SCALED SINK ATTENTION: MEASURING AND REDUCING ATTENTION SINK AFTER NORMALIZATION IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models (LLMs) attract growing interest, the attention mechanism at their core has become one of their most critical components. Yet Softmax attention still exhibits *attention sink*: a large share of the probability mass is placed on a few uninformative tokens, usually the first one. Recent remedies relax the sum-to-one constraint of Softmax with a learnable denominator bias (Sink Attention) or a sigmoid output gate (Gated Attention) and report much lower first-token scores. We point out, however, that any such per-query rescaling, including ours, leaves the *normalized* attention distribution unchanged for fixed logits; whether a method removes the sink is decided by what the trained model learns and must be measured after normalization. Under this measure, existing remedies still leave a substantial share of the attention on the first token and, in addition, suppress attention on informative tokens. We propose **Scaled Sink Attention (SSA)**, which combines two components: (i) an *input-dependent virtual sink*, a per-head logit projected from the hidden state that competes inside the Softmax and is then discarded, so that the drained mass is decided token by token against the attention evidence; and (ii) a *fixed gain* $g = 1/\rho$ on the surviving attention, parameterized by a reference surviving mass ρ, which restores the output scale of heavily draining heads and sharpens contextual attention elsewhere. SSA substantially lowers the normalized first-token share and redistributes attention to informative context tokens. SSA is also practical: it adds only one $n_h \times d$ projection per layer, and we have integrated it into FlashAttention-3 as a ready-to-use fused kernel, so that it inherits the efficiency of state-of-the-art attention kernels and incurs an end-to-end training slowdown of less than 1%. In extensive experiments on a from-scratch pre-trained Mixture-of-Experts model, SSA improves the fourteen-benchmark average by 2.74 points over Softmax attention, reaches the baseline's final pre-training loss with about $1.3\times$ fewer tokens, and never triggers the QK-clip safeguard that all baselines trigger hundreds to thousands of times.

1 INTRODUCTION

Softmax attention (Vaswani et al., 2017) is the computational core of modern large language models (LLMs), and as these models have grown in scale and context length, its behavior on long and often uninformative contexts has come under increasingly close scrutiny. One well-documented irregularity is the *attention sink* (Xiao et al., 2024; Gu et al., 2025): because the attention weights over the visible context are forced to sum to one, every query must spend a full unit of attention mass on preceding tokens even when none of them is informative, and models cope with this rigidity by dumping the surplus mass onto a small number of uninformative tokens, most often the very first one. Attention sink have been observed across model families and scales and matter in settings ranging from streaming inference (Xiao et al., 2024) to outlier features and low-bit quantization (Sun et al., 2024; Bondarenko et al., 2023; Dettmers et al., 2022; Xiao et al., 2023).

A growing line of work therefore relieves the sink by breaking the sum-to-one constraint. Sink Attention adds a learnable virtual bias to the Softmax denominator (Miller, 2023; Sun et al., 2024; OpenAI, 2025), and Gated Attention, adopted in recent production models (Qiu et al., 2025; Qwen Team, 2025), multiplies the attention output by a query-dependent sigmoid gate. Both reduce the

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.

absolute attention score placed on the first token and are therefore reported to alleviate the sink. However, the absolute score is the wrong yardstick. As shown in Section 3, both mechanisms—and, for fixed logits, any relaxation of this kind, including the one we propose—multiply all weights of a query by the same factor and therefore cannot change the *normalized* distribution over real tokens. The difference between mechanisms of this form lies not in the forward pass but in how the trained model learns to allocate its mass, and it must be measured after normalization. Measured this way, existing remedies remove only part of the sink: the first token still receives a substantial share of the normalized attention, and, because the rescaling hits every token, attention on genuinely informative tokens is suppressed as well, a side effect we call *collateral damage*.

Motivated by this analysis, we propose **Scaled Sink Attention (SSA)**, a lightweight modification of attention normalization that changes what the model learns. SSA introduces an *input-dependent virtual sink*: a projection of the current hidden state yields one additional sink logit per head, which competes with the real tokens inside the Softmax but whose probability is then discarded. Because the sink is a competing Softmax entry, the drained mass is decided by comparing a learned, token-specific threshold with the actual attention evidence—unlike a constant bias (one threshold for all tokens) or a post-Softmax gate (decoupled from the evidence). Draining mass, however, dilutes the surviving weights—the collateral damage of prior methods—so SSA rescales the surviving weights by a fixed gain $g = 1/\rho$, where ρ is a reference surviving mass (ρ=0.2, i.e., g=5, in all main runs): heads that keep a fraction ρ of their mass recover the output scale of Softmax, heads that keep less are attenuated, and heads that keep more have their contextual attention deliberately sharpened. In the language of gates, SSA is Gated Attention whose gate is tied to the attention evidence and allowed to exceed one. SSA adds a single $n_h \times d$ projection per attention layer, the same size as the gate projection of head-wise Gated Attention, and nothing else.

We validate SSA with extensive from-scratch pre-training experiments on a 20B-parameter (2B-activated) Mixture-of-Experts (MoE) LLM trained on 524B bilingual tokens, where all runs share the same architecture, data, and optimizer and differ only in the attention normalization. Our contributions are summarized as follows:

- **A normalization-invariant diagnosis and measurement protocol.** We show that per-query relaxations of Softmax cannot change the normalized attention distribution for fixed logits (Proposition 1), propose measuring the sink after normalization, and find on trained models that Sink Attention and Gated Attention reduce the normalized first-token share only from 0.54 to 0.33 and 0.21 while suppressing attention on informative tokens.
- **Scaled Sink Attention.** We propose a per-head, input-dependent virtual sink—an *evidence-aware* gate—combined with a fixed gain $g = 1/\rho$ on the surviving attention. SSA reduces the normalized first-token share to 0.10, improves the fourteen-benchmark average by $+2.74$ points over Softmax attention (largest gains on GSM8K and CMath, $+6.7$) while matching or exceeding Gated Attention, and reaches the baseline's final pre-training loss with $1.3\times$ fewer tokens (1.2–$1.4\times$ across smoothing windows). Beyond quality, SSA also markedly improves training stability—it never triggers the QK-clip safeguard, which Softmax, Sink and Gated Attention trigger 817, $4{,}433$ and $3{,}150$ times respectively —and we provide a detailed analysis and ablation of both components to attribute these gains.
- **A FlashAttention-3 implementation with negligible overhead.** We integrate SSA into FlashAttention-3 (Shah et al., 2024), folding the per-head virtual sink into the online-Softmax normalizer of the attention kernel (the fixed gain is a scalar multiply of the output) so that SSA inherits the efficiency of state-of-the-art attention kernels and can serve as a drop-in replacement in existing training stacks. On our 20B MoE model, the end-to-end training slowdown is below 1%.

