Every Token Counts: Generalizing 16M Ultra-Long Context in Large Language Models

Xiang Hu[1*], **Zhanchao Zhou**[1,2*], **Ruiqi Liang**[1], **Zehuan Li**[1],
Wei Wu[1], **Jianguo Li**[1†]
[1] Ant Group [2] Westlake University
{aaron.hx, zhouzhanchao.zzc, liangruiqi.lrq,
lizehuan.lzh, congyue.ww, lijg.zero}@antgroup.com

Abstract

This work explores the challenge of building "**Machines that Can Remember**", framing long-term memory as the problem of efficient ultra-long context modeling. We argue that this requires three key properties: **sparsity**, **random-access flexibility**, and **length generalization**. To address ultra-long-context modeling, we leverage Hierarchical Sparse Attention (HSA), a novel attention mechanism that satisfies all three properties. We integrate HSA into Transformers to build HSA-UltraLong, which is an 8B-parameter MoE model trained on over 8 trillion tokens and is rigorously evaluated on different tasks with in-domain and out-of-domain context lengths to demonstrate its capability in handling ultra-long contexts. Results show that our model performs comparably to full-attention baselines on in-domain lengths while achieving over 90% accuracy on most in-context retrieval tasks with contexts up to 16M. This report outlines our experimental insights and open problems, contributing a foundation for future research in ultra-long context modeling.

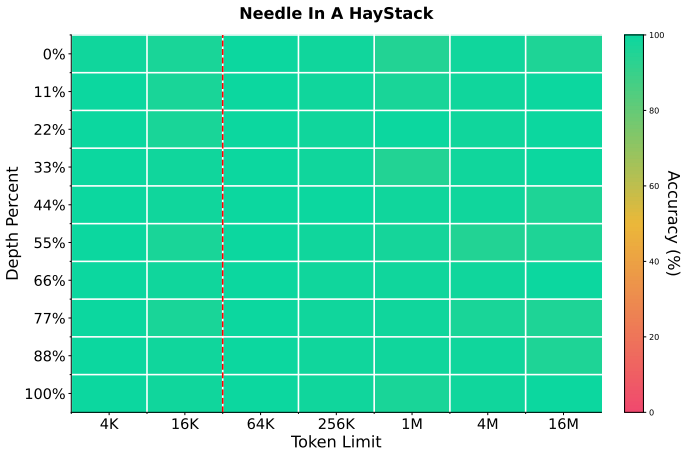


Figure 1: Despite being pre-trained with an 8K context window and mid-trained up to 32K, HSA-UltraLong achieves near-perfect accuracy on S-NIAH even at a 16M-token context length. The red dashed line at 32K marks the boundary between in-domain (left) and out-of-domain (right).

[*] Equal contribution
[†] Corresponding author

1 Introduction

Despite the impressive capabilities of Large Language Models (LLMs) [5, 1, 36], their world knowledge is confined to static parameters, making it inflexible to update and impossible to learn dynamically from daily user interactions. This limitation motivates a fundamental question: how can we build machines that truly remember? Effective memory is critical for future AI agents, enabling each user to have a personalized agent that accumulates unique experiences over time. Human memory spans the entire context from birth to the present, suggesting that the problem of machine memory is closely related to ultralong context modeling. Imagine if Transformers could efficiently handle infinite-length contexts—encompassing all pre-trained tokens—so that most world knowledge can be retrieved from context rather than compressed into model parameters. Furthermore, skills and the latest information could be acquired via in-context learning rather than through costly model retraining. Such advances would dramatically improve the online learning of knowledge and skills.

However, the Transformer [37] architecture, the backbone of modern LLMs, faces a fundamental efficiency challenge when processing ultra-long sequences, due to both poor length generalization and the quadratic computational complexity of full attention. Supporting longer contexts requires training models with extended context windows, yet simply scaling context length is computationally prohibitive. To support ultra-long context, we argue that there are three key properties:

- **Sparsity**: Human long-term memory operates through selective activation rather than full activation, retrieving relevant fragments as needed [8]. Clearly, using full attention to achieve an infinitely long context is not feasible. Therefore, implementing sparsity is one of the necessary conditions for ultra-long context modeling.
- **Random-Access Flexibility**: The effectiveness of sparsity relies on accurately retrieving relevant past information. Therefore, it is crucial to design an intrinsic retrieval mechanism within the model and optimize it end-to-end under the guidance of an auto-regressive loss.
- **Length Generalization**: Pretraining with an infinite context is impossible. To achieve the goal, the path must involve generalizing retrieval ability from short to long contexts.

While several approaches show promising paths to achieve the goal, each presents notable shortcomings. Recurrent architectures, such as Mamba [12, 9] and Linear Attentions [21, 46], compress past variable-length information into a fixed-dimensional state vector. This introduces an information bottleneck and sacrifices random access to distant tokens. Similarly, sliding-window attention [2] suffers from the same fundamental constraint on distant context accessibility. Sparse attention approaches like NSA [47] and MoBA [28] improve training and inference efficiency over long sequences, but our empirical studies show they suffer from inaccurate chunk selection, which leads to both in-domain and out-of-domain performance degradation on in-context retrieval tasks.

