SCALING PARAMETER AND CONTEXT IN ATTENTION: NATIVE SPARSE ATTENTION FROM MIXTURE-OF-HEAD

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling attention parameters can improve language model quality, but retaining full token histories makes additional heads costly at long contexts. Furthermore, since attention retrieves and combines contextual information, parameter scaling should also support longer contexts. We therefore ask *whether attention parameter scaling can directly enable efficient and effective context scaling*. We introduce NAMOH, an architecture-native sparse attention mechanism that activates K of H heads per token. Each head retains only its assigned tokens and performs causal attention within this subsequence. Head selection thus jointly determines active parameters and available context without scanning the full history. Under balanced assignments, increasing H at fixed K shortens head histories and reduces per-token key-value (KV) access without increasing total KV storage. We further support head-relative rotary position embeddings to shorten positional spans within routed subsequences, aiming to mitigate position-induced attention noise. Experiments show that NAMOH can outperform fully activated models with the same total parameters, while enabling more efficient long-context inference than smaller dense models with matched active parameter counts. It remains compatible with GQA and existing sparse attention mechanisms. We hope this work offers a new path for scaling attention, with parameter scaling directly enabling context scaling.

1 INTRODUCTION

Scaling attention parameters through expert routing can improve the performance of large language models (LLMs) (Zhang et al., 2022; Yang et al., 2025). Methods such as SwitchHead and MoH further demonstrate gains in model quality and efficiency by selectively activating attention heads or projections (Csordás et al., 2023; Jin et al., 2024). Yet most parameter growth in frontier models has been concentrated in feed-forward networks (FFNs), particularly with the rise of mixture-of-experts (MoE) architectures (Jiang et al., 2024; Dubey et al., 2024; Adler et al., 2024; MiMo Team, 2026; Yang et al., 2024a). As shown in Figure 1(a), attention parameters have scaled much more slowly.

A key challenge is to increase attention parameters without proportionally increasing the cost of storing and processing context. When each head retains its own full token history, adding heads increases key-value (KV) storage, while activating more heads increases full-attention computation. This raises activation memory and compute costs during training and prefill, as well as KV cache storage and memory traffic during decoding (Dao, 2023; Tang et al., 2024). Yet efficiency is only part of the goal. Mechanistic studies show that FFNs can store knowledge acquired during training (Geva et al., 2020), while attention heads play a central role in retrieving and combining information from the current context (Olsson et al., 2022; Guo et al., 2024). This role makes long-context processing a natural objective for attention parameter scaling.

Despite this connection, work on context scaling has primarily focused on computational efficiency rather than attention parameter scaling. The expanding context windows in Figure 1(b) have been supported by sparse and linear-time mechanisms, often combined with full attention (Yuan et al., 2025; Yang et al., 2024b; DeepSeek-AI et al., 2026; Bai et al., 2026). Linear-time recurrent architectures compress history into fixed-size states, which can lose information needed for precise recall (Jelassi et al., 2024; Cabannes et al., 2026). Selection-based sparse attention preserves explicit token memories but introduces token or block selection overhead (Lu et al., 2025; Yuan et al., 2025). For indexers that scan the full prefix, this overhead grows with context length (Xu et al., 2026). Computational

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

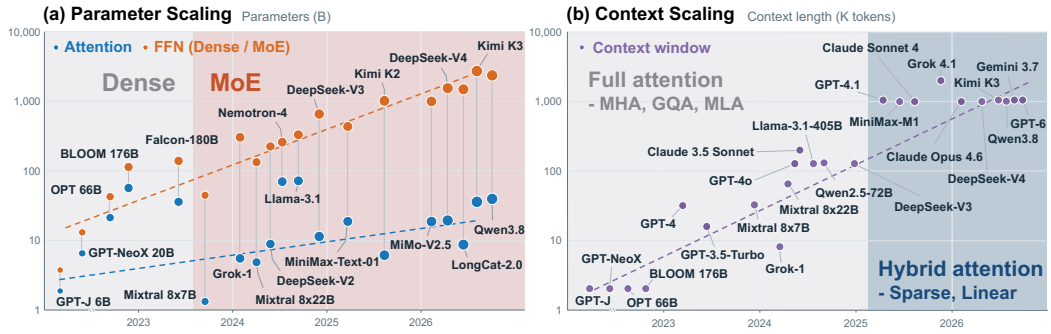

Figure 1: **Parameter and context scaling in frontier language models.** (a) Attention and FFN parameters in frontier models. Since the rise of MoE, parameter growth has been dominated by FFN experts, with comparatively little growth in attention. (b) Maximum context windows in frontier models. Sparse and linear-time mechanisms support further context expansion.

savings alone also leave positional limitations unresolved. As Du et al. (2026) show, long contexts can yield indistinguishable attention scores for different positions or tokens and disrupt token relevance rankings. Related geometric analysis links long RoPE spans to spurious query-key alignment, which can introduce attention noise by assigning weight to irrelevant tokens (Wertheimer et al., 2026).

Together, these challenges motivate a joint view of attention parameter and context scaling. The goal is not only to add attention parameters at a manageable cost, but also to use those parameters to support longer contexts. We therefore ask: ***Can scaling attention parameters directly enable efficient and effective context scaling?***

We introduce NAMOH, an architecture-native sparse attention mechanism that couples attention parameter and context scaling. Each attention head acts as an expert, and a learned router activates the K highest-scoring heads out of H total heads for each token. Routing determines both the active head parameters and the token subsequence processed by each head. Each head performs causal attention only within its routed subsequence, and the layer combines the projected head outputs using routing weights. Attention sparsity therefore arises directly from the routing structure, rather than masking a dense attention result.

We use load balancing to discourage head collapse (Jin et al., 2024; Fu et al., 2026) and distribute tokens approximately evenly across heads. With H total heads and K active heads per token, a context of L tokens requires LK key-value (KV) entries in total, with approximately LK/H entries per head. Each query token therefore accesses approximately LK^2/H KV entries across its K active heads. We define the *KV activation ratio* as this per-token KV access relative to a full-attention baseline. For baselines with the same model width and head dimension, this ratio is approximately K/H relative to K-head full attention matched in active attention parameters, and $(K/H)^2$ relative to H-head full attention matched in total attention parameters.

For example, with $H = 32$ and $K = 8$, total KV storage is $1/4$ that of 32-head full attention and equal to that of 8-head full attention. Under balanced routing, each head retains approximately $L/4$ entries, so eight active heads access approximately $2L$ entries per token. This yields KV activation ratios of $1/16$ and $1/4$ relative to the two baselines, respectively. Parameter scaling thus allows more heads to specialize in different contextual patterns, with each head processing its routed subsequence.

We further support *head-relative RoPE*, which encodes each token by its position within a head's routed subsequence rather than its global position. By shortening the encoded span to approximately LK/H under balanced routing while preserving token order, this design aims to mitigate position-induced attention noise. For efficient training and prefill, we pack head-specific subsequences for variable-length FlashAttention (Dao, 2023). During decoding, we execute only active head queries and read or update only their corresponding KV caches.

Our experiments show that NAMOH can achieve higher accuracy than fully activated models with the same total parameter count. Its reduced KV access also enables more efficient long-context inference than smaller dense models matched in active parameter count. Moreover, NAMOH is compatible with GQA and existing sparse attention mechanisms, which can further select tokens or blocks within each head's routed subsequence. Together, these properties offer a complementary path to context scaling through attention parameter scaling.

2 BACKGROUND

2.1 HEAD SPARSITY: MIXTURE-OF-HEAD

Mixture-of-experts (MoE) layers scale feed-forward capacity through conditional activation without proportional growth in per-token computation (Shazeer et al., 2017; Fedus et al., 2021). The same principle applies to attention parameters. Peng et al. (2020) mix overlapping groups of heads rather than route each token to individual heads. Mixture of Attention Heads (MoA) routes query and output projections with shared keys and values (Zhang et al., 2022), while SwitchHead routes value and output projections (Csordás et al., 2023). Grouped Query Experts (GQE) routes query heads within grouped-query attention (Tripathi & Kumar, 2026).

Mixture-of-Head attention (MoH) selects heads per token and weights their projected outputs (Jin et al., 2024). However, its heads retain full key-value (KV) histories even for tokens whose head outputs are inactive. With independent, fixed-width KV heads, cache storage grows with head count and context length, while each active query still processes the full prefix. Mixture of Sparse Attention (MoSA) introduces sequence sparsity through expert-choice routing, with each head selecting a fixed quota of top-scoring tokens from the full sequence (Piekos et al., 2025). This balances head loads, but future tokens can change earlier selections despite causal attention masking, so incremental autoregressive decoding requires routing changes.

2.2 CONTEXT SPARSITY: SPARSE ATTENTION

Selection-based sparse attention moves relevance filtering before the main softmax attention. Quest ranks KV pages using query-dependent scores from key metadata (Tang et al., 2024). Native Sparse Attention (NSA) reuses compressed attention scores to select blocks (Yuan et al., 2025), while Mixture of Block Attention (MoBA) selects blocks using query affinities to pooled keys (Lu et al., 2025). DeepSeek-V4 combines KV compression with a lightweight indexer in its compressed sparse attention branch (DeepSeek-AI et al., 2026). When indexers scan the prefix, their overhead grows with context length (Xu et al., 2026). Realizing speedups also requires specialized kernels and cache layouts for irregular KV access (DeepSeek-AI et al., 2026). NSA also reports higher average performance than its full-attention baseline on general and long-context benchmarks, supporting sparsity as a modeling choice (Yuan et al., 2025).

NAMOH provides a complementary source of sparsity through token-to-head assignments, without ranking the full history per query. Within routed subsequences, softmax attention remains compatible with dynamic token selection (Zhang et al., 2023; Tang et al., 2024) and static patterns such as sliding windows with retained sink tokens (Xiao et al., 2023).

Token selection does not necessarily shorten positional spans when original indices are retained. For rotary position embeddings (RoPE), Wertheimer et al. (2026) link length extrapolation to disrupted query-key separation and spurious attention to irrelevant tokens. Du et al. (2026) further identify indistinguishable scores across positions or tokens and reversed relevance rankings. Sparsity alone therefore does not ensure reliable scores among retained tokens. These findings motivate reducing the effective range of relative positions as a potential way to mitigate position-induced attention noise.

3 NAMOH: NATIVE SPARSE ATTENTION FROM MIXTURE-OF-HEAD

3.1 ROUTED CAUSAL ATTENTION

For a length-T sequence with token representations $x_t \in \mathbb{R}^{1\times d_{\text{model}}}$ and model width d_{model}, standard multi-head attention with H heads (Vaswani et al., 2017) can be written as

$$o_t^{\text{full}} = \sum_{i=0}^{H-1} h_{t,i}^{\text{full}} W_O^{(i)}, \tag{1}$$

where $h_{t,i}^{\text{full}} \in \mathbb{R}^{1\times d_{\text{head}}}$ is head i's causal attention output with head width d_{head}, and $W_O^{(i)} \in \mathbb{R}^{d_{\text{head}}\times d_{\text{model}}}$ is its slice of the output projection.