2 Related Work

Attention sink and massive activations. Early analyses of BERT observed heads that park their attention on special tokens as a "no-op" (Clark et al., 2019; Kobayashi et al., 2020). Xiao et al. (2024) named the phenomenon the attention sink in autoregressive LLMs and showed that keeping sink tokens is necessary for stable streaming decoding, as do KV-cache eviction methods (Zhang et al., 2023; Han et al., 2024). Sun et al. (2024) connected sinks to *massive activations*—a few hidden dimensions of extreme magnitude that act as fixed biases—and proposed learnable explicit

attention biases (a virtual key and value) to remove them; Bondarenko et al. (2023) traced the resulting quantization outliers to heads that try to "do nothing" under the sum-to-one constraint. Gu et al. (2025) showed that sinks emerge from Softmax normalization and disappear under unnormalized or sigmoid attention; further work studies their mechanistic origin (Guo et al., 2024; Barbero et al., 2025; Cancedda, 2024; Elhage et al., 2021), vision counterparts (Darcet et al., 2024) and training-free calibration (Yu et al., 2024). Barbero et al. (2025) argue that sinks are functionally useful (a brake on over-mixing). We build on the normalization diagnosis of these works, but instead of preserving sink tokens we remove the pressure that creates them.

Relaxing the Softmax normalization. Adding a constant to the Softmax denominator ("off-by-one" or clipped Softmax) allows the visible-token mass to fall below one (Miller, 2023; Bondarenko et al., 2023), as does a persistent virtual key (Sukhbaatar et al., 2019; Sun et al., 2024); a learnable per-head version is used by gpt-oss (OpenAI, 2025) and is our Sink Attention baseline. Bondarenko et al. (2023) also proposed gating the attention output, and Qiu et al. (2025) showed that a sigmoid gate after Softmax improves non-linearity and sparsity and lowers the first-token score; this design is used in Qwen3-Next (Qwen Team, 2025) and is our Gated Attention baseline. Other normalizations remove the sink by construction: Softpick rectifies the Softmax numerator (Zuhri et al., 2025), stick-breaking attention replaces the Softmax with a sequential allocation (Tan et al., 2025), sigmoid attention drops normalization entirely (Ramapuram et al., 2025), and Differential Transformer subtracts two Softmax maps (Ye et al., 2025); Scalable-Softmax rescales logits with the context length (Nakanishi, 2025). Relative to the bias and gate families, SSA differs in two respects: its virtual sink is input-dependent and decided by competition with the attention evidence, and the surviving attention is rescaled with a gain rather than shrunk (Section 4).

Training stability and logit control. Growth of attention logits is a well-documented source of instability (Wortsman et al., 2024; Zhai et al., 2023; Liu et al., 2020). Query–key normalization (Henry et al., 2020; Dehghani et al., 2023), logit soft-capping (Gemma Team, 2024) and QK-clip in MuonClip (Kimi Team, 2025; Liu et al., 2025) bound or rescale the pre-Softmax scores, and low-precision training makes well-behaved activations even more important (Fishman et al., 2024). These techniques cap large logits externally; SSA reduces them intrinsically, and the QK-clip safeguard present in all our runs is never triggered by SSA.

3 Preliminaries: What Per-Query Rescaling Can and Cannot Do

Notation. Consider one attention head at query position i with (normalized) input hidden state $\mathbf{x}_i \in \mathbb{R}^d$ and visible positions $j = 0, \dots, i$. Linear projections give queries $\mathbf{q}_i$, keys $\mathbf{k}_j$ and values $\mathbf{v}_j$, the logits $z_{i,j} = \mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d_k}$, and the Softmax weights and output

$$a_{i,j} = \frac{\exp(z_{i,j})}{\sum_{l=0}^{i} \exp(z_{i,l})}, \qquad \mathbf{o}_i = \sum_{j=0}^{i} a_{i,j}\, \mathbf{v}_j, \qquad \sum_{j=0}^{i} a_{i,j} = 1. \tag{1}$$

The constraint $\sum_j a_{i,j} = 1$ forces the model to allocate attention even when no preceding token is informative; empirically the surplus is dumped onto the first token(s). A layer has n_h heads; head indices h are omitted when clear from context.

3.1 Existing Mitigations Are Per-Query Rescalings

Two recent families relax the sum-to-one constraint. **Sink Attention** adds a learnable per-head bias b (a virtual key with no value) to the denominator,

$$a_{i,j}^{\text{sink}} = \frac{\exp(z_{i,j})}{\exp(b) + \sum_{l=0}^{i} \exp(z_{i,l})} = c_i^{\text{sink}}\, a_{i,j}, \qquad c_i^{\text{sink}} = \sigma\big(\operatorname{LSE}(\mathbf{z}_i) - b\big), \tag{2}$$

where $\operatorname{LSE}(\mathbf{z}_i) = \log \sum_{l \le i} \exp(z_{i,l})$ and σ is the logistic function. **Gated Attention** applies a head-wise sigmoid gate $\gamma_i = \sigma(\mathbf{w}_\gamma^\top \mathbf{x}_i) \in (0, 1)$ to the attention output, which is equivalent to rescaling every weight,

$$\mathbf{o}_i^{\text{gate}} = \gamma_i \sum_j a_{i,j} \mathbf{v}_j \iff a_{i,j}^{\text{gate}} = \gamma_i\, a_{i,j}. \tag{3}$$

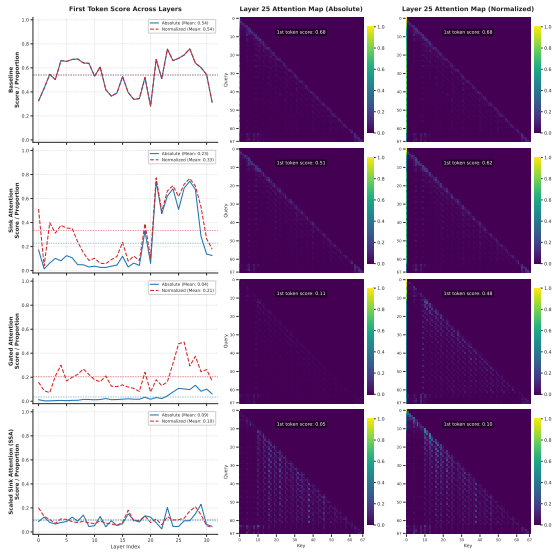


Figure 1: Absolute versus normalized attention on our trained 20B models (one row per method). **Left:** first-token attention across layers, absolute (blue) and normalized over real tokens (red); layer means are given in the legends. **Middle/right:** layer-25 attention maps before and after normalization. For SSA the absolute scores include the gain g.

Both reduce the absolute first-token weight $a_{i,0}$, but in both cases the factor multiplying $a_{i,j}$ depends on the query i and not on the key j.