A recent line of work that combines model-inherent retrieval [29, 19] with chunk-wise sparse attention, such as Hierarchical Sparse Attention (HSA) [18], has shown promising results in long-context modeling. Empirical studies [23] report that an HSA-based model pre-trained with a 4K context length can extrapolate to more than 10M context length while keeping high accuracy on the RULER [17] and BabiLong [22] benchmark, which simultaneously satisfies **sparsity**, **random-access flexibility** and **length generalization**. The method partitions text into fixed-length chunks with landmark representations; each token retrieves top-k relevant past chunks via these landmarks. The core innovation of HSA is to conduct attention with each chunk *separately*, and then *fuse the results weighted by the retrieval scores*. The overall process closely resembles the Mixture-of-Experts (MoE) [32], as illustrated in Figure 2. This design allows the retrieval scores to be integrated into the forward pass, enabling them to receive gradient updates during backpropagation. As a result, the model learns to assign higher retrieval scores to chunks that are more helpful for next token prediction. However, current work in this area is limited in scale and lacks results on data scaling, parameter scaling, and attention field scaling.

This work investigates the in-domain performance, extrapolation capability, and scaling challenges of HSA-based models, demonstrating a promising path toward effective ultra-long context modeling. We introduce HSA-UltraLong, an architecture that combines sliding-window attention with HSA, and train models from scratch across scales, including a 0.5B dense model and an 8B-A1B MoE model, on 8 trillion tokens. Training is followed by long-context extension and annealing. To preserve in-domain capability, we use a 4K-token sliding window. Our key findings include:

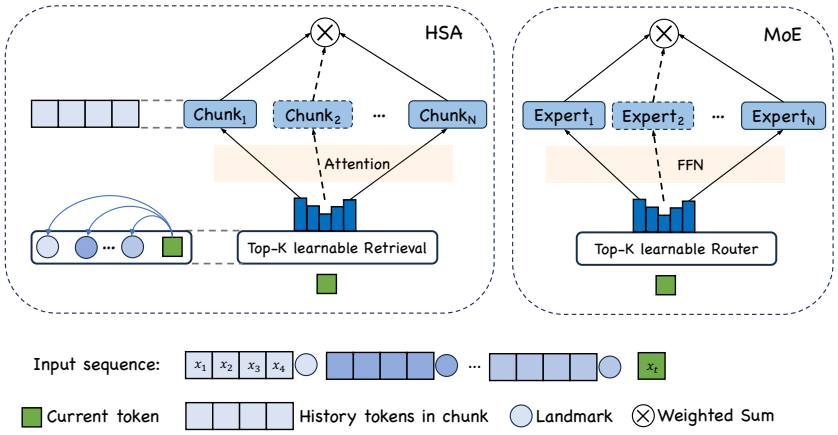


Figure 2: Hierarchical Sparse Attention (HSA) operates in a manner analogous to Mixture of Experts (MoE). First, the current token x_t computes dot products with the landmark representations of past chunks as retrieval scores, from which the top-k chunks are selected—similar to how MoE uses a router to select top-k experts. Subsequently, x_t performs attention with each of the k retrieved chunks **separately**, mirroring the process in MoE where x_t independently conducts Feed-forward with k experts. Finally, the attention outputs from each chunk are weighted by the softmax-normalized retrieval scores and summed, which is functionally equivalent to MoE's fusion of outputs from the selected FFNs.

- Effective length generalization requires the combination of chunk-wise attention, retrieval score–based fusion, and NoPE (No Positional Encoding); all three are essential.
- Sliding-window attention and HSA interact in nontrivial ways. HSA's long-range generalization arises from learning to retrieve over short contexts and transferring that ability to long contexts. However, an overly large sliding window can weaken HSA's learning of short-range dependencies, degrading generalization.
- The effective context length in the training corpus strongly influences length extrapolation.

We demonstrate successful extrapolation from a 32K pre-training window to 16M tokens trained on trillion-token-scale data. Our study presents a detailed analysis of in-domain and extrapolation performance with respect to model size, attention window size, and data scaling, offering extensive experimental results and insights for future research on infinite-length context modeling.

2 Preliminary

2.1 Limitations of Chunk Selection in NSA

NSA is a highly inspiring contribution to sparse attention. However, our empirical study in Table 1 shows that its chunk selection mechanism does not always pick the most relevant chunk. On the RULER benchmark [17], NSA fails to achieve perfect accuracy even on in-domain tasks such as Multi-Query NIAH. We trace this to a key limitation: the chunk selection action is not end-to-end learnable. Further, our analysis of NSA's extrapolation ability indicates that performance on in-context retrieval degrades rapidly as context length increases. Regarding positional encoding, we also find that No Positional Encoding (NoPE) supports extrapolation better than RoPE [33].

Table 1: NSA ablation with 4K as the in-domain length. The higher scores are shown in **bold**.