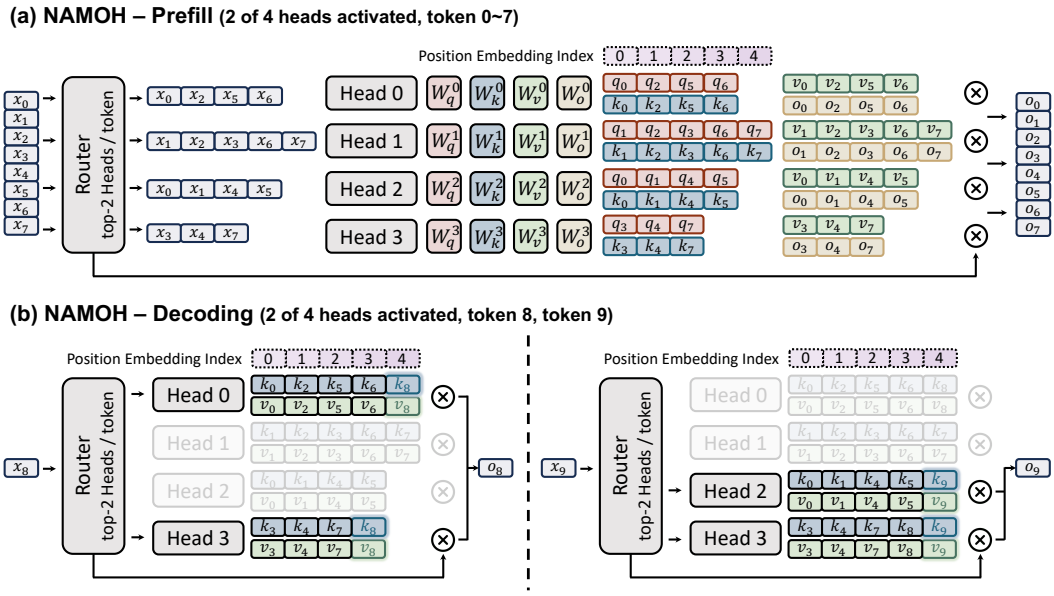

Figure 2: **NAMOH with two active heads out of four.** (a) Prefill routes tokens 0 to 7 before QKV projection. Each head attends within its ordered subsequence, and projected outputs are combined with routing weights. (b) During decoding, tokens 8 and 9 access and extend only their selected heads' KV caches. Position indices follow each head's local token order.

Token-to-head routing. Following the expert view of attention heads (Jin et al., 2024), NAMOH activates K of H heads per token, where $1 \leq K \leq H$. A learned router computes head affinities, selections, and gates as

$$a_t = \mathrm{softmax}(x_t W_r), \quad \mathcal{S}_K(t) = \mathrm{TopK}_i(a_{t,i}), \quad m_{t,i} = \mathbf{1}\{i \in \mathcal{S}_K(t)\}, \quad g_{t,i} = m_{t,i} a_{t,i}. \tag{2}$$

Here, $W_r \in \mathbb{R}^{d_{\text{model}} \times H}$ is the router matrix, and $a_t \in \mathbb{R}^{1 \times H}$ contains affinities normalized by softmax across heads. TopK returns the indices of the K largest affinities, and $\mathbf{1}\{\cdot\}$ is the indicator function. The selection mask $m_{t,i}$ sets inactive gates to zero.

Sparse projection and attention. Routing determines both which parameters a token uses and which head histories it enters. For head i, define the ordered index set $\mathcal{I}_i = \{t : m_{t,i} = 1\}$ and its length $n_i = |\mathcal{I}_i|$. Tokens are dispatched to these subsequences before projection. Only an active token-head pair computes

$$q_{t,i} = x_t W_Q^{(i)}, \qquad k_{t,i} = x_t W_K^{(i)}, \qquad v_{t,i} = x_t W_V^{(i)}, \qquad t \in \mathcal{I}_i. \tag{3}$$

Each head has independent query, key, and value matrices $W_Q^{(i)}, W_K^{(i)}, W_V^{(i)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{head}}}$. The resulting queries, keys, and values lie in $\mathbb{R}^{1 \times d_{\text{head}}}$. Inactive token-head pairs generate no QKV vectors and occupy no KV cache entries.

Let $\widetilde{q}_{t,i}$ and $\widetilde{k}_{t,i}$ denote queries and keys after positional encoding, described in Section 3.2. For $i \in \mathcal{S}_K(t)$, attention and output aggregation are

$$h_{t,i} = \mathrm{Attn}\Big(\widetilde{q}_{t,i}, [\widetilde{k}_{s,i}], [v_{s,i}]\Big), s \in \mathcal{I}_i, s \leq t, \qquad o_t = \sum_{i \in \mathcal{S}_K(t)} g_{t,i} h_{t,i} W_O^{(i)}, \tag{4}$$

where Attn denotes standard scaled dot-product attention, $h_{t,i} \in \mathbb{R}^{1 \times d_{\text{head}}}$, $W_O^{(i)} \in \mathbb{R}^{d_{\text{head}} \times d_{\text{model}}}$, and $o_t \in \mathbb{R}^{1 \times d_{\text{model}}}$. Each query therefore attends only to earlier tokens and itself within the same routed head. Sparsity is part of the computation, not a mask applied after dense attention.

Figure 2(a) illustrates dispatch and aggregation with $H = 4$ and $K = 2$. During decoding in Figure 2(b), token 8 selects heads 0 and 3, while token 9 selects heads 2 and 3. Each token appends to and attends within only its selected caches. Because routing uses the current token representation, prefill and incremental decoding implement the same causal computation.

Table 1: **Attention weight and KV cache costs.** Total and active projection weights, KV storage after T tokens, and historical KV activation for single-token decoding.

Cost	MHA	GQA	SA	MoH	**NAMOH**	GQA +SA	GQA **+NAMOH**	SA **+NAMOH**	GQA+SA **+NAMOH**
Total weights	$4Hd_md_h$	$2Hd_md_h\frac{g+1}{g}$	$4Hd_md_h$	$4Hd_md_h$	$\boldsymbol{4Hd_md_h}$	$2Hd_md_h\frac{g+1}{g}$	$2Hd_md_h\frac{g+1}{g}$	$4Hd_md_h$	$2Hd_md_h\frac{g+1}{g}$
Active weights	$4Hd_md_h$	$2Hd_md_h\frac{g+1}{g}$	$4Hd_md_h$	$4Hd_md_h$	$\boldsymbol{4Kd_md_h}$	$2Hd_md_h\frac{g+1}{g}$	$2Kd_md_h\frac{g+1}{g}$	$4Kd_md_h$	$2Kd_md_h\frac{g+1}{g}$
KV storage	HT	HT/g	HT	HT	$\boldsymbol{KT}$	HT/g	KT/g	KT	KT/g
KV activation	HT	HT/g	αHT	KT	$\boldsymbol{TK^2/H}$	$\alpha HT/g$	$TK^2/(Hg)$	$\alpha TK^2/H$	$\alpha TK^2/(Hg)$

3.2 Load Balancing and Scaling Properties

Load balancing. Balancing encourages all heads to receive training signals and discourages head collapse (Fu et al., 2026). It also supports context sparsity: if routing concentrates on a fixed subset of heads, their histories can become dense despite sparse head activation. For a training batch $\mathcal{B}$ containing N tokens, define the assignment fraction f_i and average head affinity p_i as

$$f_i = \frac{1}{NK}\sum_{t\in\mathcal{B}} m_{t,i}, \qquad p_i = \frac{1}{N}\sum_{t\in\mathcal{B}} a_{t,i}, \tag{5}$$

with batch indices suppressed. Both distributions sum to one, with uniform targets $f_i = p_i = 1/H$.

We consider three alternatives from MoE: CV-based importance regularization (Shazeer et al., 2017), Switch-style (fp) balancing (Fedus et al., 2021), and auxiliary-loss-free balancing (Wang et al., 2024). For the two auxiliary-loss methods, the training objective is $\mathcal{L} = \mathcal{L}_{\mathrm{LM}} + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}}$, where $\mathcal{L}_{\mathrm{LM}}$ is the language-modeling loss, $\mathcal{L}_{\mathrm{bal}}$ is the chosen balancing loss summed across routed attention layers, and $\lambda_{\mathrm{bal}} > 0$ controls its strength. Loss-free balancing instead adjusts head-specific routing biases with an update rate $\eta > 0$, without adding a balancing loss. Appendix A details all three strategies.

Routing as context selection. Each head maintains its own routed history, so selecting heads also selects the contexts available to a query. The head router thus serves as a learned indexer over histories without rescoring historical keys or blocks. Unlike a fixed KV budget, the available history grows with sequence length, reaching approximately TK/H entries in each head under balanced routing. This selection acts at the head level and remains compatible with further token or block sparsity within each subsequence (Tang et al., 2024; Yuan et al., 2025; Lu et al., 2025).

Parameter and context costs in attention. We compare one attention layer with H query heads at fixed model and head widths, abbreviated as d_m and d_h. Let M_{KV} count stored KV pairs after a prefix of T tokens, and let $A_{\mathrm{KV}}(T)$ count distinct historical pairs used to decode the next token. Each pair contains $2d_h$ scalars; shared pairs are counted once. GQA shares one KV head among g query heads, reducing both counts to HT/g (Ainslie et al., 2023). Block-selected sparse attention (SA) retains the full cache but accesses approximately an $\alpha \in (0, 1]$ fraction of each history (Tang et al., 2024; Lu et al., 2025). Thus, SA stores HT pairs and activates approximately αHT. MoH instead retains all head histories and activates KT pairs through its K selected heads (Jin et al., 2024).

For NAMOH, let n_i denote head i's prefix length and $\mathcal{S}_K(T)$ the next token's K selected heads. Routing determines both cache insertion and access:

$$M_{\mathrm{KV}} = TK, \qquad A_{\mathrm{KV}}(T) = \sum_{i\in\mathcal{S}_K(T)} n_i \approx \frac{TK^2}{H}. \tag{6}$$

Storage is exact, while activation assumes approximately balanced histories. Relative to H-head MHA, these costs are reduced to K/H and approximately $(K/H)^2$, respectively. Relative to K-head MHA, which matches active projection parameters, storage is unchanged and activation is reduced to approximately K/H.

Table 1 compares attention projection weights and KV cache costs. Router weights are omitted, which add $\mathcal{O}(Hd_m)$ always-active parameters for head routing. The mechanisms are complementary: GQA shares KV representations, NAMOH shortens routed histories, and SA selects blocks within those histories. With KV-group routing and group-shared block selection, combining all three stores TK/g KV pairs and activates approximately $\alpha TK^2/(Hg)$ historical pairs per decoding token. Appendix B details the combination rules, selection overhead, and prefill computation.

Head-relative positional encoding. We optionally apply RoPE using a token's rank within its routed head rather than its global position. For an active token-head pair, its zero-based rank $\rho_i(t)$ and position-encoded vectors are

$$\rho_i(t) = \sum_{s=0}^{t} m_{s,i} - 1, \qquad \widetilde{q}_{t,i} = q_{t,i} R(\rho_i(t)), \qquad \widetilde{k}_{t,i} = k_{t,i} R(\rho_i(t)), \tag{7}$$

where $R(p) \in \mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$ is the RoPE rotation for row vectors at position p. For example, head 0 in Figure 2(a) maps global positions $(0, 2, 5, 6)$ to local indices $(0, 1, 2, 3)$. Its next activated token, token 8, receives index 4. The largest index in head i is $n_i - 1$, so balanced routing contracts the encoded span from T positions to approximately TK/H. This preserves token order, but not original token distances. The shorter span is intended to improve query-key matching by limiting position-induced attention noise (Wertheimer et al., 2026; Du et al., 2026), complementing the computational benefit of sparsity. Using global-position RoPE instead amounts to replacing $\rho_i(t)$ with t.