Proposition 1 (Invariance of the normalized distribution under per-query rescaling)**.** *Let $a_{i,j}$ be any attention distribution with normalized form $\hat{a}_{i,j} = a_{i,j}/\sum_l a_{i,l}$, and let $c_i > 0$ be a scalar independent of j. The rescaled weights $a'_{i,j} = c_i\, a_{i,j}$ have the same normalized distribution: $\hat{a}'_{i,j} = \frac{c_i a_{i,j}}{\sum_l c_i a_{i,l}} = \hat{a}_{i,j}$ for all j.*

Sink Attention ($c_i = c_i^{\text{sink}}$), Gated Attention ($c_i = \gamma_i$), and—as we make explicit in Section 4.1—our own virtual sink are all of this form (a proof and further remarks are given in Appendix E). Two consequences follow. First, no relaxation of this kind can change the relative weight placed on the sink token for fixed logits, so a low absolute first-token score says nothing about whether the model still relies on the sink. Second, any genuine sink reduction must come from the logits the trained model learns, and the right diagnostic is the attention distribution after normalization.

3.2 Measuring the Sink after Normalization

We therefore measure the *normalized attention* over real tokens, $\hat{a}_{i,j} = a_{i,j}/(\sum_{l=0}^{i} a_{i,l} + \epsilon)$, which is invariant to per-query rescaling. Figure 1 contrasts absolute and normalized first-token scores on our trained 20B models (Appendix A); all numbers below are means over the heads and query positions of a probe passage (Appendix D), averaged over layers. Softmax attention spends 0.54 of its mass on the first token. Sink Attention and Gated Attention show low absolute first-token scores (0.23 and 0.04), but their normalized shares are 0.33 and 0.21: they reduce the sink, yet by far less than the absolute numbers suggest—a fifth to a third of the attention that reaches real tokens is still spent on the first token, and the first-token stripe re-appears in the normalized maps of deep layers. SSA reduces the normalized share to 0.10 (absolute 0.09, including the gain g). In other words, SSA removes most of the attention sink: the share of the real-token attention spent on the first token drops by more than a factor of five, and the first-token stripe that persists in the normalized maps of the prior methods is absent for SSA (Figure 1, right).

3.3 Collateral Damage: Suppression of Contextual Attention

Per-query rescaling has a second cost. Because the factor c_i multiplies every weight, attention on the informative context ($j \geq 1$) is suppressed together with the sink. We quantify this with the *remaining*

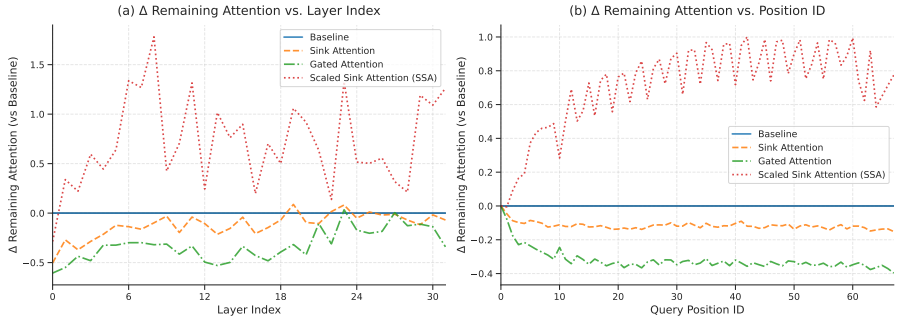


Figure 2: Change in the remaining attention on real context tokens ($j \geq 1$), $\Delta\mathcal{R}_i$ relative to the Softmax baseline, (a) across layers and (b) across query positions. Sink Attention and Gated Attention suppress the mass on informative tokens ($\Delta\mathcal{R}_i < 0$); SSA (including g) increases it. Absolute values are shown in Appendix J.

attention $\mathcal{R}_i = \sum_{j=1}^{i} a_{i,j}$, the total mass on real tokens other than the first. Figure 2 plots the change $\Delta\mathcal{R}_i$ relative to the Softmax Baseline: Sink Attention and Gated Attention are negative across layers and positions, i.e., they weaken the interaction between semantic tokens, which is consistent with their negligible loss advantage over the Softmax Baseline (Section 5.1). SSA is positive, and this is largely the gain g by construction: before the gain, SSA keeps on average only about 0.30 of its mass on real tokens, of which a tenth goes to the first token (Appendix G), less than the 0.46 that Softmax leaves for tokens $j \geq 1$. The substantive evidence that SSA moves mass from the sink token to the context is the normalized first-token share ($0.54 \to 0.10$); the gain then turns the redistributed mass into a net increase over Softmax.

4 Scaled Sink Attention

4.1 The Input-Dependent, Per-Head Virtual Sink

SSA relaxes the sum-to-one constraint with one *virtual sink* slot per head that competes with the real tokens inside the Softmax but is discarded afterwards; its logit is a learned linear function of the current hidden state. For a layer with n_h heads and model width d we introduce a projection matrix $\mathbf{W}_{\text{sink}} \in \mathbb{R}^{n_h \times d}$ whose h-th row $\mathbf{w}_{\text{sink}}^{(h)}$ produces the sink logit of head h:

$$z_{i,\text{sink}}^{(h)} = \mathbf{w}_{\text{sink}}^{(h)\top}\mathbf{x}_i, \qquad a_{i,j}^{(h)} = \frac{\exp(z_{i,j}^{(h)})}{\exp(z_{i,\text{sink}}^{(h)}) + \sum_{l=0}^{i}\exp(z_{i,l}^{(h)})}, \quad j = 0, \ldots, i. \tag{4}$$

The sink slot receives probability $p_{i,\text{sink}}^{(h)}$ but has no value vector, so the mass that survives on real tokens is

$$m_i^{(h)} = \sum_{j=0}^{i} a_{i,j}^{(h)} = 1 - p_{i,\text{sink}}^{(h)} = \sigma\big(\,\text{LSE}(\mathbf{z}_i^{(h)}) - z_{i,\text{sink}}^{(h)}\big) \;\leq\; 1, \tag{5}$$

and the output is formed from real values only, $\mathbf{o}_i^{(h)} = \sum_{j \leq i} a_{i,j}^{(h)}\mathbf{v}_j^{(h)}$.

An evidence-aware gate. Equation equation 5 makes the relation to the prior methods precise: the virtual sink is a per-query factor m_i, so Proposition 1 applies to it as well, and its effect on the sink must come through what the model learns. What differs is how m_i is determined. Gated Attention's factor $\gamma_i = \sigma(\mathbf{w}_\gamma^\top \mathbf{x}_i)$ is decoupled from the attention scores: the gate can shrink the output, but the Softmax must still place its mass somewhere, so nothing discourages a learned sink token. Sink Attention's factor $\sigma(\text{LSE}(\mathbf{z}_i) - b)$ compares the evidence with a threshold, but with the same threshold for every token; to drain on uninformative tokens yet retain mass on informative ones, the model can only separate the two through the scale of its logits, which pushes them upward. SSA compares the evidence with a token-specific threshold $z_{i,\text{sink}} = \mathbf{w}_{\text{sink}}^\top \mathbf{x}_i$: a query without a good match raises its threshold and drains, a query with a good match keeps its mass, and neither

requires inflating the logits. This is our hypothesis for why, among structurally similar mechanisms, only the input-dependent sink both lowers the normalized first-token share substantially and reduces logit growth, whereas Sink Attention increases the number of QK-clip events five-fold over Softmax (Section 5.1). The learned thresholds are strongly head- and depth-dependent (Appendix G).