Models	#params	Single-NIAH (ACC ↑)					MQ-NIAH(ACC ↑)				
		4K	8K	16K	32K	64K	4K	8K	16K	32K	64K
NSA(w/ RoPE)	370M	97.0	90.0	83.0	73.0	60.0	72.0	50.0	24.0	15.0	4.0
NSA(w/o RoPE)	370M	**99.0**	**96.0**	**88.0**	**84.0**	**73.0**	**83.0**	**66.0**	**51.0**	**40.0**	**12.0**

2.2 Attention with Chunk Retrieval

As we mentioned, the challenge of sparse attention lies in accurately retrieving the previous chunks. Hierarchical Sparse Attention (HSA) addresses the challenge by jointly learning chunk selection and attention in an end-to-end manner. Compared to NSA, HSA mainly makes two contributions:

- **Retrieval-oriented sparse attention.** Specifically, each token conducts attention with each past chunk **separately** and then fuses the attention results via retrieval scores.
- **RoPE for short, NoPE for long.** To mitigate the negative impact of RoPE on extrapolation, the sliding-window attention's KV cache employs RoPE, while the HSA uses NoPE.

Formally, for an input sequence $\mathbf{S} = \{x_0, x_1, ..., x_n\}$, where n is the length of the sequence. We denote the hidden states of tokens as $\mathbf{H} \in \mathbb{R}^{n \times d}$, where d is the hidden dimension. The whole sequence is split into chunks according to a fixed length S, which is set to 64 by default to better align with hardware, thus we have $\frac{n}{S}$ chunks in total. We use indices with $[\cdot]$ to indicate that it is indexed by chunk rather than by token, e.g., $\mathbf{H}_{[i]} := \mathbf{H}_{iS:(i+1)S} \in \mathbb{R}^{S \times d}$. For each chunk, it has its own KV cache as $\mathbf{K}_{[i]}, \mathbf{V}_{[i]} \in \mathbb{R}^{S \times h \times d_h}$, with h as the number of heads satisfying $h \times d_h = d$, and its landmark representation as $\mathbf{K}^{slc}_i \in \mathbb{R}^d$, which serves to summarize the content of the chunk. For each token, it uses $\mathbf{Q}^{slc}_t \in \mathbb{R}^d$ to retrieve chunks and $\mathbf{Q}^{attn}_t \in \mathbb{R}^{h \times d_h}$ to conduct attention with tokens inside chunks, both of which are derived from $\mathbf{H}_i$ via linear transformations.

$$s_{t,i} = \begin{cases} \mathbf{Q}^{slc\top}_t \mathbf{K}^{slc}_i / \sqrt{d}, & i \leq \lfloor \frac{t}{S} \rfloor \\ -\infty, & i > \lfloor \frac{t}{S} \rfloor \end{cases}, \quad \mathfrak{I}_t = \{i | \operatorname{rank}(s_{t,i}) < K\},$$

where $\operatorname{rank}(\cdot)$ denotes the ranking position in descending order, and $\mathfrak{I}_t$ is the indices of K chunks with the highest relevance scores for x_t.

$$\bar{\mathbf{O}}_{\mathbf{t},\mathbf{i}} = \text{Attention}(\mathbf{Q}^{attn}_t, \mathbf{K}_{[i]}, \mathbf{V}_{[i]}) = \underbrace{\text{Softmax}\left(\frac{\operatorname{norm}(\mathbf{Q}^{attn}_t) \operatorname{norm}(\mathbf{K}^{\top}_{[i]})}{\sqrt{d_h}}\right) \mathbf{V}_{[i]}}_{\textit{intra-chunk attention}}, \tag{1}$$

$$w_{t,i} = \frac{\exp(s_{t,i})}{\sum_{k \in \mathfrak{I}_t} \exp(s_{t,k})}, \quad \mathbf{O_t} = \underbrace{\sum_{k \in \mathfrak{I}_t} w_{t,k} \bar{\mathbf{O}}^l_{t,k}}_{\textit{inter-chunk fusion}}. \tag{2}$$

norm is the Query-Key Normalization [11, 41], which we find to be very important for the stability of HSA in practical trillion-token scale training.

3 Methodology

In terms of model design, we use SWA for local information retrieval and HSA for global information retrieval, fusing both local and global information through a stacking approach. A key challenge of long sequence inference is that the KV cache grows with the sequence length. Previous works [42, 30] have demonstrated that sharing the KV cache can significantly compress its size while maintaining comparable results. Inspired by these works, we share the intermediate layer KV cache among all HSA modules to serve as context memory.

3.1 Model Architecture

Overall, as shown in Figure 3, our model contains L layers, which are divided into the upper decoder and the lower decoder. The lower decoder is composed of $\frac{L}{2}$ standard Transformer layers with SWA. For the upper decoder, we divide it into G groups, where each group consists of one Transformer layer with both SWA and HSA, followed by several layers with SWA only. For each chunk, we use a bi-directional encoder to obtain its summary representation. We denote $\mathbf{H}^l$ as the output hidden states of the l-th layer, and use the derivations from the intermediate layer output $\mathbf{H}^{\frac{L}{2}}$, as the chunk summary representation and KV cache, which are shared across all HSA modules. Each

`https://github.com/ant-research/long-context-modeling`

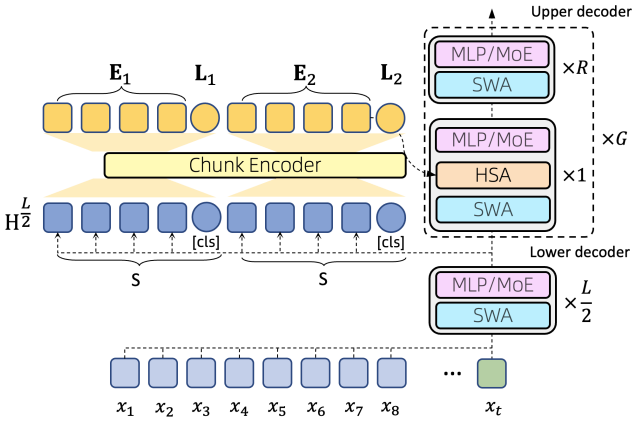


Figure 3: HSA-UltraLong model architecture.