Efficient execution. Training and prefill pack each $(\text{batch}, \text{head})$ subsequence as an independent sequence for variable-length FlashAttention (Dao, 2023). The total packed length is fixed at TK per sequence and NK per training batch, although individual head lengths vary. Load balancing helps limit this length skew. We further use length-aware scheduling to interleave query tiles with larger and smaller causal workloads across subsequences. This balances cumulative work across GPU multiprocessors to reduce tail idle time, without introducing cross-subsequence attention. Decoding launches only active head queries and reads or updates only their corresponding caches.

4 Experiments

4.1 Experimental Setup

Models and training. We train models from scratch with 0.6B to 1.2B parameters, covering multi-head attention (MHA), grouped-query attention (GQA), selection-based sparse attention (SA), and NAMOH. We fix the number of layers at 16, the model width at 2048, and the head width at 64, with head counts of 4, 8, 16, and 32. Every model receives 24B tokens from FineWeb-Edu (Penedo et al., 2024). This budget corresponds to 20 training tokens per parameter of the largest model, following the Chinchilla scaling guideline (Hoffmann et al., 2022), and is kept identical across models. For long-context evaluation, we further post-train models on LongAlign (Bai et al., 2024).

We use AdamW with a learning rate of 0.002, followed by linear decay to zero over the final 20% of training. Auxiliary-loss balancing uses a coefficient of $\lambda_{\text{bal}} = 0.001$, while loss-free balancing uses a routing-bias update rate of $\eta = 0.001$. Other optimizer hyperparameters follow the implementation defaults of the AdamW optimizer. All experiments are conducted on NVIDIA A100 GPUs.

Baselines and controls. MHA-H and SA-H use H heads, GQA-32KVK uses 32 query heads and K KV heads, and NAMOH-HAK activates K of H heads per token. Our SA baseline follows the block-selection paradigm (Lu et al., 2025; Yuan et al., 2025). Each head retains its full KV history, represents each KV block by its mean-pooled keys for selection scoring, and selects a fixed fraction of the highest-scoring blocks. We use 32-token blocks, reuse selected indices across groups of 32 queries, and additionally maintain a 128-token sliding window. The fraction in parentheses specifies the per-head KV activation budget. KV storage and activation are normalized to MHA-32. Activation counts entries accessed by the main attention operation, with shared GQA entries counted once. To isolate the gains from routed context sparsity, we equip all MHA, GQA, and SA baselines with the head gating and load-balancing mechanisms used in prior head-routing methods (Fu et al., 2026; Jin et al., 2024; Qiu et al., 2025), with matched settings across comparisons.

Evaluation. We evaluate pretrained models on MMLU (Hendrycks et al., 2020) for general knowledge, GSM8K (Cobbe et al., 2021) for mathematics, HumanEval (Chen et al., 2021) for coding, and BoolQ (Clark et al., 2019) for reading comprehension. Scientific reasoning benchmarks include ARC-Easy, ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). We assess commonsense reasoning with HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2019), and WinoGrande (Sakaguchi et al., 2019). For long-context understanding, we evaluate models on eight LongBench tasks (Bai et al., 2023): HotpotQA (Yang et al., 2018), Qasper (Dasigi et al., 2021), TriviaQA (Joshi et al., 2017), NarrativeQA (Kociský et al., 2017), 2WikiMultiHopQA (Ho et al., 2020), GovReport (Huang et al., 2021), QMSum (Zhong et al., 2021), and TREC (Li & Roth, 2002).

Table 2: Evaluation results of models trained from scratch with different attention mechanisms and KV budgets. KV storage (KV Stor.) and activation (KV Act.) are normalized to MHA-32. For sparse attention (SA), the fraction in parentheses denotes the activated KV fraction per head.

Attention Mechanism	KV Stor.	KV Act.	MMLU	GSM8K	HEval	ARC-E	ARC-C	HellaSwag	PIQA	OBQA	BoolQ	WinoG.	Average
MHA-32	1	1	32.79	5.31	6.10	65.87	34.13	45.88	69.37	37.20	46.64	53.99	39.73
MHA-16	1/2	1/2	32.35	3.11	4.27	63.76	34.39	46.84	70.08	35.80	46.18	52.49	38.93
GQA-32KV16	1/2	1/2	27.48	4.62	7.32	65.61	35.58	47.56	69.48	36.00	48.40	52.01	39.41
SA-32 (1/4)	1	1/4	25.27	2.58	3.66	60.87	33.21	45.20	59.90	37.60	44.22	50.12	36.26
SA-16 (1/2)	1/2	1/4	24.36	2.88	3.05	63.22	34.74	45.00	59.36	35.60	53.39	49.49	37.11
NAMOH-32A16	1/2	1/4	32.74	5.84	9.76	66.04	36.43	48.11	70.67	36.60	49.33	55.81	41.13
MHA-8	1/4	1/4	26.01	2.58	3.66	66.04	33.55	46.02	69.70	35.20	44.28	50.80	37.78
GQA-32KV8	1/4	1/4	25.35	2.88	6.10	63.72	32.70	45.49	69.59	34.60	48.80	52.75	38.20
SA-32 (1/16)	1	1/16	25.12	1.90	1.83	61.25	31.84	42.36	58.60	26.00	55.17	50.59	35.46
SA-8 (1/4)	1/4	1/16	25.11	1.44	3.05	61.96	35.32	43.60	58.87	29.60	51.90	50.36	36.12
NAMOH-32A8	1/4	1/16	31.31	4.62	7.93	64.86	36.26	47.78	69.26	35.80	55.38	51.54	40.47
MHA-4	1/8	1/8	25.54	1.36	3.05	58.21	30.46	47.60	70.29	36.20	44.60	53.43	37.07
GQA-32KV4	1/8	1/8	25.83	2.27	3.66	64.86	35.41	43.93	69.21	36.00	49.78	55.25	38.62
SA-32 (1/64)	1	1/64	24.47	0.53	0.61	47.54	23.04	37.69	58.05	28.20	54.43	49.09	32.37
SA-4 (1/8)	1/8	1/64	25.27	0.45	1.22	50.99	25.77	38.10	58.54	31.20	54.04	49.17	33.48
NAMOH-32A4	1/8	1/64	27.48	2.50	4.88	64.52	35.14	45.99	66.92	35.20	56.18	54.43	39.32

Table 3: Evaluation results on LongBench. KV stor. and act. are normalized to MHA-32; SA parentheses denote the KV fraction activated.

Method / Dataset	MHA 32	MHA 8	GQA 32KV8	SA-32 (1/16)	SA-8 (1/4)	NAMOH 32A8
KV Stor.	1	1/4	1/4	1	1/4	1/4
KV Act.	1	1/4	1/4	1/16	1/16	1/16
HotpotQA	27.52	23.89	**31.94**	21.89	24.13	30.23
Qasper	21.29	20.92	19.49	18.68	17.60	**22.58**
TriviaQA	**56.32**	35.21	44.97	29.17	34.60	46.82
NarrativeQA	11.14	12.04	9.93	9.94	11.06	**13.35**
2WikiMQA	28.83	30.48	31.87	26.69	27.09	**34.34**
GovReport	**18.07**	15.33	15.88	16.02	16.02	16.73
QMSum	22.41	20.60	22.61	20.80	21.65	**22.81**
TREC	71.50	69.50	70.50	67.00	68.50	**75.00**
Average	32.14	28.50	30.90	26.27	27.58	**32.73**

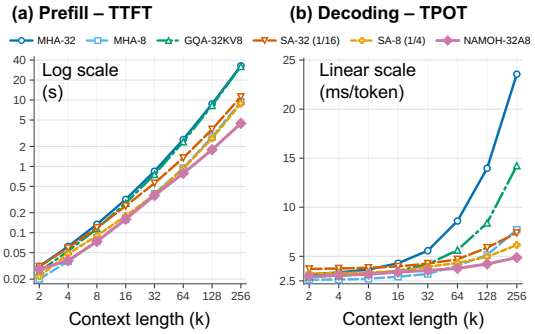

Figure 3: Inference efficiency. (a) Prefill TTFT and (b) decoding TPOT across attention mechanisms at different context lengths. Lower is better.

4.2 Model Quality and Inference Efficiency

General capabilities. Table 2 compares NAMOH-32AK with MHA, GQA, and SA for $K \in \{4, 8, 16\}$. All variants in this table use CV-based importance regularization for load balancing, with details provided in Appendix A. Across these settings, NAMOH achieves stronger overall performance than the corresponding K-head MHA and SA models and compares favorably with GQA. Its performance can also match or exceed MHA-32 despite sparse head activation. These results show that a larger pool of selectively activated heads can preserve the quality of a fully activated model while using fewer head parameters per token.

Long-context quality and KV budgets. Table 3 extends the comparison to LongBench with a maximum context length of 32K tokens, where NAMOH-32A8 improves overall performance over the MHA, GQA, and SA baselines. Under balanced routing, its normalized KV storage is $1/4$ and its KV activation is approximately $1/16$. MHA-8 and GQA-32KV8 match its storage but access approximately four times as many KV entries. SA-32 $(1/16)$ matches its activation budget but requires four times the storage, while SA-8 $(1/4)$ matches both.

Inference efficiency. Figure 3 reports time to first token (TTFT) and time per output token (TPOT) across context lengths. TTFT measures the latency to process the prompt and produce the first output token. TPOT measures the average latency of subsequent decoding steps. For the SA baselines, we use the implementation from MoBA (Lu et al., 2025). At long contexts, NAMOH-32A8 achieves lower TTFT and TPOT than MHA-32, GQA-32KV8, and SA variants, and even outperforms MHA-8.

These gains are consistent with the different bottlenecks of decoding and prefill. Long-context decoding is typically memory-bound, so reducing KV traffic helps lower TPOT (Yuan et al., 2025). Compute-bound prefill benefits from fewer query-key interactions and sparse projection. GQA

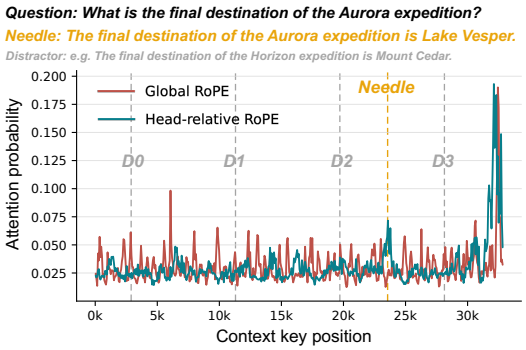

Figure 4: Full-question attention over a 32k-token haystack in NAMOH-32A8. Dashed lines mark the needle and four distractors.

Table 4: LongBench evaluation of NAMOH-32A8 with different RoPE variants across different context lengths. G and HR denote global RoPE and head-relative RoPE.

