9%	6.9%	4.4%	5.7%	7.4%	10.0%	6.7%

Table 2: Average Full Attention usage of the proposed All-or-here Attention across various window sizes.

attention via the gating mechanism $g_{i,j} = \mathbb{I}(s_{i,j} > \tau)$. This incentivizes the model to default to efficient sliding window attention, invoking global context only when necessary.

3 Experiments

3.1 Implementation Details

To evaluate the efficacy of our proposed method, we implement the AHA architecture using OLMo-2-0425-1B-SFT (OLMo et al., 2024) as the base model. We initialize the model with all original pre-trained parameters. In every attention block, we introduce the linear projection matrix $\mathbf{W}_{\text{Router}}$ to generate the routing score $s_{i,j}$ for the i-th token in the j-th head. The gating mechanism operates with a fixed threshold of $\tau = 0.5$. regarding the training objective, we set the regularization coefficient to $\lambda = 3 \times 10^{-4}$. We fine-tune the model for one epoch on the Tulu-v3 dataset (Lambert et al., 2024). Since the base model was originally fine-tuned on this same dataset, this experimental design effectively isolates the impact of AHA from potential data distribution shifts. Optimization is performed using AdamW with a learning rate of 3×10^{-5}, $\beta_1 = 0.9$, and $\beta_2 = 0.95$, alongside a linear warmup ratio of 0.03 and a global batch size of 128.

3.2 Benchmarks

We evaluate our method across a diverse set of tasks, categorized into single-token and multi-token generation. For single-token tasks, we include MMLU (Hendrycks et al., 2021) for massive multitask understanding, HellaSwag (Zellers et al., 2019) for grounded reasoning, and CSQA (Talmor et al., 2019) for commonsense question answering. For multi-token tasks, we evaluate on GSM8K (Cobbe et al., 2021) for multi-step mathematical reasoning, MBPP (Austin et al., 2021) for Python code generation, and MultiNews (Fabbri et al., 2019) for multi-document summarization.

All evaluations are conducted using the `lm-evaluation-harness` toolkit (Gao et al., 2024). We employ a 5-shot setting for MMLU, HellaSwag, CSQA, and GSM8K, and a zero-shot setting for MBPP and MultiNews. Metrics include accuracy (Acc) for MMLU and CSQA, normalized accuracy (Acc_norm) for HellaSwag, exact-match (EM) for GSM8K, Pass@1 for MBPP, and ROUGE scores for MultiNews. To ensure the sliding window attention is strictly operative, we filter out samples where the input context length is shorter than the window size. The average input context lengths for these tasks are 755, 532, 332, 939, 675, and 2,196 tokens, respectively.

3.3 Main Results

We evaluate the proposed attention mechanism by training a suite of models with varying local sliding window sizes $w \in \{16, 32, 64, 128, 256\}$, while keeping all other implementation details and hyperparameters identical to the previous section. This setup allows us to investigate how the capacity of the local context influences the router's reliance on

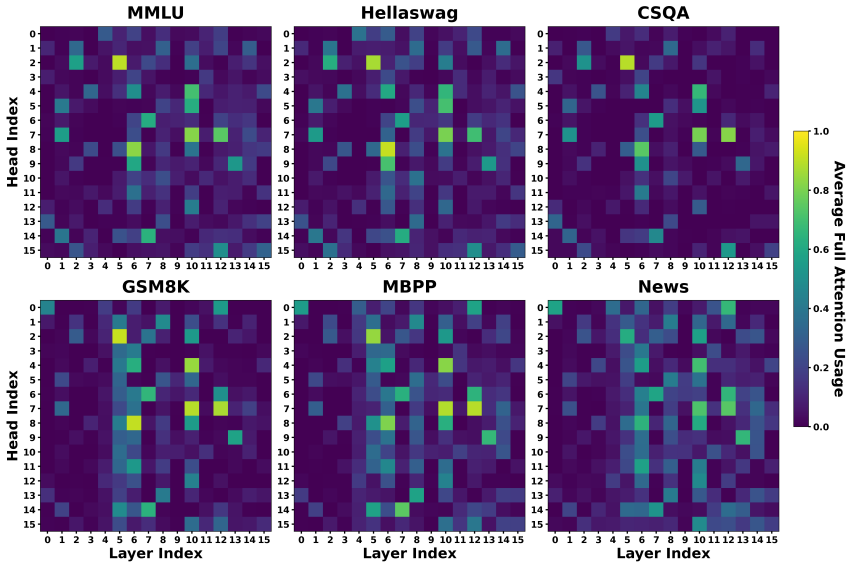

Figure 3: Visualization of the average full attention usage μ_f across different layers and heads. Lighter colors indicate a higher frequency of triggering full attention.