$\mathbf{H}^{\frac{L}{2}}_{[i]} \in \mathbb{R}^{S\times d}$ is accompanied by a [CLS] token, and is fed into a bi-directional encoder, as shown in Figure 3, to obtain $\mathbf{E}_{[i]} \in \mathbb{R}^{S\times d}$ and $\mathbf{L}_i \in \mathbb{R}^d$. Finally, the keys and values used in Equation 1 are obtained by applying a linear transformation to $\mathbf{E}_{[i]}$. Regarding MoE, we follow the design of Ling-2.0 [26], where the first layer of the model adopts a dense MLP structure and all subsequent layers use MoE. Each MoE block has one shared expert, following the design in DeepSeek V3 [10]. We use training-free balance strategy [38] as the expert balancing strategy.

3.2 Training

Previous work [23] demonstrates that with a 512-token sliding window, HSA-based models pre-trained on a 4K context length can generalize to 32M on the RULER task with high accuracy. However, such a small sliding window often sacrifices downstream performance. To address this, we increase the sliding window size to 4K tokens. However, when training models from scratch with a 4K sliding window, we observe that ***they fail to generalize beyond the 4K context length.*** We hypothesize that HSA's length generalization ability arises from its inherent retrieval mechanism, which learns to generalize from short to long contexts. An overly large sliding window can randomly access any short-range information, making it unnecessary for the HSA to focus on short-range patterns, thus preventing it from receiving meaningful gradients for short-context learning. To overcome this limitation, we introduce a warmup stage before the pre-training phase. The whole pre-training procedure is as follows:

- **Warm-up.** We use a short sliding window attention (SWA) of 512 tokens with a global HSA, setting top-k large enough to cover the full sequence. Synthetic ruler tasks are randomly inserted into 1% of training samples. The warm-up phase is considered complete once the model achieves high needle-in-a-haystack retrieval accuracy on contexts well beyond the 512-token window. At this step, the context length is set to 16K.
- **Pre-training.** After the warm-up phase, we increase the SWA window size to 4K and decrease the HSA top-k, transitioning from dense to sparse attention. Continue training from the checkpoint at the end of warm-up. The context length remains 16K.
- **Long-context mid-training.** Switch to corpora with longer effective contexts and raise HSA top-k to cover the full sequence. The context length is expanded to 32K.
- **Annealing.** Perform annealing on high-quality data while keeping a 32K context length.
- **Supervised fine-tuning.** Perform supervised fine-tuning (SFT) with an 8K context length.

4 Experiments

Training Data In the first phase of general pretraining, we utilized a large-scale deduplicated, multi-domain dataset totaling 10T tokens, with differential sampling ratios across various sub-datasets from different domains. This distribution comprised predominantly Web content (50%),

Table 2: Preliminary experiments on HSA-UltraLong-Base with a training context length of 16k tokens. The highest and second-best scores are shown in **bold** and underlined, respectively.

Models	#params	Warmup	PG19 (PPL ↓)			MQ-NIAH(ACC ↑)			
			4K	8K	16K	4K	8K	64K	1M
BaseLM	519.6M	-	18.61	17.53	16.77	89.0	23.0	5.0	0.0
SWA+HSA	537.7M	self-copy	18.87	17.44	16.50	**100.0**	**96.0**	**93.0**	**93.0**
SWA+HSA	537.7M	short-swa,full-hsa	**18.30**	**17.13**	**15.96**	99.0	95.0	90.0	66.0

followed by Code (14.4%), Math (12.0%), Code-nlp (5.6%), Reason (5%), Multilingual (4.0%), Books (2.0%), Wikipedia (1.5%), and Others (5.5%). During this phase, the MoE model processed 8T tokens while the dense model was trained on 4T tokens. The second phase uses a dataset of 32K-length long-text sequences totaling 175B tokens and the third phase consisted of 400B tokens with a high proportion of reasoning data. During the Supervised Fine-tuning phase, we utilized the same dataset as described in [43].

Hyperparameters All models are trained using AdamW optimizer with a weight decay of 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and gradient clipping norm of 1.0. We use FSDP2 for distributed training. For the MoE model, we employ a learning rate of 3.87e-4, sequence length of 16,384, and batch size of 16.8M tokens. The dense model is trained with a learning rate of 4.96e-4 and batch size of 5.2M tokens. The learning rate schedule begins with a linear warmup phase followed by a constant learning rate maintained until training completion. For the Supervised Fine-tuning stage, we adopt a cosine decay learning rate schedule. The dense model uses a learning rate of 5.5e-5 and was trained for up to 5 epochs, while the MoE model uses a learning rate of 3.87e-4 and was trained for up to 3 epochs. In both cases, we select the checkpoint from the epoch that yields the best performance.