Length	8k		16k		32k	
Method	**G**	**HR**	**G**	**HR**	**G**	**HR**
HotpotQA	24.98	**27.46**	25.65	**29.43**	26.69	**30.23**
Qasper	**19.88**	18.21	20.41	**24.12**	20.03	**22.58**
TriviaQA	**45.75**	45.11	46.55	**50.10**	**47.63**	46.82
NarrativeQA	9.11	**12.16**	13.26	**15.28**	9.65	**13.35**
2WikiMQA	28.53	**30.54**	31.90	**32.48**	31.64	**34.34**
GovReport	16.60	**18.13**	**15.65**	15.50	**17.12**	16.73
QMSum	21.18	**21.49**	**23.52**	22.27	22.60	**22.81**
TREC	67.00	**68.00**	**70.50**	**70.50**	73.00	**75.00**
Average	29.13	**30.14**	30.93	**32.46**	31.05	**32.73**

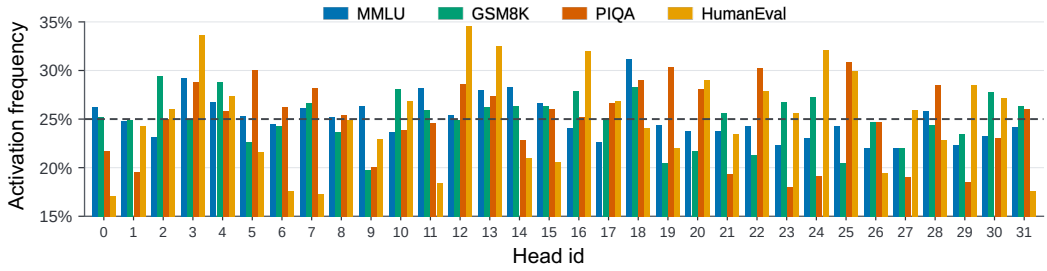

Figure 5: Task-dependent head utilization in NAMOH-32A8. Activation frequencies of individual heads in one attention layer on MMLU, GSM8K, PIQA, and HumanEval.

reduces KV projection costs but retains full-prefix attention for all 32 query heads. SA reduces the attended KV set but still incurs block-scoring and selection overhead (Lu et al., 2025; Yuan et al., 2025). In contrast, NAMOH performs attention over shorter packed subsequences without a history-scanning indexer, which also benefits comparisons at matched KV activation.

4.3 Positional Encoding and Routing Behavior

Head-relative RoPE. Figure 4 compares global and head-relative RoPE in NAMOH-32A8 on a 32k-token needle-in-a-haystack example with four distractors. The visualization aggregates attention from all question tokens to each haystack position. In this example, head-relative RoPE assigns more attention to the needle and less to irrelevant positions. Table 4 provides a broader comparison at context lengths of 8k, 16k, and 32k. Head-relative RoPE achieves higher LongBench scores, with the average gap increasing at longer contexts. These observations are consistent with the intended benefit of shortening the encoded positional span, as discussed in Section 3.2.

Head utilization across tasks. Figure 5 shows head activation frequencies on four datasets. Frequencies cluster around the balanced rate of $K/H = 25\%$, indicating broad head utilization rather than concentration on a small fixed subset. At the same time, individual heads exhibit clear frequency differences across tasks. This pattern suggests task-dependent specialization.

Routing-induced attention structure. Figure 6 examines a 32-token example. Panel (a) shows the heads selected by each token, with lines linking tokens assigned to the same head. For this sequence of length $T = 32$, let $M \in \{0,1\}^{T\times H}$ collect the routing masks, with $M_{t,i} = m_{t,i}$. Panel (b) visualizes $R = \mathrm{tril}(MM^\top)$, where tril retains the lower triangle, including the diagonal. Thus, $R_{t,s}$ counts the active heads shared by query token t and an earlier or current token s.

Panel (c) shows the gate-weighted attention map $A_{t,s} = \sum_{i=0}^{H-1} g_{t,i} m_{s,i} \alpha_{t,s}^{(i)}$. Here, $\alpha_{t,s}^{(i)}$ is the softmax attention weight from token t to token s within head i, defined as zero outside that head's routed causal pairs. Panels (b) and (c) therefore distinguish available connections from the attention weights assigned to them. Panel (d) displays individual head maps at the original token positions. Their separated support reflects the different routed subsequences, with non-routed positions absent from each head's computation.

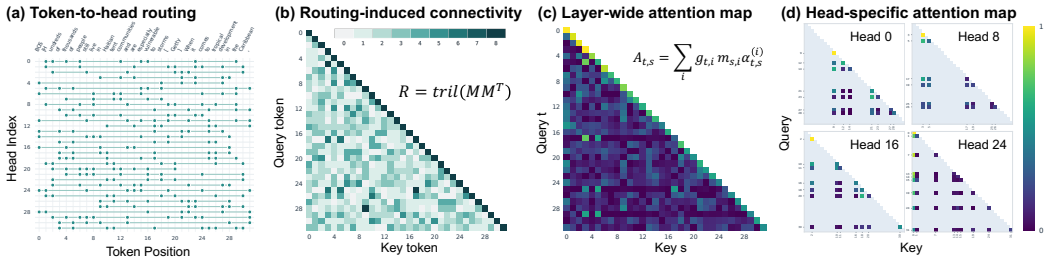

Figure 6: Routing shapes attention connectivity in NAMOH. (a) Token-to-head assignments. (b) The number of shared active heads for each causal token pair. (c) Layer-wide attention weighted by query-specific routing gates. (d) Attention maps for heads 0, 8, 16, and 24.

Table 5: Further refinement of NAMOH-32A8 with load-balancing strategies and shared heads.

Method	MMLU	GSM8K	HEval	ARC-E	ARC-C	HellaSwag	PIQA	OBQA	BoolQ	WinoG.	Average
Load Balancing Variants											
CV (Shazeer et al., 2017)	31.31	4.62	7.93	64.86	36.26	47.78	69.26	35.80	55.38	51.54	40.47
fp (Fedus et al., 2021)	32.40	5.31	9.76	64.66	37.11	46.16	70.40	36.80	56.81	52.96	41.24
Loss-Free (Wang et al., 2024)	33.23	3.11	7.93	62.76	34.23	45.12	68.23	35.00	55.63	52.96	39.82
Shared-Head Variants											
+ 2 Shared (Full)	30.99	5.38	8.64	64.10	37.64	47.12	70.97	37.00	58.31	53.28	41.34
+ 2 Shared (Sliding)	32.21	5.53	9.76	65.19	37.21	47.51	70.88	36.80	57.02	52.33	41.44
+ 4 Shared (Full)	33.14	6.22	10.37	66.12	38.24	47.37	71.24	37.60	58.63	53.75	42.27
+ 4 Shared (Sliding)	33.50	5.84	11.59	66.33	38.75	48.74	72.06	38.40	59.86	54.22	42.93

4.4 Load Balancing and Shared Heads

Load-balancing strategies. Table 5 compares CV-based importance regularization (Shazeer et al., 2017), Switch-style (fp) balancing (Fedus et al., 2021), and loss-free balancing (Wang et al., 2024). The formulations of these load-balancing strategies are detailed in Appendix A. All variants retain 32 routed heads and activate eight per token. At the settings specified above, fp achieves the best overall performance among the three alternatives.

Shared heads for local context. Inspired by shared expert isolation in DeepSeekMoE (Dai et al., 2024), we augment NAMOH-32A8 with H_s always-active shared heads, testing $H_s \in \{2, 4\}$. Each shared head attends either to the full causal prefix or to the most recent $w = 128$ original tokens, including the current token, where w is the sliding-window size. Their independently projected outputs are added to the routed output o_t. Shared heads are excluded from H, K, and the routing balance statistics. The sliding-window heads use global RoPE, while routed heads retain head-relative RoPE. This supplies local context even when neighboring tokens select different routed heads.

Shared heads add $4H_s d_{\text{model}} d_{\text{head}}$ always-active projection parameters. For a fixed window size w, the sliding variant requires at most $H_s w$ additional KV pairs per layer and $\mathcal{O}(TH_s w d_{\text{head}})$ attention work for a length-T sequence. Table 5 shows that adding shared heads improves overall performance. Sliding-window variants perform similarly to their full-attention counterparts, suggesting that reliable local context accounts for much of the benefit. This supports a complementary design in which shared heads cover nearby tokens while routed heads can focus on broader context retrieval.

5 Conclusion

We introduced NAMOH, an architecture-native sparse attention mechanism that connects attention parameter scaling with context scaling. Each token activates a subset of heads, and each head stores and attends only to its assigned tokens. Routing thus jointly selects active parameters and available context without scanning the full history. Under balanced routing, expanding the head pool at a fixed active head count shortens head histories. This reduces per-token KV access without increasing total KV storage. Head-relative RoPE further shortens positional spans and improves long-context quality in our evaluations. Experiments show that NAMOH can outperform fully activated models with the same total parameter count. It also enables faster long-context prefill and decoding than smaller dense models matched in active parameter count. The design remains compatible with GQA and existing sparse attention mechanisms. Together, these findings support a complementary path for scaling attention, in which parameter growth directly enables more efficient and effective context scaling.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity and readability of the manuscript and to assist with code implementation. The research ideas, methodology, and core software design were developed by the authors. We reviewed the manuscript for accuracy and verified the code for correctness, including all AI-assisted contributions. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Nvidia Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzek, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Long, Ameya Sunil Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martínez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong T. Nguyen, Osvald Nitski, Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhumoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Scowcroft, Jason D. Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, David Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivian Zhang, Yian Zhang, and Chen Zhu. Nemotron-4 340b technical report. *ArXiv*, abs/2406.11704, 2024.

Joshua Ainslie, James P. Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit K. Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *ArXiv*, abs/2305.13245, 2023. URL https://api.semanticscholar.org/CorpusID:258833177.

Kimi Team Yifan Bai, Yifan Bai, Yiping Bao, Chandru. M, Jianfeng Cai, Xin-Hao Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xinxian Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjian Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Dazhi Cheng, Yean Cheng, Jialei Cui, Ji-Zhe Cui, An-Bang Dai, Jiaqi Deng, Haochen Ding, Rui Ding, Shaofeng Ding, Mengfan Dong, Meng xiao Dong, Yuhao Dong, Yuxin Dong, Angang Du, Chenzhuang Du, Dikang Du, Jusen Du, Yulun Du, Yu Fan, Jing Feng, Qiulin Feng, Yichen Feng, Kelin Fu, Qiang Fu, Fu-Juan Gao, Hongcheng Gao, Jingyue Gao, Tong Gao, Wei Gao, Shangyi Geng, Jie Gong, Lin Gong, Sheng Gong, Xiaocheng Gong, Qizheng Gu, Yicheng Gu, Shuhao Guan, Haiqing Guo, Shiqi Guo, Xiang Guo, Zhen Guo, Beixi Hao, Wenxing Hao, Xiaoru Hao, Dailan He, Haotian He, Lehan He, Qi He, Weiran He, Xinran He, Xin-Yi He, Yibo He, Yun He, Chao Hong, Tiange Hong, Hao-Xing Hu, Jiaxi Hu, Rui-Jia Hu, Weiming Hu, Yangyang Hu, Zhenxing Hu, Liang Hua, Jinbin Huang, Ke Huang, Ruiyuan Huang, Siying Huang, Weixiao Huang, Yan Huang, Zhengjie Huang, Zhiqi Huang, Yulong Hui, Chao Jia, Yutong Jiang, Zhejun Jiang, Zuoyou Jiang, W.M. Jin, Xinyi Jin, Yu Jing, Huanjun Kong, Guokun Lai, Aidi Li, Cheng Li, Chengyu Li, Cong Li, Fang Li, Guanyu Li, Haoyang Li, Jia Li, Junxiong Li, Lei Li, Letian Li, Lincan Li, Weihong Li, Wentao Li, Xintong Li, Yang Li, Yisheng Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxin Li, Zheming Li, Zheng-Xiao Li, Zhiyuan Li, Jia-Wei Lin, Xiaohan Lin, Yibo Lin, Zichao Lin, Ziyan Lin, Bill Liu, B. Liu, Chuangye Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yipeng Liu, Zhengying Liu, Zhiheng Liu, Enzhe Lu, Haoyu Lu, Lin Lu, Ting Lu, Zhiyuan Lu, Aotian Luo, Gen Luo, Junyu Luo, Yifan Luo, Bohan Lyu, Wenzhou Lyu, Shaoguang Mao, Yuan Mei, Xin Men, Mi Ni, Yixuan Niu, Siyuan Pan, Shujun Peng, Zhangyang Qi, Ruoyu Qin, ZeChao Qin, Zeyu Qin, Haiquan Qiu, Jian Qiu, Jiezhong Qiu, Bowen Qu, Yuhao Qu, Zeyu Shang, Youbo Shao, Han Shen, Jin Shi, Juanfeng Shi, Li-Na Shi, Sheng-Peng Shi, Wingchun Siu, Pengwei Song, Xiaoxiu Song, Jianlin Su, Yu-Li Su, Zhaochen Su, Lin Sui, Jingsong Sun, Junyao Sun, Shaoning Sun, Shuzhen Sun, Tong Sun, Yujun Sun, Yu-Chuan Tai, Chuning Tang, He-Yi Tang, Si-Si Tang, Ze-Cheng Tang, Chao Tian, Rong Tian, Yu Tian, Wei Tu, Chensi Wang, Chuang Wang, Chun