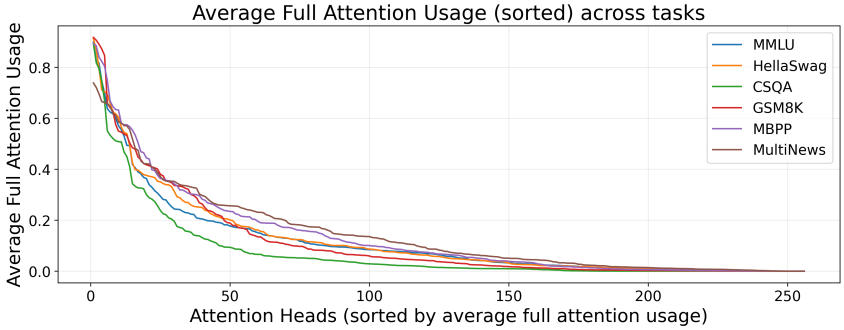

Figure 4: Sorted average full attention usage (μ_f) for all attention heads. The data shows that a small minority of heads account for the majority of full attention usage.

full attention. Crucially, we ensure that the window size w used during inference strictly matches the configuration used during training.

Table 1 summarizes the performance of the proposed All-or-here Attention across various window sizes. We observe a clear positive correlation between the window size and model performance. As the window size w increases, the model's accuracy progressively recovers, eventually matching the vanilla result of OLMo-2-SFT. Specifically, with window sizes of $w = 128$ and $w = 256$, AHA maintains 100.9% and 102.5% of the Vanilla performance, respectively. Even under the extreme constraint of $w = 16$, the model retains 92.7% of the original performance.

Table 2 details the sparsity patterns across different benchmarks. We define the average full attention usage μ_f as the mean of the binary gating values $g_{i,j}^{(k)}$ (Equation 2) averaged across all L layers, m heads, and n generated tokens:

$$\mu_f = \frac{1}{L \cdot n \cdot m} \sum_{k=1}^{L} \sum_{i=1}^{n} \sum_{j=1}^{m} g_{i,j}^{(k)}. \tag{8}$$

This metric directly reflects the frequency with which the model selects the full attention path. Analyzing the variations of μ_f presented in Table 2, the results underscore both the efficiency and the adaptability of our approach. First, in terms of efficiency, AHA demonstrates that the vast majority of full attention is redundant given sufficient local context. At $w = 256$, the model successfully bypasses 93.3% of the full attention operations. Our method also exhibits robust adaptability to the available window size. We observe a clear inverse correlation between window size and full attention usage. When the local window is severely restricted (e.g., $w = 16$), the router effectively identifies the information deficit and compensates by increasing

global access to 52.7%. This dynamic adjustment allows the model to retain 92.7% of the original quality even under extreme constraints, confirming that the gating mechanism is actively responsive to the token-level context needs rather than learning a static bias.

Finally, we analyze the relationship between the window size and the model's reliance on full attention based on the statistics in Table 2. We observe that the demand for full attention diminishes rapidly as the local window expands. A distinct turning point becomes evident between $w = 128$ and $w = 256$. As the window size reaches 128, the average full attention usage drops to 11.6%, and further declines to a mere 6.7% at $w = 256$. This empirical evidence suggests that context dependencies exhibit a long-tailed nature: the vast majority of tokens depend primarily on the local area, while only a sparse minority necessitate attention to access the global context. We hypothesize that this mirrors the intrinsic structure of natural language, where most syntactic and anaphoric dependencies occur within short ranges, with only a sparse long-range tail spanning hundreds of tokens (Liu et al., 2017; Fan and Jiang, 2023; Futrell et al., 2015).

3.4 Sparsity Patterns

To investigate the internal mechanism of All-or-Here Attention, we visualize the average full attention usage μ_f across all layers and heads for various tasks in Figure 3. The heatmaps illustrate the average full attention usage of different attention heads. From these visualizations, we characterize two primary phenomena regarding the model's attention behavior.

First, the utilization of full attention is remarkably sparse and follows a distinct long-tail distribution. As shown in the heatmaps, the majority of the heads remain dark, indicating they rely almost exclusively on local context. This is further quantified in Figure 4, where heads are ranked by their average full attention usage μ_f. The resulting curve exhibits a sharp decay: a few "heavy-hitter" heads maintain high activation rates, while the usage for the vast majority drops precipitously toward zero.