Evaluation Benchmarks To conduct a comprehensive evaluation of the model, we selected a diverse range of assessment tasks, encompassing four major categories: general tasks, mathematical tasks, coding tasks, and alignment tasks:

- **General Tasks:** MMLU [15], CMMLU [24], C-Eval [20], ARC [3], AGIEval [49], PIQA [4], HellaSwag [48] and BBH [34].
- **Math Tasks:** GSM8K [7], MATH [16], CMATH [40], MATH-500 [25] and OlympiadBench [14].
- **Coding Tasks:** HumanEval [6], HumanEval+ [27], MBPP [35], MBPP+ [27], and CRUX-O [13].
- **Alignment Tasks:** We report the average prompt-level strict accuracy of IFEval [50]

4.1 Small-scale Preliminary Experiments

In this section, we detail the architecture of HSA-UltraLong and validate its ability to balance in-domain performance with length extrapolation capabilities through small-scale experiments. We compared our model against Base Language Model (BaseLM), which uses full attention across all layers. HSA-UltraLong employs SWA with a window size of 4K across all layers, while strategically replacing two layers with HSA layers. The HSA layers maintain a SWA window size of 512 and, following our findings in Section 2.1, we removed positional encoding information from these layers. Each HSA layer includes an additional encoder sub-layer, resulting in less than 5% parameter increase compared to BaseLM. Within the HSA layers, we set the chunk size to 64 and top-k to 64, establishing a fixed historical context window of 4,096 tokens.

Our initial experiments revealed limited length extrapolation capabilities when training the model directly with such configuration. We hypothesized that this limitation stemmed from the predominance of pretraining data requiring only short-range modeling capabilities within the 4K window of SWA layers, leaving the HSA modules insufficiently trained. To address this issue, we explored two warm-up strategies:

- **Self-copy warm-up:** We keep the model architecture unchanged and initialize training with a self-copy objective. Given an input sequence $\mathbf{S} = \{x_1, \ldots, x_n\}$, we construct

Table 3: Comparison among HSA-UltraLong-Base (HSA-UL-Base) and other baselines. All models were evaluated under a unified framework for fair comparison.

	Qwen2.5 Annealing	**Qwen3 Annealing**	**HSA-UL Annealing**	**TRM-MoE Base**	**HSA-UL Base**	**HSA-UL Annealing**
Architecture	Dense	Dense	Dense	MoE	MoE	MoE
# Total Params	0.5B	0.6B	0.5B	8B	8B	8B
# Activated Params	0.5B	0.6B	0.5B	1B	1B	1B
# Training Tokens	18T	36T	4T	8T	8T	8T
General Tasks						
BBH	32.27	41.28	18.15	50.34	51.70	60.11
ARC-C	55.25	66.10	46.10	72.20	67.80	71.53
AGIEval	30.01	33.58	29.29	38.64	36.52	44.08
HellaSwag	48.05	48.88	44.48	67.69	67.39	67.43
PIQA	70.46	71.33	70.29	77.48	78.84	80.69
MMLU	49.73	54.40	41.76	58.74	57.83	60.71
CMMLU	52.10	51.97	42.08	57.68	57.49	64.41
C-Eval	54.17	54.57	44.30	56.87	58.36	65.98
Math Tasks						
GSM8K	41.32	60.88	37.45	66.41	67.02	72.93
MATH	18.14	31.44	20.66	37.96	41.98	48.00
CMATH	52.09	66.67	60.75	74.59	74.13	82.88
Coding Tasks						
HumanEval+	24.39	26.83	29.27	48.17	50.61	61.59
MBPP+	32.80	38.36	20.63	50.26	55.82	62.17
CRUX-O	14.38	31.62	22.56	35.12	36.31	40.75
AVG	41.08	48.42	37.70	56.58	57.27	63.09

a target sequence $\mathbf{S}' = \{x_1, \ldots, x_n, x_1, \ldots, x_n\}$ by concatenating $\mathbf{S}$ with itself. This objective encourages the model to attend to and retrieve long-range prefix information, enabling it to reconstruct the second half of the sequence.

- **Full HSA + Short SWA warm-up:** Setting top-k in HSA layers to 256 and sliding window size to 512 during the initial training phase.

All experiments were conducted on a 0.5B parameter dense model trained on 100B tokens with a pre-training context length of 16k. We incorporated 1% ruler-specific synthetic data into the pre-training data to facilitate evaluation using ruler benchmarks. Performance was evaluated based on the perplexity of the last 4k tokens on the PG19 dataset and accuracy on the Multi-key NIAH (MK-NIAH) task within the ruler benchmark.

The results in Table 2 demonstrated that the self-copy warm-up strategy yielded the best length extrapolation performance, albeit with some negative impact on in-domain performance. The full HSA + short SWA warm-up approach achieved a better balance, maintaining in-domain performance while delivering reasonable length extrapolation capabilities.

4.2 Scaling and Evaluation of Pretrained and Fine-tuned Models

For HSA-UltraLong, we developed two variants: a 0.5B dense model and an 8B MoE model with 1B activated parameters. We compared the MoE variant against a standard Transformer-based model (TRM-MoE) with similar parameter count—trained on identical data with matching hyperparameters. The architectures are largely consistent, with only one structural difference: HSA-UltraLong modifies the MoE configuration from 32-expert/2-activated to 64-expert/4-activated with halved expert dimensions. Additionally, HSA-UltraLong uses 16K pretraining context compared to TRM-MoE's 4K. For evaluation, we used the TRM-MoE's 8T-token checkpoint (pre-annealing).