Sink Attention is the special case $\mathbf{W}_{\text{sink}} = \mathbf{0}$ plus a bias; adding such a bias to our sink hurts (Section 5.2), so SSA uses a pure, zero-initialized projection, which starts as a mild uniform relaxation and is switched on only where the gradient consistently asks for it (Appendix E). The explicit attention bias of Sun et al. (2024) (a virtual key $\mathbf{k}'$ with a value $\mathbf{v}'$) and persistent memory vectors (Sukhbaatar et al., 2019) also yield a query-dependent sink logit $\mathbf{q}_i^\top \mathbf{k}'$; ours is instead a direct linear function of the hidden state (width d rather than d_k), not tied to the query geometry used for content matching, and its slot carries no value, so draining changes the output magnitude but not its direction. We call the mechanism up to this point, i.e., SSA without the gain, **Adaptive Sink Attention (ASA)**; it serves as an ablation throughout.

4.2 Rescaling the Surviving Attention with a Fixed Gain

Draining mass dilutes the surviving weights by $m_i \leq 1$, which on its own is precisely the collateral damage of prior methods. SSA counteracts it—and goes further, actively sharpening contextual attention—by rescaling the surviving weights with a fixed gain $g > 1$:

$$\tilde{a}_{i,j}^{(h)} = g \cdot a_{i,j}^{(h)}, \qquad \mathbf{o}_i^{(h)} = g \sum_{j=0}^{i} a_{i,j}^{(h)} \, \mathbf{v}_j^{(h)}, \qquad g = \frac{1}{\rho}. \tag{6}$$

We parameterize the gain by a *reference surviving mass* $\rho \in (0, 1]$: $g = 1/\rho$ is exactly the factor at which a head that keeps a fraction ρ of its mass on real tokens recovers the output scale of vanilla Softmax; heads that keep less are attenuated and heads that keep more are amplified (exact per-query conservation, $1/m_i$, would recover vanilla Softmax). Our default, used in every main comparison, is $\rho = 0.2$, i.e. $g = 5$. In the gate view, SSA is Gated Attention with the evidence-aware gate $g\,\sigma(\mathrm{LSE}(\mathbf{z}_i) - z_{i,\text{sink}})$, which can exceed one: the virtual sink lets the real-token mass fall below one, while the gain sets the magnitude of the residual-stream update.

Choice of ρ. We fixed ρ from two angles. The first is the measured sink usage of the trained model: in the gain-free variant, one third of the heads drain at least 80% of their mass, so $\rho = 0.2$ ($g = 5$) conserves the output scale of these heavily draining heads in expectation and amplifies the remaining heads by at most a factor of five, whereas the gain that conserves the mean head ($g \approx 2.8$) would attenuate more than half of them (Appendix E). The second is downstream performance: a fixed $g = 5$ outperforms $g = 10$, which over-compensates a quarter of the heads by more than five-fold, and matches or exceeds learnable and input-dependent gains (Section 5.2). A mechanistic hypothesis for why the gain helps—it lowers the attention sharpness, and hence the logit magnitude, a head needs for an output of a given size—is discussed in Appendix H.

Implementation and cost. SSA adds one $n_h \times d$ matrix per attention layer (16×2048 parameters per layer, 1.0M in total, 0.005% of our 20B model) and one extra GEMV per token. Inside attention the sink logit is folded into the running log-sum-exp of the online Softmax, so no extra key column is materialized. Our implementation extends FlashAttention-3 (Shah et al., 2024) to accept a per-(token, head) sink logit (Appendix C); the end-to-end training slowdown is below 1%.

5 Experiments

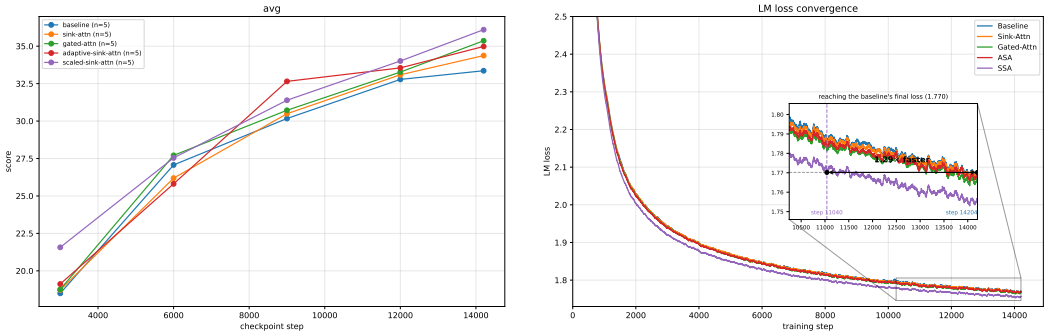
Setup. All main experiments evaluate a 20B-parameter Mixture-of-Experts (MoE) model trained from scratch under identical conditions on 524B tokens; full details regarding the architecture, training recipe, compared methods, and evaluation protocol are deferred to Appendices A–D.

5.1 Main Results

SSA attains the highest average score. Table 1 reports results after 524B tokens and Figure 4(a) visualizes the per-benchmark gains over the Baseline. SSA attains an average of 36.10, $+2.74$

Table 1: Results after 524B training tokens. The vertical rule separates prior methods from ours (shaded columns); best per row in **bold**. **Avg** (shaded rows) is the unweighted mean over the fourteen benchmarks; results at all five checkpoints are in Appendix P.

Domain	Benchmark	Baseline	Sink Attention	Gated Attention	ASA (ours)	SSA (ours)
Knowledge & Reasoning	MMLU	56.98	57.22	**58.33**	57.14	57.48
	MMLU-Pro	26.04	28.91	**30.41**	29.27	27.91
	CMMLU	48.55	49.52	49.93	49.85	**50.92**
	C-Eval	49.01	**50.29**	47.55	50.02	49.15
	GPQA	21.88	22.77	24.55	**25.89**	22.99
	SuperGPQA	14.17	**15.21**	15.12	15.16	14.92
Math	GSM8K	48.98	50.11	51.18	53.15	**55.72**
	MATH	23.70	24.00	23.94	20.64	**26.88**
	CMath	54.46	57.10	56.10	53.64	**61.11**
Code	HumanEval	44.51	41.46	46.95	45.12	**49.39**
	CRUXEval-I	25.12	34.00	34.12	**38.94**	33.12
	CRUXEval-O	28.12	**28.25**	27.88	26.14	27.75
	LiveCodeBench	8.72	9.38	**10.24**	8.72	9.57
	DS-1000	16.80	13.00	**18.70**	16.00	18.50
Avg (14 benchmarks)		33.36	34.37	35.36	34.98	**36.10**



(a) Average score versus training tokens.

(b) Next-token LM loss versus training tokens.