jia Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Huaqing Wang, Hui Wang, Jiayi Wang, Jing-Lei Wang, Jinhong Wang, Jiuzheng Wang, Linian Wang, Shaobo Wang, Shenzhi Wang, Shuyi Wang, Siyuan Wang, Siyuan Wang, Tianfu Wang, Wenju Wang, Xingran Wang, Xin-Meng Wang, Xinyuan Wang, Xusheng Wang, Yalin Wang, Yang-Xi Wang, Yao Wang, Yaoyu Wang, Yejie Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhenhao Wang, Zhongsheng Wang, Zifan Wang, Chu Wei, Ming Wei, Shouxin Wei, Zichen Wen, Fan Wu, Haoning Wu, Rucong Wu, Wenhao Wu, Xiaoxue Wu, Ying-Nian Wu, Yong Wu, Yuxin Wu, Zijian Wu, Xinglang Xian, Chen Xiang, Yu-Ting Xiang, Bo Xiao, Chenjun Xiao, Xin Xiao, Jin Xie, Xiao-Ming Xie, Yifeng Xie, Zhe Xie, Bowei Xing, Yiming Xiong, Bao-Xin Xu, Boyu Xu, Jiale Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Qin Xu, Shuyao Xu, Suting Xu, Tiantian Xu, Tianxiang Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Yezhao Xu, Yue Xu, Ziyao Xu, Hao Xue, Junjie Yan, Yaoyao Yan, Fan Yang, Guang-Fu Yang, Hao Yang, Junwei Yang, Ruo-Tong Yang, Wenjie Yang, Xiaofei Yang, Xin-Yu Yang, Yi Yang, Yiling Yang, Ying Yang, Yuchen Yang, Zhen Yang, Zhilin Yang, Ruihan Yang, Zuhao Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhan Guo Ye, Bohong Yin, Haoxiang Yin, Xi Yin, Chengzheng Yu, Haozhe Yu, Long Yu, Sheng Yu, Shuying Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Tongtian Yue, Weize Yue, Yang Yue, Dunyuan Zha, Haobing Zhan, B. Zhang, Dehao Zhang, Feichang Zhang, Hao Zhang, Haoyuan Zhang, Huanyu Zhang, Jiapei Zhang, Jia-Xing Zhang, Jin Zhang, Kai-Yi Zhang, Miao Zhang, Pu Zhang, Qingle Zhang, Rong Zhang, Rui Zhang, Shaoshuai Zhang, Shiyi Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y Zhang, Yangkun Zhang, Yeqing Zhang, Yichi Zhang, Yikun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Zijing Zhang, Bin Zhao, Chenguang Zhao, Feifan Zhao, Jing Zhao, Jinxiang Zhao, Shuai Zhao, Wenshuo Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Hao Zheng, Huabin Zheng, Ruihan Zheng, Shao Jian Zheng, Tengyang Zheng, Haofeng Zhong, Lei Zhong, Longguang Zhong, M. Zhou, Qian Zhou, Runjie Zhou, Ru Zhou, Xinyu Zhou, Yiqiao Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yangjunfeng Zhu, Yuxuan Zhu, Zhengxin Zhu, Zhuang Chen, Weiyu Zhuang, and Xinxing Zu. Kimi k3: Open frontier intelligence. 2026.

Yushi Bai, Xin Lv, Jiajie Zhang, Hong Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. *ArXiv*, abs/2308.14508, 2023.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juan-Zi Li. Longalign: A recipe for long context alignment of large language models. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:267335080.

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.

Loïc Cabannes, Pierre-Emmanuel Mazaré, Gergely Szilvasy, Matthijs Douze, Maria Lomeli, Ilze Amanda Auzina, Justin Carpentier, Gabriel Synnaeve, and Hervé Jégou. Sparse delta memory: Scaling the state of linear rnns through sparsity. *ArXiv*, abs/2607.07386, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mo Bavarian, Clemens Winter, Phil Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, Suchir Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *ArXiv*, abs/1905.10044, 2019.

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

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021.

Róbert Csordás, Piotr Piekos, Kazuki Irie, and Jürgen Schmidhuber. Switchhead: Accelerating transformers with mixture-of-experts attention. *ArXiv*, abs/2312.07987, 2023.

Damai Dai, Chengqi Deng, Chenggang Zhao, Runxin Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *ArXiv*, abs/2307.08691, 2023.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. *ArXiv*, abs/2105.03011, 2021.

DeepSeek-AI, Anyi Xu, Bang Lin, Bing Xue, Bing-Li Wang, Bin Xu, Bo Wu, Bowei Zhang, Chao Lin, Chengyao Dong, Chen Ling, Chengda Lu, Chen Zhao, Chengqi Deng, Cheng Hou, Chen Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, De-Bin Yang, Deli Chen, Dong-Hui Li, Dong-Li Ji, Erhang Li, Fangchen Wei, Fangyun Lin, Fang Yuan, Fei Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guo Cao, Guo-Hui Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Hao Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Hao Yuan, Haowei Zhang, Haowen Luo, Hao Chen, Haozhe Ji, Heng Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J. Yang, J.-Q. Zhu, Jianming Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, JingChang Chen, Jing Zhou, Jing Xiang, Jingyang Yuan, Jing Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Junliang Ran, Jun Jiang, Junjie Qiu, Jun-Long Li, Junming Zheng, Jun-Mei Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lin Luo, Lin Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, Mingxi Di, My Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mi Zhou, Min Han, Ning Wang, Pan Huang, Pan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qi Jiang, Rui Tian, Rui qin Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Run Da Chen, Runqiu Yin, Runxin Xu, Ruo-Han Shen, Ruoyu Zhang, Ruyi Chen, Sh. Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shan-Shan Chen, Shaofei Cai, Shao Nie, Shao-Ping Wu, Shaoyuan Chen, Shengding Hu, Sheng Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tianhong Pei, Tian Ye, Tianle Lin, Tian Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tu-Tu Wang, W. Y. Zhang, Wl Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjin Yao, Wenjun Gao, Wenkai Yang, Wen-Wen Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xin-Yu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xu Fu, YC Yan, Yq. Wang, Yw Ma, Yanfen Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yao-Hui Wang, Yi Qian, Yingxia Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yi-Yu Xiong, Yi Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglu Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yukun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yue Xu, Yuhan Wu, Yu Meng, Yu Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yu Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, Zhanghua Wu, Ze Wang, Ze Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhe Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixiang Huang, Zhixuan Chen,

Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zi-Long Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zong xiang Yao. Deepseek-v4: Towards highly efficient million-token context intelligence. 2026.

Yufeng Du, Phillip Harris, Minyang Tian, Eliu A. Huerta, S. Ronanki, Subendhu Rongali, A.G. Galstyan, and Hao Peng. Rope distinguishes neither positions nor tokens in long contexts, provably. *ArXiv*, abs/2605.15514, 2026.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany M. Biron, Binh Tang, Bobbie Chern, Char lotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, E I Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cu-curell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Kenneth Heafield, Kevin R. Stone, Khalid El-Arini, Krithika Iyer, Kshi tiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Ma hesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bash-lykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, R. S. Mughil Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ron nie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sa hana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Sheng-Ye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stéphane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Fein-stein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin R. Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto de Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan

Smoth-ers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco (Paco) Guzmán, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry As-pegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Francis Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthias Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Re-strepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Mun ish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navy ata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Ro han Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, S. Yu. Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Kumar Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria O Ajayi, Victoria Montanez, Vijai Mohan, Vinay Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. 2024.

William Fedus, Barret Zoph, and Noam M. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *ArXiv*, abs/2101.03961, 2021.

Zizhuo Fu, Wenxuan Zeng, Runsheng Wang, and Meng Li. Attention sink forges native moe in attention layers: Sink-aware training to address head collapse. *ArXiv*, abs/2602.01203, 2026.

Mor Geva, R. Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *ArXiv*, abs/2012.14913, 2020.

Tianyu Guo, Druv Pai, Yu Bai, Jiantao Jiao, Michael I. Jordan, and Song Mei. Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in llms. *ArXiv*, abs/2410.13835, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2020.

Xanh Ho, A. Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *ArXiv*, abs/2011.01060, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy,

Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022.

Luyang Robby Huang, Shuyang Cao, Nikolaus Nova Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *North American Chapter of the Association for Computational Linguistics*, 2021.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Barbara Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *ArXiv*, abs/2401.04088, 2024.

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng Yan. Moh: Multi-head attention as mixture-of-head attention. *ArXiv*, abs/2410.11842, 2024.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *ArXiv*, abs/1705.03551, 2017.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2017.

Xin Li and Dan Roth. Learning question classifiers. In *International Conference on Computational Linguistics*, 2002.

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, Jianling Su, Yuxin Wu, Neo Y. Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. Moba: Mixture of block attention for long-context llms. *ArXiv*, abs/2502.13189, 2025.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

MiMo Team. Mimo-v2.5. https://huggingface.co/collections/XiaomiMiMo/mimo-v25, 2026.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova Dassarma, Thomas Henighan, Benjamin Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, John Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom B. Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *ArXiv*, abs/2209.11895, 2022.

Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. *ArXiv*, abs/2406.17557, 2024.

Hao Peng, Roy Schwartz, Dianqi Li, and Noah A. Smith. A mixture of h - 1 heads is better than h heads. *ArXiv*, abs/2005.06537, 2020.

Piotr Piekos, R'obert Csord'as, and Jürgen Schmidhuber. Mixture of sparse attention: Content-based learnable sparse attention via expert-choice routing. *ArXiv*, abs/2505.00315, 2025. URL https://api.semanticscholar.org/CorpusID:278237209.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *ArXiv*, abs/2505.06708, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande. *Communications of the ACM*, 64:99 – 106, 2019.

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

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *ArXiv*, abs/2406.10774, 2024.

Vishesh Tripathi and Abhay Kumar. Grouped query experts: Mixture-of-experts on gqa self-attention. *ArXiv*, abs/2606.20945, 2026.

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

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *ArXiv*, abs/2408.15664, 2024.

Davis Wertheimer, Aozhong Zhang, Derrick Liu, Penghang Yin, and Naigang Wang. Frayed rope and long inputs: A geometric perspective. *ArXiv*, abs/2603.18017, 2026.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ArXiv*, abs/2309.17453, 2023.

Yufei Xu, Fanxu Meng, Fan Jiang, YuXuan Wang, Ruijie Zhou, Jie Wu, Zhixin Pan, Zhaohui Wang, Xiaojuan Tang, Wenjie Pei, Tongxuan Liu, Di Yin, Xing Sun, and Muhan Zhang. Hisa: Efficient hierarchical indexing for fine-grained sparse attention. *ArXiv*, abs/2603.28458, 2026.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024a.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *ArXiv*, abs/2412.06464, 2024b.