Second, our analysis shows that many attention heads develop distinct functional specializations, where certain heads predominantly invoke full attention while others rely almost exclusively on local sliding window attention. As shown in Table 3, which measures the average token interval between consecutive full attention triggers, specific components (e.g., Layer 5, Head 2) maintain an interval close to 1.0 across all benchmarks, attending globally for nearly every token, whereas others (e.g., Layer 3, Head 6) process thousands of tokens between global triggers.

Surprisingly, these observations broadly align with the head classifications reported in DuoAttention (Xiao et al., 2025) and Prulong (Bhaskar et al., 2025), where attention heads are characterized as either 'streaming' or 'retrieval' components. However, a closer inspection of Figure 3 reveals that the activation intensities and patterns of these heads still exhibit nuanced variations across tasks. Intuitively, since different tasks inherently impose distinct demands on global context, an adaptive setting might be better suited to capturing shifting requirements than a static configuration. We will demonstrate in the next section that this flexibility enables AHA to outperform static assignment strategies under identical global sparsity constraints.

3.5 Ablation Studies

Impact of Regularization Penalty. To investigate the trade-off between efficiency and performance, we trained models with varying L_1 regularization coefficients $\lambda \in \{1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}\}$ while fixing the window size at $w = 128$. Table 4 summarizes the results. We observe that the penalty coefficient serves as a pivotal hyperparameter for balancing sparsity and model capability. An overly aggressive penalty ($\lambda = 1 \times 10^{-3}$) forces the model into extreme sparsity (only 4.5% global attention), precipitating performance degradation, particularly on reasoning-intensive tasks like GSM8K (dropping to 0.3730) and MBPP (dropping to 0.1320). This indicates that complex reasoning necessitates a minimum threshold of global context. Conversely, relaxing the penalty to $\lambda = 1 \times 10^{-4}$ significantly increases global attention usage to 53.5% without yielding meaningful performance gains. This suggests that even with a larger computational budget for global attention, the model extracts no additional signal, confirming that full attention is largely redundant.

Dynamic vs. Static Head Assignment. To verify the necessity of the dynamic routing mechanism, we compare AHA against a static assignment baseline. Drawing from the sparsity patterns identified in Figure 3, we rank all 256 attention heads ($L = 16, m = 16$) of the trained AHA model

Head	MMLU	Hellaswag	CSQA	GSM8K	MBPP	NEWS
Layer 5, Head 2	1.11	1.15	1.12	1.09	1.07	1.04
Layer 12, Head 8	8.83	22.23	525.43	56.07	27.68	2.02
Layer 3, Head 6	427.54	1114.39	4154.94	10591.20	29228.29	268.54

Table 3: Average full attention gap for representative heads across different benchmarks. The gap is defined as the mean number of tokens processed between consecutive triggers of the global attention path for a specific head.

Penalty (λ)	MMLU	HellaSwag	CSQA	GSM8K	MBPP	News	Avg Full %
1×10^{-3}	0.4045	0.6419	0.4878	0.3730	0.1320	0.1997	4.5%
3×10^{-4}	0.4087	0.6445	0.4996	0.4291	0.1580	0.2053	11.6%
1×10^{-4}	0.4042	0.6456	0.4996	0.4397	0.1600	0.2047	53.5%

Table 4: Ablation study on the L1 regularization penalty (λ).

by their average full attention usage μ_f. We then evaluate static variants where the top-k heads are fixed to full attention while the remaining heads are restricted to a local sliding window of $w = 128$.

Table 5 presents the results for $k \in \{32, 64, 128, 256\}$. We observe that static assignment is significantly less efficient than AHA's adaptive routing. For instance, the Top-32 static configuration utilizes 12.5% full attention, yet its performance is markedly inferior to the default AHA ($w = 128$), which achieves better results across nearly all benchmarks while using only 11.6% full attention (as shown in Table 4). The disparity is particularly acute in reasoning-heavy tasks; on GSM8K, the Top-32 static model drops to a near-zero score of 0.0682, whereas AHA maintains 0.4291. Even as we increase the static budget to Top-128 (50% full attention), the performance on GSM8K and MBPP still lags behind our adaptive model. These results demonstrate that the optimal attention scope for a head is not merely a function of its identity but is highly dependent on the real-time context. AHA's ability to dynamically evoke full attention on demand allows it to preserve critical reasoning capabilities with a smaller computational footprint.