Figure 3: Training dynamics. **(a)** Fourteen-benchmark average at five checkpoints (thirteen benchmarks at 327B, see Appendix P). **(b)** Loss smoothed with a centered 51-iteration window; the inset marks where SSA reaches the baseline's final loss.

points over the Baseline and $+0.74$ over the strongest prior method, Gated Attention (35.36). The virtual sink alone (ASA, 34.98) contributes $+1.62$ and the gain (ASA→SSA) a further $+1.12$. SSA beats the Baseline on 13 of 14 benchmarks (two-sided sign test $p = 0.002$), and a bootstrap over benchmarks places its mean gap to the Baseline at $[+1.5, +4.2]$ (95% CI). Therefore, SSA matches or exceeds the strongest prior method.

The largest gains are on mathematical reasoning. SSA is best on all three math benchmarks: GSM8K $48.98 \to 55.72$ ($+6.74$), MATH $23.70 \to 26.88$ ($+3.18$), CMath $54.46 \to 61.11$ ($+6.65$). A plausible reading is that multi-step arithmetic depends on precise long-range token interactions, which the increased remaining attention strengthens (Figure 2); consistently, the prior methods, which suppress it, gain little on math. On knowledge and code, SSA is best on CMMLU and HumanEval and competitive elsewhere.

SSA converges faster. In Figure 3a, SSA shows a pronounced head start—21.57 after 101B tokens, about 2.5 points above every other method—and is the best method at four of the five checkpoints, while the Baseline plateaus over the last 83B tokens ($32.78 \to 33.36$). The pre-training loss (Figure 3b) tells the same story: SSA stays below all other methods throughout and reaches the Baseline's final loss of 1.770 after 404B tokens, 119B tokens (23%) earlier, a $1.30\times$ reduction in the

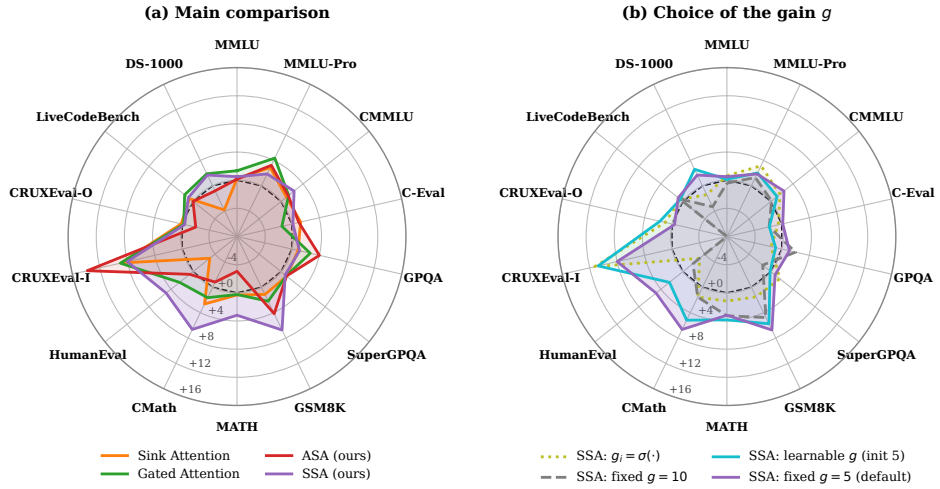


Figure 4: Per-benchmark score difference to the Softmax Baseline (points) after 524B tokens; the dashed circle marks zero. **(a)** Prior remedies and our two variants. **(b)** Choice of the gain on top of the default virtual sink (fixed $g = 10$ is clipped at the plot boundary on the two CRUXEval axes, where it collapsed to 0.5 and 1.4 points). Radar charts for every checkpoint and for the virtual-sink design are given in Appendices L and K.

tokens needed to reach the same loss. The speed-up grows over training, from about $1.1\times$ early on to 1.2–$1.3\times$ in the second half (Appendix M). Sink Attention and Gated Attention offer no visible loss advantage over the Baseline, consistent with their suppressed contextual attention; their benchmark gains ($+1.0$ and $+2.0$) therefore come without a lower loss, whereas SSA improves both. The sink alone (ASA) lowers the loss by only about 0.003, so the loss improvement of SSA is attributable mostly to the gain; the stability improvement, discussed next, is attributable mostly to the sink.

SSA trains more stably. All runs monitor the maximum raw QK logit and clip it whenever it exceeds $\tau = 1385$ (Appendix F). The Baseline exceeds τ in 817 of 14,204 iterations, and the prior remedies are markedly worse: Sink Attention in 4,433 and Gated Attention in 3,150, i.e. their logits sit at the clip threshold for a fifth to a third of training. The virtual sink alone (ASA) reduces the count to 63 and SSA never triggers the clip: its maximum logit peaks at 933, a third below τ, and settles at a final-quarter median of 692, roughly half that of Sink and Gated Attention. Most of the stability gain is therefore due to the sink and the remainder to the gain, the mirror image of the loss attribution above; Appendices F and H analyze where the outliers come from.

5.2 Ablations

Table 4 (Appendix K) ablates the two components of SSA; Figure 4(b) shows the per-benchmark picture for the gain and Figure 12 the trajectories over training.

Virtual-sink design. The simplest parameterization—a zero-initialized projection with no bias—is the best (34.98), ahead of random initialization (34.02), which injects an arbitrary sink that must be unlearned, and of adding a learnable per-head bias (33.27, below the Baseline and unstable on MATH). The bias adds an input-independent threshold—the Sink Attention mechanism—that interferes with the input-dependent one rather than helping it.

Gain. The fixed gain $g = 5$ is the strongest at the final checkpoint (36.10; best among the gain variants on C-Eval, CMath, GSM8K and HumanEval), followed by a learnable per-head gain initialized at 5 (35.73); the five input-dependent gates (σ, 2σ, $1 + \sigma$, SiLU, $\sqrt{\text{softplus}}$; Table 5) land between 33.46 and 34.61, i.e. close to ASA. Doubling the gain to $g = 10$ ($\rho = 0.1$) is competitive early but falls behind from 441B tokens (33.52 vs. 34.01) and collapses on both CRUXEval tasks at the final checkpoint; even over the other twelve benchmarks it trails $g = 5$ by 2.2 points, in line with the mass-conservation argument of Appendix E (over-compensation far beyond the measured sink mass). That the learnable gain does not improve on its own initialization is consistent with the

view that a fixed $g > 1$ acts as a *structural prior*: it forces the network to use the virtual sink to keep its outputs bounded, whereas an adaptive gain can shrink itself and revert to the uniform-rescaling regime of Gated Attention. A fixed $g = 5$ is thus at least as effective as any adaptive gain and the most lightweight choice.

5.3 Results at Larger Scale

To evaluate whether the advantage persists at a larger scale, we trained Gated Attention and SSA on a substantially larger MoE model with identical data and initialization (approximately 264B total and 12B activated parameters; Appendix B). Both models were trained for 1,500 iterations (44B tokens) and evaluated at this checkpoint (Appendix N). SSA is ahead on average (15.26 vs. 13.51 over twelve benchmarks) and on 11 of the 12 benchmarks, including every math and code benchmark, most visibly DS-1000 (+3.9), HumanEval (+3.1) and MATH (+2.3). Overall, these results demonstrate that the effectiveness of SSA scales successfully to larger models. Combined with the 20B experiments, this confirms that our method works consistently across different model sizes and is a highly scalable approach.