Yuanhang Yang, Chaozheng Wang, and Jing Li. Umoe: Unifying attention and ffn with shared experts. *ArXiv*, abs/2505.07260, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

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. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*, 2019.

Xiaofeng Zhang, Yikang Shen, Zeyu Huang, Jie Zhou, Wenge Rong, and Zhang Xiong. Mixture of attention heads: Selecting attention heads per token. In *Conference on Empirical Methods in Natural Language Processing*, 2022.

Zhenyu (Allen) Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *ArXiv*, abs/2306.14048, 2023.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zairi Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir R. Radev. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In *North American Chapter of the Association for Computational Linguistics*, 2021.

A Load Balancing Strategies

We describe three balancing strategies for a single routed attention layer. For a training batch $\mathcal{B}$ containing N tokens, each token selects K of H heads. We use the head affinities $a_{t,i}$, selection indicators $m_{t,i}$, assignment fractions f_i, and average affinities p_i defined in the main text. Auxiliary losses are summed across routed attention layers and weighted by λ_{bal}. Shared heads are excluded from all balancing statistics.

A.1 CV-Based Load Balancing

Following Shazeer et al. (2017), CV-based balancing penalizes variation in both routing importance and expected token load. For a nonnegative head-statistic vector $\boldsymbol{u} = (u_0, \dots, u_{H-1})$ with positive mean $\bar{u}$, the squared coefficient of variation is

$$\mathrm{CV}(\boldsymbol{u})^2 = \frac{H^{-1}\sum_{i=0}^{H-1}(u_i - \bar{u})^2}{\bar{u}^2}, \qquad \bar{u} = \frac{1}{H}\sum_{i=0}^{H-1} u_i. \tag{8}$$

This measures variance relative to the squared mean and is minimized when all entries are equal.

For head i, define its routing importance I_i and expected assignment count ℓ_i as

$$I_i = \sum_{t\in\mathcal{B}} a_{t,i} = Np_i, \qquad \ell_i = \sum_{t\in\mathcal{B}} \Pr_{\text{noise}}\left(i \in \mathcal{S}_K^{\text{noise}}(t)\right), \tag{9}$$

where $\mathcal{S}_K^{\text{noise}}(t)$ denotes the heads selected by noisy Top-K routing, and the probability is taken over the routing noise. The importance term measures total affinity, while the expected-load term measures how often a head is selected under noisy routing. Let $\boldsymbol{I} = (I_0, \dots, I_{H-1})$ and $\boldsymbol{\ell} = (\ell_0, \dots, \ell_{H-1})$. The combined loss is

$$\mathcal{L}_{\text{CV}} = \alpha_{\text{imp}}\,\mathrm{CV}(\boldsymbol{I})^2 + \alpha_{\text{load}}\,\mathrm{CV}(\boldsymbol{\ell})^2, \tag{10}$$

where $\alpha_{\text{imp}}, \alpha_{\text{load}} > 0$ set the relative weights of the two penalties. Their overall strength is controlled by λ_{bal}.

The importance penalty is equivalent to $H\sum_i (p_i - 1/H)^2$ and is differentiable through the head affinities. The load penalty additionally requires a differentiable expected-load estimator, obtained through noisy Top-K routing (Shazeer et al., 2017). It is not computed by directly differentiating the hard assignment counts NKf_i. Thus, the full CV formulation balances both affinity mass and expected head utilization, rather than affinity mass alone.

A.2 Switch-Style Auxiliary Balancing

The Switch-style loss (Fedus et al., 2021) combines observed head utilization with differentiable routing affinities:

$$\mathcal{L}_{fp} = H\sum_{i=0}^{H-1} f_i p_i. \tag{11}$$

For Top-K head routing, f_i is normalized by the total number of assignments NK, so both $\{f_i\}$ and $\{p_i\}$ sum to one. The loss equals one under uniform assignments and affinities, independently of H and K.

During backpropagation, f_i is treated as constant, while gradients flow through p_i. Since $\partial\mathcal{L}_{fp}/\partial p_i = Hf_i$, a head with a larger assignment fraction receives a stronger penalty on its average affinity. The observed load therefore guides the router toward less-used heads without requiring a differentiable estimate of assignment counts.

This objective is also related to CV regularization. For $\boldsymbol{f} = (f_0, \dots, f_{H-1})$, when $p_i \approx f_i$,

$$\mathcal{L}_{fp} \approx H\sum_{i=0}^{H-1} f_i^2 = 1 + \mathrm{CV}(\boldsymbol{f})^2. \tag{12}$$

Unlike direct count-based CV regularization, the affinity factor provides a differentiable path to the router. As with the CV strategy, an overly large λ_{bal} can make balancing gradients compete with the head specialization favored by the language-modeling objective.

A.3 AUXILIARY-LOSS-FREE BALANCING

Following Wang et al. (2024), loss-free balancing controls head utilization through routing biases rather than an auxiliary objective. Each head maintains a scalar bias b_i, initialized to zero. Selection uses biased affinities, while output gates retain the original affinities:

$$\mathcal{S}_K(t) = \mathrm{TopK}_i(a_{t,i} + b_i), \qquad m_{t,i} = \mathbf{1}\{i \in \mathcal{S}_K(t)\}, \qquad g_{t,i} = m_{t,i} a_{t,i}. \tag{13}$$

Thus, biases change which heads are selected but do not enter their gating weights. Selected affinities are not renormalized.

Let c_i denote the number of token assignments received by head i in the current batch, and let $\bar{c}$ be the average count across heads. After each completed training batch, we update

$$c_i = \sum_{t \in \mathcal{B}} m_{t,i} = NKf_i, \qquad \bar{c} = \frac{NK}{H}, \qquad b_i \leftarrow b_i + \eta\, \mathrm{sign}(\bar{c} - c_i), \tag{14}$$

where $\eta > 0$ is the update rate and $\mathrm{sign}(0) = 0$. An overloaded head receives a negative bias adjustment, reducing its chance of future selection. An underloaded head receives a positive adjustment. This forms a feedback loop from observed assignments to subsequent routing decisions.

Bias updates occur outside backpropagation. In NAMOH, biases remain fixed within each training batch and during inference, so updates do not revise earlier token assignments. The router learns token-to-head affinities through the language-modeling objective, while the biases regulate head utilization. Unlike the two auxiliary-loss strategies, this method introduces no direct load-balancing gradient into the router.

B COMPLEXITY ANALYSIS AND COMPARISONS

B.1 SETUP AND ACCOUNTING

We analyze one layer with H query heads, model width $d_m = d_{\text{model}}$, head width $d_h = d_{\text{head}}$, and a prefix of $T \geq 1$ tokens. Routed variants activate K query heads per token, with $1 \leq K \leq H$. GQA places g query heads in each KV group, where $g \mid H$; group-routed variants additionally require $g \mid K$. For ungrouped variants, $g = 1$. SA uses an integer block size $B \geq 1$ and a retained fraction $\alpha \in (0, 1]$ of each available local history. All widths are held fixed; we do not require $d_m = Hd_h$.

Let P_{total} and P_{active} count all learned weights and those evaluated for one token. We use a single bias-free linear router for MoH and NAMOH to isolate their attention differences. An H-way head router contains Hd_m weights, while an H/g-way group router contains Hd_m/g. All router weights are active for every token. Projection biases, always-active shared heads, and extra attention branches are excluded. The metadata-based block selectors considered here introduce no learned weights.

As in the main text, M_{KV} counts stored KV pairs, and $A_{\text{KV}}(T)$ counts distinct historical pairs used by the next token. Multiplying either count by $2d_h$ gives its scalar data volume. Let C_{attn} count all causal query-key interactions during prefill, including self-attention. Unlike KV activation, this count includes separate interactions for query heads that share the same KV pair. All counts are logical and exclude padding, temporary buffers, and training activations.

B.2 MHA AND GQA

MHA. Each head has independent query, key, value, and output projections (Vaswani et al., 2017). Each projection contributes $d_m d_h$ weights, so

$$P_{\text{total}} = P_{\text{active}} = 4Hd_m d_h, \qquad M_{\text{KV}} = HT, \qquad A_{\text{KV}}(T) = HT. \tag{15}$$

A query at one-based causal rank r attends to r entries per head. Summing over all ranks gives

$$C_{\text{attn}} = H \sum_{r=1}^{T} r = \frac{HT(T+1)}{2}. \tag{16}$$

GQA. Let $\mathcal{Q}_j$ denote the g query heads assigned to KV group j, for $j = 0, \ldots, H/g - 1$. GQA retains separate query and output projections but shares key and value projections within each group (Ainslie

et al., 2023). Therefore,

$$P_{\text{total}} = P_{\text{active}} = 2Hd_m d_h + \frac{2H}{g} d_m d_h = 2H\left(1 + \frac{1}{g}\right) d_m d_h,$$
$$M_{\text{KV}} = \frac{HT}{g}, \qquad A_{\text{KV}}(T) = \frac{HT}{g}. \tag{17}$$

Each stored key still interacts with g queries at each causal position:

$$C_{\text{attn}} = \sum_{j=0}^{H/g-1} \sum_{r=1}^{T} gr = \frac{HT(T+1)}{2}. \tag{18}$$

Thus, KV sharing reduces storage and distinct access, but not the query-key interaction count at fixed H.

B.3 Block-Selected Sparse Attention

Selection model. We analyze a fixed-fraction version of block selection that scans historical metadata, using blocks of B entries. MoBA scores mean-pooled keys, while Quest uses channelwise key minima and maxima (Lu et al., 2025; Tang et al., 2024). We apply the specified selector during both prefill and decoding; this does not reproduce every cited method's execution settings.

For a nonnegative history length ℓ, let $N_B(\ell)$ be its number of blocks. At local causal rank $r \geq 1$, let $R_{\alpha,B}(r)$ be the selected block count, including the current block:

$$N_B(\ell) = \left\lceil \frac{\ell}{B} \right\rceil, \qquad R_{\alpha,B}(r) = \lceil \alpha N_B(r) \rceil. \tag{19}$$

For example, $\alpha = 0.2$ targets approximately 20% of each local history. We always include the current block and select the remaining blocks from completed historical blocks. Future blocks are excluded. The current block uses a causal mask and is not ranked using metadata that contains future keys, following the causality rule of MoBA (Lu et al., 2025).

Exact selected-entry count. Let $a_{\alpha,B}(r)$ count entries attended by a query at local rank r, including itself. Each selected historical block contains B entries, and the current block contributes $r - B[N_B(r) - 1]$ visible entries. Hence,

$$\begin{aligned} a_{\alpha,B}(r) &= B[R_{\alpha,B}(r) - 1] + r - B[N_B(r) - 1] \\ &= r - B[N_B(r) - R_{\alpha,B}(r)]. \end{aligned} \tag{20}$$

Writing $\varepsilon_r = a_{\alpha,B}(r) - \alpha r$ for the rounding error gives

$$\begin{aligned} \varepsilon_r &= B[R_{\alpha,B}(r) - \alpha N_B(r)] + (1-\alpha)[r - BN_B(r)], \\ |\varepsilon_r| &\leq B. \end{aligned} \tag{21}$$

Thus, $a_{\alpha,B}(r) = \alpha r + \mathcal{O}(B)$. The relative approximation is useful when $\alpha r \gg B$; a history contained in one block is attended densely.

SA retains every KV pair because a later query may select a previously unused block. Its weights are unchanged from MHA, and

$$\begin{aligned} M_{\text{KV}} &= HT, \\ A_{\text{KV}}(T) &= H[a_{\alpha,B}(T+1) - 1] = \alpha HT + \mathcal{O}(HB). \end{aligned} \tag{22}$$

The subtraction removes the next token's self-entry.