Table 4: Comparison among HSA-UltraLong-Inst (HSA-UL-Inst) and Qwen3 (Non-thinking) after supervised fine-tuning. All models were evaluated under a unified framework for fair comparison.

	Qwen3-Inst	**HSA-UL-Inst**	**Qwen3-Inst**	**HSA-UL-Inst**
Architecture	Dense	Dense	Dense	MoE
# Total Params	0.6B	0.5B	1.7B	8B
# Activated Params	0.6B	0.5B	1.7B	1B
# Training Tokens	36T	4T	36T	8T
General Tasks				
BBH	42.56	26.25	59.48	57.25
MMLU	45.87	42.24	63.05	61.34
CMMLU	41.64	43.33	60.84	64.06
C-Eval	43.81	45.41	62.70	62.86
Math Tasks				
GSM8K	55.65	55.42	79.00	82.94
MATH	45.26	40.76	64.32	61.56
MATH500	53.00	41.00	73.20	71.00
OlympiadBench	16.89	8.74	36.30	27.85
Coding Tasks				
HumanEval	40.24	39.63	65.24	71.95
MBPP	29.20	34.40	51.00	57.00
HumanEval+	35.37	37.20	61.59	70.73
MBPP+	34.39	39.95	59.52	65.87
CRUX-O	28.00	23.25	50.00	50.75
Alignment Tasks				
IFEval Strict Prompt	55.08	33.09	64.33	63.22
AVG	40.50	36.48	60.76	62.03

We benchmarked the dense variant against Qwen 2.5-0.5B [44] and Qwen3-0.6B [45]. These comparison models have similar parameter counts but were trained on substantially larger datasets—4.5 times and 9 times our training data volume, respectively.

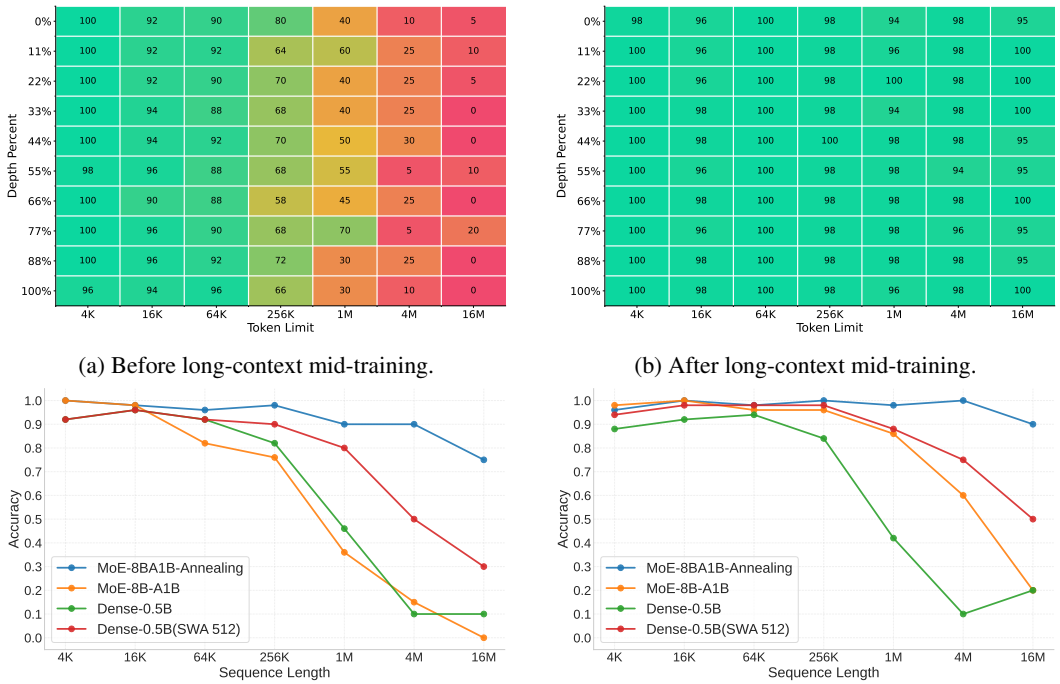
Our primary evaluation focused on assessing model performance within the pretraining context length across standard benchmarks. Results in Table 3 show that the HSA-UltraLong-MoE achieved parity with TRM-MoE in average performance scores, while the dense variant demonstrated only a 3.3-point deficit compared to Qwen 2.5-0.5B, despite having significantly less training data.

Additionally, we evaluated the MoE and Dense models after supervised fine-tuning. The results in Table 4 indicate that while a few general tasks showed no significant performance improvement after supervised fine-tuning, most tasks—particularly math and coding tasks—demonstrated substantial enhancements compared to the base models. Notably, our HSA-UltraLong-MoE achieved scores averaging 1.3 points higher than Qwen3-1.7B (Non-thinking), despite requiring fewer training flops. Similarly, our dense variant performed competitively, scoring only approximately 4 points below Qwen3-0.6B, despite being trained on a dataset merely one-ninth the size.