Table 2: RULER accuracy (%) at 128K context after 50B tokens of continued pre-training at sequence length 131,072. NIAH averages the eight needle-in-a-haystack tasks, Aggregation the common- and frequent-words extraction tasks, QA the two question-answering tasks; **Avg** is the unweighted mean over all thirteen tasks. Best per column in **bold**; per-task scores are in Table 15.

Method	NIAH (8 tasks)	Variable tracking	Aggregation (2 tasks)	QA (2 tasks)	**Avg** (13 tasks)
Baseline	85.27	61.76	29.85	21.90	65.18
Gated Attention	86.79	71.04	**37.52**	10.90	66.32
SSA (ours)	**89.03**	**75.44**	35.95	**30.80**	**70.86**

5.4 Long-Context Continued Pre-Training

Table 2 reports RULER accuracy at 128K context by task group; per-task scores and a detailed analysis are given in Appendix O. SSA attains the highest average over all thirteen tasks (70.86), outperforming Gated Attention (66.32) by 4.5 points and the Baseline (65.18) by 5.7 points. SSA leads three of the four task groups, with the most pronounced gains on tasks that require integrating evidence from multiple context positions throughout the document. For instance, on question answering, Gated Attention loses half of the Baseline's accuracy (10.90 vs. 21.90), whereas SSA significantly improves upon it to reach 30.80. Similarly, SSA establishes wide margins on variable tracking (75.44 vs. 71.04 for Gated and 61.76 for Baseline) and multi-key retrieval (78.13 vs. 71.93 and 63.53), where models must accurately follow assignment chains and reject distractors. As analyzed in Section 3, prior mitigations broadly suppress contextual attention and consequently struggle on these complex integration tasks, whereas SSA's increased remaining attention avoids this collateral damage and successfully preserves long-range token interactions.

6 Conclusion

In this paper, we demonstrated that existing remedies for the attention sink act as per-query rescalings that leave the normalized attention distribution unchanged for fixed logits, ultimately suppressing attention on informative tokens. To address this, we proposed Scaled Sink Attention (SSA), which introduces a lightweight, input-dependent virtual sink to drain attention mass selectively, paired with a fixed gain to restore and sharpen contextual attention. SSA adds negligible parameters and compute overhead, readily integrating into FlashAttention-3. Across extensive from-scratch pre-training experiments on a 20B MoE model, SSA largely eliminates the attention sink, significantly improves downstream performance, accelerates convergence by $1.3\times$, and uniquely eliminates QK-clip training instabilities. These benefits persist when scaling to a 264B-parameter model and during long-context continued pre-training, establishing SSA as a highly effective, efficient, and scalable drop-in replacement for standard Softmax attention.

AI USE STATEMENT

In this work, we have not used generative AI tools for generating synthetic datasets, developing theoretical models, formulating or proving mathematical claims, designing experiments, implementing methods, or interpreting results. We strictly used generative AI tools (specifically, Claude Fable 5.1 and Gemini 3.1 Pro Preview) solely to aid and polish the writing of this manuscript. Specifically, the tools were used to edit the research paper to improve readability, correct grammar, and refine the language. We have thoroughly reviewed all AI-assisted work by carefully proofreading the polished text to ensure it accurately reflects our original scientific intent and meaning. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work studies the attention normalization of language models and proposes an architectural modification evaluated on standard public benchmarks. The pre-training corpus is a general web-scale bilingual text corpus; no human subjects, personal data or sensitive attributes are involved, and no new datasets are released. SSA does not introduce risks beyond those already associated with training large language models, and more stable, outlier-free training may reduce the compute wasted on failed runs.

REPRODUCIBILITY STATEMENT

We have made extensive efforts to ensure the reproducibility of our work. The method is fully specified by Equations equation 4–equation 6 and the design choices in Section 4; a step-by-step description of the FlashAttention-3 kernel modification and PyTorch-style pseudo-code are given in Appendix C. The experimental setup is described in Appendix A; all architectural and optimization hyper-parameters of the main, larger-scale and long-context runs are listed in Appendix B, the evaluation protocol in Appendix D, complete per-checkpoint results for every method and ablation in Appendices K and P, and per-task long-context results in Appendix O. The token accounting used for all figures is described in Appendix B, the proofs and derivations in Appendices E and H, and the statistical procedures (sign test, bootstrap) in Appendix L. Our FlashAttention-3 modifications, training scripts, evaluation configurations, and plotting code will be released as anonymized supplementary material.

REFERENCES

Federico Barbero, Álvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do LLMs attend to the first token? In *Conference on Language Modeling (COLM)*, 2025. URL https://arxiv.org/abs/2504.02732.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2306.12929.

Nicola Cancedda. Spectral filters, dark signals, and attention sinks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://arxiv.org/abs/2402.09221.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019. URL https://arxiv.org/abs/1906.04341.

Karl Cobbe, Vineet Kosaraju, Mohammad 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 preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2205.14135.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.16588.

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024a. URL https://arxiv.org/abs/2405.04434.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024b. URL https://arxiv.org/abs/2412.19437.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2302.05442.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2208.07339.

Bradley Efron and Robert J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall/CRC, 1994. URL https://doi.org/10.1201/9780429246593.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://arxiv.org/abs/2101.03961.

Maxim Fishman, Brian Chmiel, Ron Banner, and Daniel Soudry. Scaling FP8 training to trillion-token LLMs. *arXiv preprint arXiv:2409.12517*, 2024. URL https://arxiv.org/abs/2409.12517.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2404.19737.

Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I. Wang. CRUXEval: A benchmark for code reasoning, understanding and execution. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2401.03065.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.10781.

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 preprint arXiv:2410.13835*, 2024. URL https://arxiv.org/abs/2410.13835.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. LM-Infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024. URL https://arxiv.org/abs/2308.16137.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021b. URL https://arxiv.org/abs/2103.03874.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP*, 2020. URL https://arxiv.org/abs/2010.04245.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, et al. C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.08322.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination-free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2403.07974.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. Blog post, 2024. URL https://kellerjordan.github.io/posts/muon/.

Kimi Team. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025. URL https://arxiv.org/abs/2507.20534.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://arxiv.org/abs/2004.10102.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. DS-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2211.11501.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL*, 2024. URL https://arxiv.org/abs/2306.09212.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025. URL https://arxiv.org/abs/2502.16982.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://arxiv.org/abs/2004.08249.

M-A-P Team. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025. URL https://arxiv.org/abs/2502.14739.

Maxim Milakov and Natalia Gimelshein. Online normalizer calculation for softmax. *arXiv preprint arXiv:1805.02867*, 2018. URL https://arxiv.org/abs/1805.02867.

Evan Miller. Attention is off by one, 2023. URL https://www.evanmiller.org/attention-is-off-by-one.html.

Ken M. Nakanishi. Scalable-Softmax is superior for attention. *arXiv preprint arXiv:2501.19399*, 2025. URL https://arxiv.org/abs/2501.19399.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.06708.