Causal prefill interactions. Define the selected causal sum for a local sequence of length ℓ as

$$\begin{aligned}\Phi_{\alpha,B}(\ell) &= \sum_{r=1}^{\ell} a_{\alpha,B}(r) \\ &= \frac{\ell(\ell+1)}{2} - B\sum_{r=1}^{\ell}[N_B(r) - R_{\alpha,B}(r)] \\ &= \frac{\alpha\ell(\ell+1)}{2} + \sum_{r=1}^{\ell}\varepsilon_r \\ &= \frac{\alpha\ell(\ell+1)}{2} + \mathcal{O}(B\ell).\end{aligned} \tag{23}$$

Self-attention and the mandatory current block are included exactly in $\Phi_{\alpha,B}$. Applying this sum to all H heads yields

$$C_{\text{attn}} = H\Phi_{\alpha,B}(T) = \frac{\alpha HT^2}{2} + \mathcal{O}(HBT). \tag{24}$$

Metadata storage and scanning. Let ν be the number of metadata vectors per block, each of width d_h. Mean pooling uses $\nu = 1$, and minima and maxima use $\nu = 2$. The metadata storage in scalars is $M_{\text{meta}} = \nu d_h H N_B(T)$. Construction costs $\mathcal{O}(HTd_h)$ and supports incremental updates.

Let $D_{\text{meta}}(T)$ and C_{meta} count candidate block summaries inspected during the next decoding step and throughout prefill, respectively. A query at rank r scans $N_B(r) - 1$ completed blocks. To sum these visits, define

$$\Gamma_B(\ell) = \sum_{r=1}^{\ell} N_B(r). \tag{25}$$

Write $\ell = uB + v$, where $u = \lfloor \ell/B \rfloor$ and $0 \le v < B$. Each complete block contributes B queries with the same block count, giving

$$\begin{aligned}\Gamma_B(\ell) &= B\sum_{b=1}^{u} b + (u+1)v \\ &= \frac{Bu(u+1)}{2} + (u+1)v = \frac{\ell^2}{2B} + \mathcal{O}(\ell).\end{aligned} \tag{26}$$

Here, b indexes complete local blocks. Consequently,

$$\begin{aligned}D_{\text{meta}}(T) &= H[N_B(T+1) - 1] = H\left\lfloor \frac{T}{B} \right\rfloor, \\ C_{\text{meta}} &= H[\Gamma_B(T) - T] = \frac{HT^2}{2B} + \mathcal{O}(HT).\end{aligned} \tag{27}$$

The subtraction excludes the unscored current block. These visits do not acquire a factor of α, since the selector scans all candidates before choosing blocks.

B.4 MoH

We use the dense-projection MoH baseline: all query, key, value, and output projection weights are evaluated, but only K heads perform attention (Jin et al., 2024). Every head stores every prefix token, including tokens for which its output was inactive. Under the router convention in Section B.1,

$$\begin{aligned}P_{\text{total}} = P_{\text{active}} &= 4Hd_m d_h + Hd_m, \\ M_{\text{KV}} = HT, &\qquad A_{\text{KV}}(T) = KT.\end{aligned} \tag{28}$$

At global causal rank r, each of the K selected heads attends to all r entries. Therefore,

$$C_{\text{attn}} = \sum_{r=1}^{T} Kr = \frac{KT(T+1)}{2}. \tag{29}$$

Head selection reduces KV activation and attention interactions, but not projection activation or cache storage in this baseline.

B.5 NAMOH

Parameters and routed storage. Routing precedes projection, so inactive token-head pairs produce no queries, keys, or values. Only selected output slices are evaluated:

$$P_{\text{total}} = 4Hd_m d_h + Hd_m, \qquad P_{\text{active}} = 4Kd_m d_h + Hd_m. \tag{30}$$

Let $m_{t,i}$ indicate whether token t selects head i, and let $\mathcal{S}_K(t)$ be the selected set. The head lengths satisfy

$$n_i = \sum_{t=0}^{T-1} m_{t,i}, \qquad \sum_{i=0}^{H-1} n_i = \sum_{t=0}^{T-1}\sum_{i=0}^{H-1} m_{t,i} = TK. \tag{31}$$

Only these assignments create KV pairs. Hence,

$$M_{\text{KV}} = TK, \qquad A_{\text{KV}}(T) = \sum_{i\in\mathcal{S}_K(T)} n_i. \tag{32}$$

Let $\bar{n} = TK/H$ denote the mean head length. Under equal lengths, activation is $K\bar{n} = TK^2/H$. Equal lengths require integer $\bar{n}$; approximate balance gives the main-text estimate.

Causal interaction count. Within head i, the query at local rank r attends to the first r entries. Thus,

$$\begin{aligned} C_{\text{attn}} &= \sum_{i=0}^{H-1}\sum_{r=1}^{n_i} r = \frac{1}{2}\sum_{i=0}^{H-1} n_i(n_i+1) \\ &= \frac{1}{2}\sum_{i=0}^{H-1} n_i^2 + \frac{TK}{2}. \end{aligned} \tag{33}$$

For the load vector $\boldsymbol{n} = (n_0, \ldots, n_{H-1})$, define its squared coefficient of variation using the population variance:

$$\text{CV}(\boldsymbol{n})^2 = \frac{H^{-1}\sum_{i=0}^{H-1}(n_i - \bar{n})^2}{\bar{n}^2}. \tag{34}$$

Since $\sum_i (n_i - \bar{n}) = 0$, expanding around the mean gives

$$\begin{aligned} \sum_{i=0}^{H-1} n_i^2 &= H\bar{n}^2 + 2\bar{n}\sum_{i=0}^{H-1}(n_i - \bar{n}) + \sum_{i=0}^{H-1}(n_i - \bar{n})^2 \\ &= H\bar{n}^2[1 + \text{CV}(\boldsymbol{n})^2] \\ &= \frac{T^2K^2}{H}[1 + \text{CV}(\boldsymbol{n})^2]. \end{aligned} \tag{35}$$

Substitution into Equation 33 yields

$$C_{\text{attn}} = \frac{T^2K^2}{2H}[1 + \text{CV}(\boldsymbol{n})^2] + \frac{TK}{2}. \tag{36}$$

The causal sum includes self-attention exactly; the second moment determines the leading cost.

Balance and matched comparisons. The inequalities $\sum_i n_i^2 \geq (\sum_i n_i)^2/H$ and $n_i^2 \leq Tn_i$ imply

$$\frac{T^2K^2}{2H} + \frac{TK}{2} \leq C_{\text{attn}} \leq \frac{KT(T+1)}{2}. \tag{37}$$

Equal lengths attain the lower bound when feasible. If every token selects the same K heads, the upper bound is attained and decoding activation becomes KT. Storage remains TK in both cases. Batch-level load balancing does not guarantee balanced histories within every prefix.

For an all-active baseline with $J \in \{H, K\}$ heads, the balanced finite-length prefill ratio is

$$\frac{C_{\text{attn}}}{JT(T+1)/2} = \frac{TK^2/H + K}{J(T+1)} \longrightarrow \frac{K^2}{HJ} \quad \text{as } T \to \infty. \tag{38}$$

The choices $J = H$ and $J = K$ match total and active head-projection parameters, respectively; router weights are additional.

B.6 GQA+SA

Independent or shared selection. Query heads sharing a KV group can select blocks independently. Let $\mathcal{E}_{t,i}$ be the set of historical KV entries selected by query head i for token t. Distinct access within group j then counts the union:

$$\max_{i\in\mathcal{Q}_j}|\mathcal{E}_{T,i}| \le \left|\bigcup_{i\in\mathcal{Q}_j}\mathcal{E}_{T,i}\right| \le \min\left\{T, \sum_{i\in\mathcal{Q}_j}|\mathcal{E}_{T,i}|\right\}. \tag{39}$$

Ignoring rounding, the union can range from αT to $\min(T, g\alpha T)$ entries. We instead use one shared block selection per group, following the group-consistent design of Yuan et al. (2025). Queries share block indices, not their attention weights.

Costs with shared indices. Parameter counts remain those of GQA. Each of its H/g groups stores one full history and selects one block set:

$$\begin{aligned} M_{\mathrm{KV}} &= \frac{HT}{g}, \\ A_{\mathrm{KV}}(T) &= \frac{H}{g}[a_{\alpha,B}(T+1)-1] = \frac{\alpha HT}{g} + \mathcal{O}\left(\frac{HB}{g}\right). \end{aligned} \tag{40}$$

All g queries still use each selected entry, so

$$\begin{aligned} C_{\mathrm{attn}} &= \frac{H}{g}\, g\, \Phi_{\alpha,B}(T) = \frac{\alpha HT^2}{2} + \mathcal{O}(HBT), \\ D_{\mathrm{meta}}(T) &= \frac{H}{g}\left\lfloor \frac{T}{B} \right\rfloor, \\ C_{\mathrm{meta}} &= \frac{H}{g}[\Gamma_B(T) - T] = \frac{HT^2}{2gB} + \mathcal{O}\left(\frac{HT}{g}\right). \end{aligned} \tag{41}$$

Shared indices versus shared scoring. Let χ_g denote the number of d_h-scale scoring operations used to form one group-block score, with $\chi_1 = 1$. Aggregating separately computed scores from all g queries gives $\chi_g = g$, as in the per-head score aggregation of NSA (Yuan et al., 2025). A single dot product using a pooled group query instead gives $\chi_g = 1$, but generally defines a different scoring rule. Both produce one shared block set and have the same KV counts. Their scoring arithmetic is $\Theta(\chi_g d_h D_{\mathrm{meta}})$ during decoding and $\Theta(\chi_g d_h C_{\mathrm{meta}})$ during prefill. Shared block indices therefore do not by themselves imply shared scoring arithmetic.

B.7 GQA+NAMOH

Query-head routing. One option selects K of the H query heads independently. Let $u_{t,j} = \mathbf{1}\{\mathcal{S}_K(t) \cap \mathcal{Q}_j \neq \emptyset\}$ indicate whether token t touches KV group j, where $\mathbf{1}$ is the indicator function. The number of KV groups written by a token satisfies

$$\left\lceil \frac{K}{g} \right\rceil \le \sum_{j=0}^{H/g-1} u_{t,j} \le \min\left(K, \frac{H}{g}\right). \tag{42}$$

If a shared pair is stored whenever any query in its group is selected, then $n_j = \sum_{t=0}^{T-1} u_{t,j}$ and

$$M_{\mathrm{KV}} = \sum_j n_j, \qquad T\left\lceil \frac{K}{g} \right\rceil \le M_{\mathrm{KV}} \le T\min\left(K, \frac{H}{g}\right). \tag{43}$$

Allowing active queries to read their groups' union histories gives $A_{\mathrm{KV}}(T) = \sum_j u_{T,j} n_j$. Preserving separate query-head histories instead requires membership masks and makes access depend on overlaps between current and historical assignments. Neither storage nor access is determined by T, H, K, and g alone.