These findings demonstrate that HSA-UltraLong models maintain their capabilities within standard contexts while extending their effective context length to 16M tokens, further highlighting the superiority of our architectural approach.

4.3 Long-context Evaluation

During training, we randomly convert samples into RULER tasks with a 1% probability by inserting a "needle" in a long context and appending the Needle-in-a-Haystack (NIAH) prompt and answer at the end of the sample so the model can follow the NIAH instructions. This modification serves as a probe task to evaluate the model's extrapolation ability while having minimal impact on training. The results for RULER tasks are reported in Figure 4, we identify three key findings:



(a) Before long-context mid-training.

(b) After long-context mid-training.

(c) Multi-Query NIAH Task.

(d) Variable Tracking Task.

Figure 4: Evaluation of length generalization using the Needle-in-a-Haystack test. (a) and (b) present the results of the HSA-UltraLong-MoE before and after the long-context continued training phase on the Single-NIAH task at various depths. In (c) and (d), we evaluate the performance of different models on the Multi-Query NIAH Task (2 queries, 6 key-value pairs) and the Variable Tracking Task.

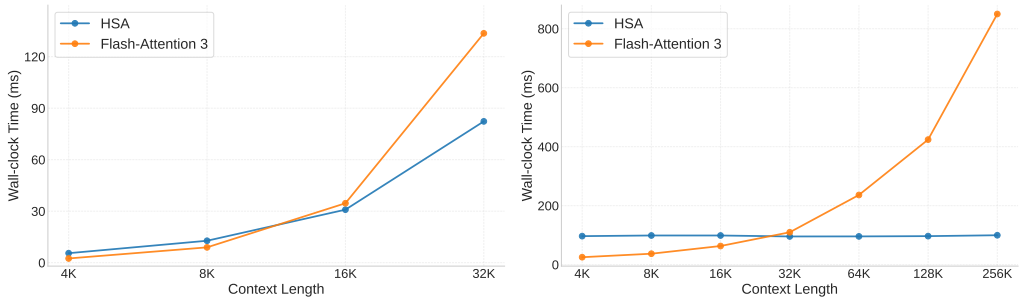
1. **Effective context length of training data is critical for HSA extrapolation.** As shown in Figure4(a), models pretrained on standard corpora exhibit a progressive decline in retrieval accuracy with longer contexts. This occurs despite a 16K pretraining context window, because the effective context length of the data is often much shorter. In contrast, training on data with longer effective contexts (>32K), as in Figure4(b), yields substantially improved extrapolation. This principle underpins the trends in Figures 4(c) and (d).

2. **A seesaw effect exists between HSA and Sliding Window Attention.** Figures 4(c) and (d) indicate that a smaller SWA window (512) during continued pretraining leads to better HSA extrapolation than a larger window (4K). Given that training from scratch with a 4K window fails to develop extrapolative HSA, we conclude that larger SWA windows impair HSA's long-range generalization. We posit that HSA learns a form of retrieval-based extrapolation. Large SWA windows handle most short-range dependencies inherently, reducing the incentive for HSA to learn them and thus weakening its ability to generalize to longer sequences.

3. **HSA capability scales with parameter size in reasoning-retrieval tasks.** While MoE-8B-A1B and Dense-0.5B exhibit comparable performance on the pure retrieval task (MQ-NIAH; Figure 4(c)), MoE-8B-A1B consistently outperforms Dense-0.5B in the variable-tracking task (Figure 4(d)), demonstrating that larger models better support joint reasoning and retrieval.

4.4 Training/Inference Efficiency

To further evaluate the efficiency of the sparse attention module, we benchmark the HSA operator against the FlashAttention-3 [31] operator on H800 for both training and inference, with HSA implemented using TileLang [39]. As shown in Figure 5, at shorter sequence lengths, FlashAttention-3 still leads in both training and inference, and HSA only gains an advantage with longer contexts. We attribute this to two factors: (1) the sparsity in HSA causes the kernel to incur more memory ac-



(a) Training efficiency.

(b) Inference efficiency.

Figure 5: Comparison of training/inference efficiency between HSA kernel and Flash-attention 3.

cesses compared to FlashAttention-3; and (2) FlashAttention-3 is implemented in CUDA, enabling it to better leverage the features of the Hopper architecture.

5 Conclusion

Although HSA has shown promising extrapolation capabilities, several challenges remain:

- The HSA/SWA seesaw problem. After training on short SFT data, extrapolation can degrade. The main reason is that an excessively long sliding-window attention reduces the need for HSA to learn short-range dependencies, which in turn hampers its ability to extrapolate to long-range dependencies.
- The head ratio constraint. HSA currently requires a 16:1 ratio of query heads to key–value heads, creating a severe information bottleneck. Future work should pursue kernel-level optimizations to alleviate this constraint.
- When sequences are short, training and inference show no clear advantage over FlashAttention-3; further kernel-level optimizations are needed to improve efficiency.

Despite these limitations, HSA-UltraLong presents a highly promising paradigm for long-context processing. The core insight of HSA is to *perform attention chunk by chunk and fuse the results via retrieval scores*, rather than selecting chunks and then concatenating them for attention. The experimental results provide a meaningful step toward effectively handling infinite-long context, advancing progress on long-term memory in machines.

References

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

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

[3] Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge. *CoRR*, abs/2102.03315, 2021. URL `https://arxiv.org/abs/2102.03315`.