Qwen Team. Qwen3-Next: Towards ultimate training & inference efficiency, 2025. URL https://qwen.ai/blog?id=4074cca80393150c248e508aa62983f9cb7d27cd.

Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russ Webb. Theory, analysis, and best practices for sigmoid self-attention. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2409.04431.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2311.12022.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2407.08608.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL https://arxiv.org/abs/2002.05202.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1701.06538.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. URL https://arxiv.org/abs/1909.08053.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. URL https://arxiv.org/abs/2104.09864.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Hervé Jégou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019. URL https://arxiv.org/abs/1907.01470.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2402.17762.

Shawn Tan, Yikang Shen, Songlin Yang, Aaron Courville, and Rameswar Panda. Scaling stick-breaking attention: An efficient implementation and in-depth study. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.17980.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024a. URL https://arxiv.org/abs/2408.15664.

Yubo Wang, Xueguang Ma, Ge Zhang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b. URL https://arxiv.org/abs/2406.01574.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. CMATH: Can your language model pass Chinese elementary school math test? *arXiv preprint arXiv:2306.16636*, 2023. URL https://arxiv.org/abs/2306.16636.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D. Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.14322.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2211.10438.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.17453.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.05258.

Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan Celine Lin. Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2406.15765.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2303.06296.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. URL https://arxiv.org/abs/1910.07467.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2306.14048.

Zayd M. K. Zuhri, Erland Hilman Fuadi, and Alham Fikri Aji. Softpick: No attention sink, no massive activations with rectified softmax. *arXiv preprint arXiv:2504.20966*, 2025. URL https://arxiv.org/abs/2504.20966.

APPENDIX OVERVIEW

The appendix is organized as follows. Appendix A describes the experimental setup of the main study (model, training recipe, compared methods and benchmarks); Appendix B lists all training configurations and the token accounting; Appendix C describes the FlashAttention-3 kernel modification, reports its throughput and gives pseudo-code; Appendix D details the evaluation protocol. Appendix E collects proofs and derivations (Proposition 1, the surviving-mass identity, the gradient of the sink logit, the mass-conservation argument behind $\rho = 0.2$ and the implied output scale). Appendices F–J analyze the trained models: the training-stability results (maximum QK logits and QK-clip events), how the virtual sink is used across heads and depth, where QK-logit outliers come from and our hypothesis for why the gain lowers them, their relation to attention sharpness, and additional attention visualizations. Appendix K contains the ablation tables (including every ablation variant at every checkpoint), Appendix L additional downstream analyses (per-checkpoint radar charts, per-benchmark trajectories, domain aggregates, rank statistics, bootstrap intervals and checkpoint-to-checkpoint volatility), and Appendix M additional loss and stability analyses. Appendix N reports the per-benchmark results of the larger-scale experiment of Section 5.3, Appendix O the per-task RULER results and analysis of Section 5.4, and Appendix P the full per-checkpoint results of the main comparison.

A EXPERIMENTAL SETUP

This appendix details the setup summarized at the beginning of Section 5.

Model. All main experiments use a frontier-style MoE Transformer (Shazeer et al., 2017; Fedus et al., 2022; DeepSeek-AI, 2024b; Yang et al., 2025) with 20.2B total and 2.1B activated parameters: 32 layers, hidden size 2048, RMSNorm (Zhang & Sennrich, 2019), SwiGLU (Shazeer, 2020), decoupled RoPE (Su et al., 2024), Multi-head Latent Attention (MLA) (DeepSeek-AI, 2024a) with 16 heads (128-dimensional content and 64-dimensional RoPE key components), one dense layer followed by 31 MoE layers routing each token to 4 of 128 experts plus one shared expert with a sigmoid router (Wang et al., 2024a), and one multi-token-prediction head (Gloeckle et al., 2024). Full details are in Appendix B.

Training. Models are trained from scratch in Megatron-LM (Shoeybi et al., 2019) on a bilingual (English and Chinese) corpus with sequence length 4096 and a global batch of 9216 sequences (37.7M tokens; ramped up from 3072 over the first 10B tokens). Matrix parameters are updated with Muon (Jordan et al., 2024; Liu et al., 2025) and vectors, scalars and embeddings with AdamW; weight decay is 0.1, precision bfloat16, and the learning rate is warmed up over 3B tokens to 10^{-3} and then held constant for the studied budget. All runs enable the QK-clip safeguard of Muon-Clip (Kimi Team, 2025) with threshold $\tau = 1385$ on the raw (unscaled) logit $\mathbf{q}_i^\top \mathbf{k}_j$, equivalent to the customary $\tau = 100$ after $1/\sqrt{d_k}$ scaling. All variants share identical architecture, data order and hyper-parameters and differ only in the attention normalization; all models are trained for 14,204 iterations, i.e. 524B tokens, and evaluated at 101, 214, 327, 441 and 524B tokens.

Compared methods. (i) vanilla Softmax attention (**Baseline**); (ii) **Sink Attention**, Equation equation 2; (iii) **Gated Attention**, Equation equation 3, in its head-wise form so that the gate projection has the same size as our sink projection; (iv) **ASA**, our virtual sink without the gain (g=1); and (v) **SSA**, our full method with $\rho = 0.2$ (g=5).

Evaluation. We evaluate on fourteen benchmarks in three domains. *Knowledge & reasoning*: MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024b), CMMLU (Li et al., 2024), C-Eval (Huang et al., 2023), GPQA (Rein et al., 2024), SuperGPQA (M-A-P Team, 2025). *Math*: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), CMath (Wei et al., 2023). *Code*: HumanEval (Chen et al., 2021), CRUXEval-I/-O (Gu et al., 2024), LiveCodeBench (Jain et al., 2025), DS-1000 (Lai et al., 2023). We report accuracy (Pass@1 for code generation) and the unweighted average over the fourteen benchmarks.

B TRAINING DETAILS

Table 3 lists the configurations of the three training settings. The main and larger-scale runs share the data pipeline and differ only in model size and peak learning rate; the long-context runs continue from the 524B-token checkpoints of the corresponding main runs.

Table 3: Training configurations. "Main" is the 20B-parameter setting of Appendix A used in Sections 5.1–5.2; "Large" the setting of Section 5.3; "Long-context" the continued pre-training of Section 5.4. Parameter counts are computed from the architecture.