Adopted KV-group routing. We instead select exactly K/g of the H/g KV groups. Let $\mathcal{G}_t$ denote the selected group set for token t. All g query heads in each selected group participate and share one

routing gate. The group writes one KV pair. Thus, exactly K query and output projections and K/g key and value projections are evaluated:

$$P_{\text{total}} = 2H\left(1+\frac{1}{g}\right)d_m d_h + \frac{H}{g}d_m,$$
$$P_{\text{active}} = 2K\left(1+\frac{1}{g}\right)d_m d_h + \frac{H}{g}d_m. \tag{44}$$

For this routing rule, the group lengths satisfy

$$n_j = \sum_{t=0}^{T-1} \mathbf{1}\{j \in \mathcal{G}_t\}, \qquad \sum_{j=0}^{H/g-1} n_j = \frac{TK}{g}, \qquad \bar{n} = \frac{TK/g}{H/g} = \frac{TK}{H}. \tag{45}$$

The group activation fraction remains K/H, so the average history length is TK/H, not $TK/(Hg)$. Consequently,

$$M_{\text{KV}} = \frac{TK}{g}, \qquad A_{\text{KV}}(T) = \sum_{j\in\mathcal{G}_T} n_j \approx \frac{K}{g}\frac{TK}{H} = \frac{TK^2}{Hg}. \tag{46}$$

Causal prefill derivation. For group loads $\boldsymbol{n} = (n_0, \ldots, n_{H/g-1})$, the squared coefficient of variation is

$$\text{CV}(\boldsymbol{n})^2 = \frac{(g/H)\sum_j (n_j - \bar{n})^2}{\bar{n}^2}. \tag{47}$$

Expanding the second moment gives

$$\sum_j n_j^2 = \frac{H}{g}\bar{n}^2 + \sum_j (n_j - \bar{n})^2$$
$$= \frac{T^2K^2}{Hg}[1 + \text{CV}(\boldsymbol{n})^2]. \tag{48}$$

Each local rank now produces g queries, so

$$C_{\text{attn}} = \sum_j \sum_{r=1}^{n_j} gr = \frac{g}{2}\sum_j n_j^2 + \frac{g}{2}\sum_j n_j$$
$$= \frac{T^2K^2}{2H}[1 + \text{CV}(\boldsymbol{n})^2] + \frac{TK}{2}. \tag{49}$$

KV sharing therefore preserves NAMOH's balanced interaction count while reducing distinct KV storage and access by g. When head-relative RoPE is enabled, all queries and the shared key use the same group-local position index. Group routing also coarsens the routing choices and requires $g \le K$; a single KV group leaves no nontrivial group selection.

B.8 SA+NAMOH

We first construct routed head histories, then form blocks in each head's local token order. The retained fraction α applies to that local history, not to the original T-token prefix. Parameter counts and KV storage remain those of NAMOH. Using the head lengths from Section B.5,

$$A_{\text{KV}}(T) = \sum_{i\in\mathcal{S}_K(T)} [a_{\alpha,B}(n_i + 1) - 1]$$
$$= \alpha \sum_{i\in\mathcal{S}_K(T)} n_i + \mathcal{O}(KB) \approx \frac{\alpha TK^2}{H}. \tag{50}$$

For prefill, apply the selected causal sum independently to every head:

$$C_{\text{attn}} = \sum_i \Phi_{\alpha,B}(n_i)$$
$$= \frac{\alpha}{2}\sum_i n_i(n_i+1) + \mathcal{O}\left(B\sum_i n_i\right) \tag{51}$$
$$= \frac{\alpha T^2K^2}{2H}[1 + \text{CV}(\boldsymbol{n})^2] + \mathcal{O}(BTK).$$

The indexer scans only the selected routed histories during decoding:

$$\begin{aligned} D_{\text{meta}}(T) &= \sum_{i \in \mathcal{S}_K(T)} \left\lfloor \frac{n_i}{B} \right\rfloor = \frac{1}{B} \sum_{i \in \mathcal{S}_K(T)} n_i + \mathcal{O}(K), \\ C_{\text{meta}} &= \sum_i [\Gamma_B(n_i) - n_i] = \frac{1}{2B} \sum_i n_i^2 + \mathcal{O}(TK) \\ &= \frac{T^2K^2}{2HB}[1 + \mathrm{CV}(\boldsymbol{n})^2] + \mathcal{O}(TK). \end{aligned} \tag{52}$$

Under balance, the decoding scan has leading term $TK^2/(HB)$. Routing reduces the histories scanned by the selector; block selection then reduces token-level reads within those histories.

B.9 GQA+SA+NAMOH

We combine KV-group routing with group-shared block selection. Each token activates K/g groups, and each active group selects blocks from its own routed history. All g queries in that group use the same selected blocks. Parameter counts remain those in Equation 44, and

$$\begin{aligned} M_{\text{KV}} &= \frac{TK}{g}, \\ A_{\text{KV}}(T) &= \sum_{j \in \mathcal{G}_T} [a_{\alpha,B}(n_j + 1) - 1] \\ &= \alpha \sum_{j \in \mathcal{G}_T} n_j + \mathcal{O}\left(\frac{KB}{g}\right) \approx \frac{\alpha TK^2}{Hg}. \end{aligned} \tag{53}$$

Combining the local causal sum with the group second moment gives

$$\begin{aligned} C_{\text{attn}} &= g \sum_j \Phi_{\alpha,B}(n_j) \\ &= \frac{\alpha g}{2} \sum_j n_j(n_j + 1) + \mathcal{O}\left(Bg \sum_j n_j\right) \\ &= \frac{\alpha T^2K^2}{2H}[1 + \mathrm{CV}(\boldsymbol{n})^2] + \mathcal{O}(BTK). \end{aligned} \tag{54}$$

The shared indexer visits each candidate block once per active group:

$$\begin{aligned} D_{\text{meta}}(T) &= \sum_{j \in \mathcal{G}_T} \left\lfloor \frac{n_j}{B} \right\rfloor = \frac{1}{B} \sum_{j \in \mathcal{G}_T} n_j + \mathcal{O}\left(\frac{K}{g}\right), \\ C_{\text{meta}} &= \sum_j [\Gamma_B(n_j) - n_j] = \frac{1}{2B} \sum_j n_j^2 + \mathcal{O}\left(\frac{TK}{g}\right) \\ &= \frac{T^2K^2}{2HgB}[1 + \mathrm{CV}(\boldsymbol{n})^2] + \mathcal{O}\left(\frac{TK}{g}\right). \end{aligned} \tag{55}$$

Under balance, the decoding scan has leading term $TK^2/(HgB)$. Relative to H-head MHA, persistent KV storage is $K/(Hg)$ and activation is approximately $\alpha K^2/(H^2g)$ of the baseline. The factor α reduces activation, not persistent KV storage.

B.10 Overall Costs and Scaling

Persistent cache storage. For any SA variant, index its KV histories by $j = 0, \ldots, H/g - 1$, with $g = 1$ for ungrouped attention. The lengths n_j equal T without routing and are assignment-dependent otherwise. Metadata contributes

$$M_{\text{meta}} = \nu d_h \sum_j \left\lceil \frac{n_j}{B} \right\rceil \le \nu d_h \left(\frac{M_{\text{KV}}}{B} + \frac{H}{g} \right) \tag{56}$$

Table 6: **Leading attention and indexer terms.** The prefill column reports $C_{\text{attn}} + \chi_g C_{\text{meta}}$; the decoding column reports $gA_{\text{KV}}(T) + \chi_g D_{\text{meta}}(T)$. Insert these terms into Equation 58 with the active weights in Table 1 to obtain total arithmetic costs. Routed histories are balanced, and finite-length and block-rounding terms are omitted here but retained in the derivations.

Mechanism	Prefill	Single-token decoding
MHA	$HT^2/2$	HT
GQA	$HT^2/2$	HT
SA	$\left(\alpha + \frac{1}{B}\right)\frac{HT^2}{2}$	$\left(\alpha + \frac{1}{B}\right)HT$
MoH	$KT^2/2$	KT
NAMOH	$T^2K^2/(2H)$	TK^2/H
GQA+SA	$\left(\alpha + \frac{\chi_g}{gB}\right)\frac{HT^2}{2}$	$\left(\alpha + \frac{\chi_g}{gB}\right)HT$
GQA+**NAMOH**	$T^2K^2/(2H)$	TK^2/H
SA+**NAMOH**	$\left(\alpha + \frac{1}{B}\right)\frac{T^2K^2}{2H}$	$\left(\alpha + \frac{1}{B}\right)\frac{TK^2}{H}$
GQA+SA+**NAMOH**	$\left(\alpha + \frac{\chi_g}{gB}\right)\frac{T^2K^2}{2H}$	$\left(\alpha + \frac{\chi_g}{gB}\right)\frac{TK^2}{H}$

in scalars. Total KV and metadata storage is $2d_h M_{\text{KV}} + M_{\text{meta}}$. For the three-way combination, this is at most

$$\frac{2d_h TK}{g} + \nu d_h \left(\frac{TK}{gB} + \frac{H}{g}\right). \tag{57}$$

Block allocation can additionally leave fewer than B unused KV slots per nonempty history. Metadata construction costs $\mathcal{O}(d_h M_{\text{KV}})$ over the prefix and is dominated by projection work in this model.

Arithmetic accounting. Let h_{act} be the number of active query heads: H without head routing and K for MoH and NAMOH variants. With group-shared access, the next token evaluates $gA_{\text{KV}}(T) + h_{\text{act}}$ query-key interactions, including self-attention. Let F_{prefill} and F_{decode} denote total arithmetic work over prefill and one decoding step. Assuming linear-work head and block selection,

$$\begin{aligned} F_{\text{prefill}} &= \mathcal{O}(TP_{\text{active}} + d_h[C_{\text{attn}} + \chi_g C_{\text{meta}}]), \\ F_{\text{decode}} &= \mathcal{O}(P_{\text{active}} + d_h[gA_{\text{KV}}(T) + h_{\text{act}} + \chi_g D_{\text{meta}}(T)]). \end{aligned} \tag{58}$$

Set $C_{\text{meta}} = D_{\text{meta}} = 0$ without block selection. For ungrouped attention, $\chi_1 = 1$. Sorting all scores can add work beyond this linear-selection model.

For the three-way combination with balanced histories and $\alpha TK/H \gg B$, Equation 58 becomes

$$\begin{aligned} F_{\text{prefill}} &= \mathcal{O}\left(T\left[2K\left(1 + \frac{1}{g}\right)d_m d_h + \frac{H}{g}d_m\right] + \frac{d_h T^2 K^2}{H}\left[\alpha + \frac{\chi_g}{gB}\right]\right), \\ F_{\text{decode}} &= \mathcal{O}\left(2K\left(1 + \frac{1}{g}\right)d_m d_h + \frac{H}{g}d_m + \frac{d_h TK^2}{H}\left[\alpha + \frac{\chi_g}{gB}\right]\right). \end{aligned} \tag{59}$$

With per-head score aggregation, $\chi_g = g$, so KV sharing does not reduce the leading indexer arithmetic by g. A pooled-query indexer with $\chi_g = 1$ realizes that additional scoring reduction. Neither case changes the g separate attention computations per shared KV entry.

Fractional versus fixed budgets. The multiplicative activation formulas assume a fixed retained fraction of each local history. They do not imply the same gain under a fixed absolute budget. If at most R blocks are retained per active history and $S = RB$ is the corresponding entry budget, the group-routed hybrid instead satisfies

$$A_{\text{KV}}(T) \le \frac{K}{g}S, \qquad C_{\text{attn}} \le TKS. \tag{60}$$

Once histories exceed this budget, head routing does not multiply these saturated bounds by another factor of K/H. It still reduces storage and the histories scanned by the indexer.

At fixed α, B, H, K, and g, both fractional attention and full metadata scanning remain quadratic in prefill length T. Their coefficients are reduced, not their asymptotic order. Setting $\alpha = 1$ restores

dense attention within each available local history, and block selection can then be bypassed. These costs describe computational structure rather than guaranteed latency; physical gains also depend on cache reuse, data movement, and kernel scheduling.