4 Related Work

4.1 Efficient Attention

To address the quadratic computational complexity of standard self-attention, efficient attention mechanisms are generally categorized into linear, sparse, and hierarchical approaches. Linear attention reformulates the attention mechanism to scale linearly with sequence length, often employing kernel feature maps or recurrent state updates, as seen in the autoregressive framework of LinearAttn (Katharopoulos et al., 2020) and the selective state-space models of Mamba (Gu and Dao, 2024). Sparse attention reduces memory footprint by restricting token interactions to specific subsets. This includes static block-sparse patterns in Sparse Transformers (Child et al., 2019a) and structured sliding windows in Longformer (Beltagy et al., 2020a) and BigBird (Zaheer et al., 2020b). More advanced works have explored adaptive sparsity, ranging from Differentiable Attention Window (Nguyen et al., 2020) to Adaptive Attention Span (Sukhbaatar et al., 2019; Fu et al., 2025), which dynamically determine the optimal context span for each head. Finally, hierarchical attention organizes information processing across multiple levels of granularity, from the sentence-to-document aggregation in HAN (Yang et al., 2016) to the hardware-aligned, coarse-to-fine compression in Native Sparse Attention (Yuan et al., 2025).

In contrast to these increasingly sophisticated designs, our method demonstrates that complex span decision processes or hierarchical selection could be simplified when handling many tasks. We show that a minimalist binary routing decision—selecting strictly between local and global scopes—can eliminate over 90% of full attention operations while maintaining model performance. Our method does not focus on the memory complexity of attention. A recent comprehensive analysis of KV cache eviction and assignment methods in long-context settings is provided in (LI et al., 2025). Their finding shows that sub-O(n) methods struggle with complex interactions.

4.2 Conditional Computation

The notion of selectively skipping computation in deep neural networks can be traced back to condi-

Setting	MMLU	HellaSwag	CSQA	GSM8K	MBPP	News	Full %
Top-32	0.3962	0.6399	0.4976	0.0682	0.1180	0.1973	12.5%
Top-64	0.3938	0.6349	0.5078	0.1562	0.1340	0.1958	25.0%
Top-128	0.3962	0.6399	0.5291	0.4064	0.1540	0.2031	50.0%
Top-256 (full)	0.3975	0.6419	0.5127	0.4238	0.1640	0.2047	100.0%

Table 5: Ablation study on static head assignment. Heads are ranked by their average full attention usage. "Top-k" indicates the number of heads assigned to full attention.

tional computation, which aims to reduce computational cost by activating only a subset of model components conditioned on the input (Bengio et al., 2013). This idea has been extensively explored through mechanisms such as adaptive computation time (Graves, 2017), mixture-of-experts (Shazeer et al., 2017), and mixture-of-depth (Raposo et al., 2024). In Transformer-based architectures, prior work has shown that many attention heads are redundant or contribute unevenly across inputs, motivating approaches such as head pruning and head importance estimation (Michel et al., 2019; Voita et al., 2019). Complementary to these efforts, sparse and block-wise attention mechanisms restrict global attention to a subset of tokens or regions to improve efficiency (Child et al., 2019b; Beltagy et al., 2020b; Zaheer et al., 2020a). More recent methods focus on how to make attention more efficient, e.g., FlashAttention (Dao et al., 2022a) and hierarchical attention (Yang et al., 2016; Yuan et al., 2025). Orthogonal to these studies, we demonstrate that in most cases global attention is not required at all, providing strong motivation for learning when and where to attend globally.

5 Conclusion

In this paper, we introduced All-or-Here Attention, a simple yet effective paradigm that dynamically toggles between efficient local sliding window attention and full global attention at the token level. Our empirical results on OLMo-2 confirm that the vast majority of global attention operations are redundant. We demonstrate that with a window size of 256 tokens, AHA can eliminate over 93% of full attention without compromising performance on standard benchmarks. Furthermore, our analysis reveals that context dependency follows a long-tail distribution: as the local receptive field expands, the need for global access decays rapidly. These findings indicate that the key to efficiency lies not only in speeding up full attention, but also in accurately identifying the sparse moments when global context is truly required. We hope this work inspires further exploration of simple, dynamic strategies for efficient language modeling and their GPU-friendly adaptation.

6 Limitations

While our experiments empirically demonstrate that the vast majority of global attention operations are redundant, our analysis focuses on the resulting attention sparsity rather than system-level metrics such as wall-clock speedup. Realizing the practical acceleration from this sparsity presents non-trivial engineering challenges. Current hardware accelerators and standard attention kernels (e.g., FlashAttention (Dao et al., 2022b)) are heavily optimized for static, dense matrix operations. In contrast, AHA introduces dynamic, heterogeneous execution paths where different heads process varying context lengths within the same batch. Consequently, we position AHA as a proof-of-concept for computational redundancy, leaving the development of hardware-aware kernels to leverage this sparsity for future work. Furthermore, our current approach employs a strictly binary routing mechanism. It is worth extending this design to support a multi-choice decision space, enabling the model to dynamically select among global attention and multiple local windows of varying sizes, potentially integrated with paradigms like Mixture of Attention Spans (Fu et al., 2025).