[4] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6239. URL `https://doi.org/10.1609/aaai.v34i05.6239`.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, et al. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL `https://arxiv.org/abs/2107.03374`.

[7] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.

[8] Nelson Cowan. What are the differences between long-term, short-term, and working memory? *Progress in brain research*, 169:323–38, 2008. URL `https://api.semanticscholar.org/CorpusID:205921304`.

[9] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=ztn8FCR1td`.

[10] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, et al. Deepseek-v3 technical report, 2024.

[11] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, et al. Scaling vision transformers to 22 billion parameters, 2023. URL `https://arxiv.org/abs/2302.05442`.

[12] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752, 2023. doi: 10.48550/ARXIV.2312.00752. URL `https://doi.org/10.48550/arXiv.2312.00752`.

[13] Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. CruxEval: A Benchmark for Code Reasoning, Understanding and Execution. *arXiv preprint arXiv:2401.03065*, 2024.

[14] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. *arXiv preprint arXiv:2402.14008*, 2024.

[15] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[16] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving with the Math Dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[17] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

[18] Xiang Hu, Jiaqi Leng, Jun Zhao, Kewei Tu, and Wei Wu. Hardware-aligned hierarchical sparse attention for efficient long-term memory access. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=dIHSZTx9Lu.

[19] Xiang Hu, Zhihao Teng, Jun Zhao, Wei Wu, and Kewei Tu. Efficient length-generalizable attention via causal retrieval for long-context language modeling. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=6HVcoIbZoC.

[20] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, et al. C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models. *Advances in Neural Information Processing Systems*, 36:62991–63010, 2023.

[21] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020. URL http://proceedings.mlr.press/v119/katharopoulos20a.html.

[22] Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack, 2024. URL https://arxiv.org/abs/2406.10149.

[23] Jiaqi Leng, Xiang Hu, Junxiong Wang, Jianguo Li, Wei Wu, and Yucheng Lu. Understanding and improving length generalization in hierarchical sparse attention models, 2025. URL https://arxiv.org/abs/2510.17196.

[24] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring Massive Multitask Language Understanding in Chinese. *arXiv preprint arXiv:2306.09212*, 2023.

[25] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations*, 2023.

[26] Ling-Team, Ang Li, Ben Liu, Binbin Hu, Bing Li, Bingwei Zeng, Borui Ye, et al. Every activation boosted: Scaling general reasoner to 1 trillion open language foundation, 2025. URL https://arxiv.org/abs/2510.22115.

[27] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. *Advances in Neural Information Processing Systems*, 36:21558–21572, 2023.

[28] Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Yutao Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. MoBA: Mixture of block attention for long-context LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=RlqYCpTu1P.

[29] Amirkeivan Mohtashami and Martin Jaggi. Random-access infinite context length for transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=7eHn64wOVy.

[30] Ohad Rubin and Jonathan Berant. Retrieval-pretrained transformer: Long-range language modeling with self-retrieval. *Transactions of the Association for Computational Linguistics*, 12:1197–1213, 2024.

[31] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=tVConYid20.

[32] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

[33] Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

[34] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging Big-Bench Tasks and Whether Chain-of-Thought Can Solve Them. *arXiv preprint arXiv:2210.09261*, 2022.

[35] Ning Tao, Anthony Ventresque, Vivek Nallur, and Takfarinas Saber. Enhancing program synthesis with large language models using many-objective grammar-guided genetic programming. *Algorithms*, 17(7):287, 2024. doi: 10.3390/A17070287. URL https://doi.org/10.3390/a17070287.

[36] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[38] Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *CoRR*, abs/2408.15664, 2024. doi: 10.48550/ARXIV.2408.15664. URL https://doi.org/10.48550/arXiv.2408.15664.

[39] Lei Wang, Yu Cheng, Yining Shi, Zhengju Tang, Zhiwen Mo, Wenhao Xie, Lingxiao Ma, Yuqing Xia, Jilong Xue, Fan Yang, and Zhi Yang. Tilelang: A composable tiled programming model for ai systems, 2025. URL https://arxiv.org/abs/2504.17577.

[40] Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. CMATH: can your language model pass chinese elementary school math test? *CoRR*, abs/2306.16636, 2023. doi: 10.48550/ARXIV.2306.16636. URL https://doi.org/10.48550/arXiv.2306.16636.

[41] Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie Everett, et al. Small-scale proxies for large-scale transformer training instabilities, 2023.

[42] Haoyi Wu and Kewei Tu. Layer-condensed KV cache for efficient inference of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 11175–11188. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.602. URL https://doi.org/10.18653/v1/2024.acl-long.602.

[43] Haoyuan Wu, Haoxing Chen, Xiaodong Chen, Zhanchao Zhou, Tieyuan Chen, Yihong Zhuang, Guoshan Lu, Zenan Huang, Junbo Zhao, Lin Liu, et al. Grove moe: Towards efficient and superior moe llms with adjugate experts. *arXiv preprint arXiv:2508.07785*, 2025.

[44] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[45] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

[46] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

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

[48] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

[49] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 2299–2314. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-NAACL.149. URL https://doi.org/10.18653/v1/2024.findings-naacl.149.

[50] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models. *arXiv preprint arXiv:2311.07911*, 2023.