Due to limitations in both lab computational resources and available training data, we conducted experiments only on OLMo-2, which uses full attention. It would be interesting to extend our evaluation to other attention mechanisms. Since index-based attention (Yang et al., 2016), top-k attention (Xiao et al., 2025), and sparse attention (Child et al., 2019a) are all approximations of full attention, if full attention itself does not need to be activated frequently, it is reasonable to expect that these approximations would likewise remain inactive even if included. However, this hypothesis requires further experimental verification.

References

Joshua Ainslie, Tao Lei, Michiel de Jong, and 1 others. 2023. Colt5: Faster long-range transformers with conditional computation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*.

Jacob Austin, Augustus Odena, Maxwell Nye, and 1 others. 2021. Program synthesis with large language models. *ArXiv*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2020. Controlling computation versus quality for neural sequence models. *CoRR*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020a. Longformer: The long-document transformer. *ArXiv*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020b. Longformer: The long-document transformer. *arXiv:2004.05150*.

Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. 2015. Conditional computation in neural networks for faster models. *CoRR*.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv*.

Adithya Bhaskar, Alexander Wettig, Tianyu Gao, and 1 others. 2025. Cache me if you can: How many kvs do you need for effective long-context lms? *CoRR*.

Tom B. Brown, Benjamin Mann, Nick Ryder, and 1 others. 2020. Language models are few-shot learners. *ArXiv*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019a. Generating long sequences with sparse transformers. *ArXiv*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019b. Generating long sequences with sparse transformers. *Preprint*, arXiv:1904.10509.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, and 1 others. 2021. Training verifiers to solve math word problems. *ArXiv*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022a. Flashattention: fast and memory-efficient exact attention with io-awareness. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022b. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Alexander R. Fabbri, Irene Li, Tianwei She, and 1 others. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Annual Meeting of the Association for Computational Linguistics*.

Lu Fan and Yue Jiang. 2023. Probability distribution of dependency distance and dependency type in translational language. *Humanities and Social Sciences Communications*.

Tianyu Fu, Haofeng Huang, Xuefei Ning, Genghan Zhang, Boju Chen, Tianqi Wu, Hongyi Wang, Zixiao Huang, Shiyao Li, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. 2025. Mixture of attention spans: Optimizing LLM inference efficiency with heterogeneous sliding-window lengths. In *Second Conference on Language Modeling*.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*.

Tow Gao, Leo and 1 others. 2024. A framework for few-shot language model evaluation.

Alex Graves. 2017. Adaptive computation time for recurrent neural networks. *Preprint*, arXiv:1603.08983.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Franccois Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*.

Nathan Lambert, Jacob Daniel Morrison, Valentina Pyatkin, and 1 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *ArXiv*.

YUCHENG LI, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. 2025. SCBench: A KV cache-centric analysis of long-context methods. In *The Thirteenth International Conference on Learning Representations*.

Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of life reviews*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32.

Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq R. Joty, and Xiaoli Li. 2020. Differentiable window for dynamic local attention. *ArXiv*.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, and 1 others. 2024. 2 olmo 2 furious. *ArXiv*.

Long Ouyang, Jeff Wu, Xu Jiang, and 1 others. 2022. Training language models to follow instructions with human feedback. *ArXiv*.

David Raposo, Samuel Ritter, Blake A. Richards, Timothy P. Lillicrap, Peter Conway Humphreys, and Adam Santoro. 2024. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *CoRR*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. adaspan. *ArXiv*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *ArXiv*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, and 1 others. 2023. Llama: Open and efficient foundation language models. *ArXiv*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, and 1 others. 2025. Duoattention: Efficient long-context LLM inference with retrieval and streaming heads. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.

An Yang, Anfeng Li, Baosong Yang, and 1 others. 2025. Qwen3 technical report. *ArXiv*.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2024. Gated delta networks: Improving mamba2 with delta rule. *ArXiv*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *North American Chapter of the Association for Computational Linguistics*.

Jingyang Yuan, Huazuo Gao, Damai Dai, and 1 others. 2025. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Annual Meeting of the Association for Computational Linguistics*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020a. Big bird: transformers for longer sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, and 1 others. 2020b. Big bird: Transformers for longer sequences. *ArXiv*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*.

Xiaofeng Zhang, Yikang Shen, Zeyu Huang, and 1 others. 2022. Mixture of attention heads: Selecting attention heads per token. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*.