	Main (20B-A2B)	Large	Long-context (from Main)
Layers / hidden size / dense FFN size	32 / 2048 / 7168	43 / 4096 / 12288	32 / 2048 / 7168
Attention heads	16	64	16
MLA q-LoRA / kv-LoRA rank	2048 / 512	1536 / 512	2048 / 512
QK head dim (content + RoPE) / V head dim	128 + 64 / 128	128 + 64 / 128	128 + 64 / 128
Dense layers / MoE layers	1 / 31	3 / 40	1 / 31
Routed experts (top-k) / shared experts	128 (4) / 1	256 (6) / 1	128 (4) / 1
Expert intermediate size	768	2048	768
Router	sigmoid, seq. aux. loss 10^{-4}, bias update 10^{-3}	same	same
MTP layers / loss weight	1 / 0.3	1 / 0.3	1 / 0.3
Vocabulary size	170,624	170,624	170,624
Total / activated parameters	20.2B / 2.1B	≈264B / ≈12B	20.2B / 2.1B
Sequence length	4,096	4,096	131,072
Global batch (sequences)	9,216 (ramp from 3,072)	9,216 (ramp from 3,072)	256
Tokens per iteration (after ramp-up)	37.7M	37.7M	33.5M
Iterations / tokens	14,204 / 524B	1,500 / 44B	1,500 / 50B
Optimizer (matrices / others)	Muon (mom. 0.95, 10 NS steps, RMS 0.2) / AdamW	same	same
Peak LR / min LR	10^{-3} / – (constant)	5×10^{-4} / –	10^{-3} / 10^{-4} (cosine)
Warm-up	3B tokens	3B tokens	none
Weight decay / grad clip	0.1 / 1.0	0.1 / 1.0	0.1 / 1.0
RoPE base	10^4	10^4	3.2×10^7
QK-clip threshold τ (raw logit)	1385	1385	1385
Precision	bf16	bf16	bf16
Parallelism (TP / PP / EP / CP)	1 / 2 / 8 / 1	1 / 4 / 8 / 1	1 / 2 / 8 / 8

Method-specific settings. Sink Attention adds one learnable scalar per head to the Softmax denominator, initialized at 0 (updated with AdamW). Gated Attention uses a head-wise gate projection $\mathbf{W}_\gamma \in \mathbb{R}^{n_h \times d}$ followed by a sigmoid, applied to the attention output before the output projection (Qiu et al., 2025). ASA and SSA use the zero-initialized, bias-free sink projection $\mathbf{W}_{\text{sink}} \in \mathbb{R}^{n_h \times d}$ of Equation equation 4; SSA multiplies the attention output by the gain $g = 1/\rho = 5$ before the output projection. The learnable-gain ablation replaces the fixed gain by one learnable scalar per head initialized at 5 (AdamW); the input-dependent ablations reuse the gate projection of Gated Attention with the indicated activation. All variants are otherwise identical.

Token accounting. Megatron-LM ramps the global batch from 3,072 to 9,216 sequences in two increments of 3,072, each held for 1,221,120 samples. Consequently iterations 3,000, 6,000, 9,000, 12,000 and 14,204 correspond to 100.7, 214.0, 327.2, 440.5 and 523.7B tokens, and the loss crossing at iteration 11,040 (inset of Figure 3b) to 404.2B tokens. All figures in the paper use this mapping. The plotted LM loss is the next-token loss and excludes the multi-token-prediction term.

C IMPLEMENTATION INSIDE FLASHATTENTION-3

Folding the sink into the online Softmax. FlashAttention (Dao et al., 2022; Dao, 2024; Shah et al., 2024) never materializes the $n \times n$ score matrix; it processes key blocks sequentially and keeps, for every query row, a running maximum m and a running normalizer ℓ of the online Softmax (Milakov & Gimelshein, 2018). A virtual sink is simply one more logit that participates in the normalizer but contributes no value. We therefore initialize the running statistics of query i and head h with the sink logit instead of with $(-\infty, 0)$:

$$m^{(0)} = z^{(h)}_{i,\text{sink}}, \qquad \ell^{(0)} = 1, \qquad \mathbf{acc}^{(0)} = \mathbf{0}, \tag{7}$$

after which the standard block updates are unchanged. Because the sink has no value vector, $\mathbf{acc}$ accumulates real values only, and the final output $\mathbf{acc}/\ell$ equals $\sum_j a_{i,j}^{(h)} \mathbf{v}_j$ with $a_{i,j}^{(h)}$ from Equation equation 4. The returned log-sum-exp $\mathrm{LSE}' = m + \log \ell$ includes the sink and is used unchanged in the backward pass, whose recomputation of the probabilities is likewise correct because the same initialization is applied. The sink logit therefore costs one extra scalar read per (query, head) and no additional key column, FLOPs or memory traffic in the inner loop; per-head scalar biases (Sink Attention) are the special case in which the scalar does not depend on i. The implementation supports the variable-length (packed) and context-parallel code paths, in which the sink logit is attached to the query shard and the partial log-sum-exps are merged exactly as without a sink.

Throughput. We measured end-to-end training throughput in the main pre-training setting of Appendix A (20B MoE model, sequence length 4096, FlashAttention-3 kernels for all variants). Softmax attention sustains 292 TFLOPS per GPU and SSA 290 TFLOPS per GPU, a slowdown of 0.7%. The per-(token, head) sink logit therefore has negligible cost: it adds one $n_h \times d$ GEMV per token outside the kernel and one scalar read per (query, head) inside it.

Pseudo-code. Algorithm 1 gives a reference (non-fused) implementation of one SSA layer for clarity; the fused kernel implements the same computation as described above.

Algorithm 1 Scaled Sink Attention for one layer (reference implementation, PyTorch-style).

Require: hidden states $\mathbf{X} \in \mathbb{R}^{n \times d}$; projections $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$; sink projection $\mathbf{W}_{\text{sink}} \in \mathbb{R}^{n_h \times d}$ (zero-initialized, no bias); reference mass ρ ($g = 1/\rho$).

$\mathbf{Q}, \mathbf{K}, \mathbf{V} \leftarrow \textsc{Project}(\mathbf{X})$ ▷ $n \times n_h \times d_k$, RoPE applied as usual
$\mathbf{Z}_{\text{sink}} \leftarrow \mathbf{X}\mathbf{W}_{\text{sink}}^{\top}$ ▷ $n \times n_h$ sink logits, kept in fp32
for each head h **do**
  $\mathbf{S} \leftarrow \mathbf{Q}^{(h)}\mathbf{K}^{(h)\top}/\sqrt{d_k}$ with causal mask ▷ $n \times n$
  $\mathbf{S}' \leftarrow [\,\mathbf{S} \mid \mathbf{Z}_{\text{sink}}^{(h)}\,]$ ▷ append the sink column: $n \times (n+1)$
  $\mathbf{P}' \leftarrow \mathrm{softmax}(\mathbf{S}', \mathrm{dim}=-1)$
  $\mathbf{P} \leftarrow \mathbf{P}'_{:,:n}$ ▷ discard the sink probability; rows now sum to $m_i \le 1$
  $\mathbf{O}^{(h)} \leftarrow g \cdot \mathbf{P}\mathbf{V}^{(h)}$ ▷ fixed gain $g = 1/\rho$ on the surviving attention
end for
return $\mathrm{concat}_h(\mathbf{O}^{(h)})\,\mathbf{W}_o$

D Benchmarks and Evaluation Protocol

All benchmarks are evaluated on base models with greedy decoding. Knowledge benchmarks report multiple-choice accuracy: MMLU, CMMLU and C-Eval have four options, GPQA four options, MMLU-Pro ten options and SuperGPQA up to ten options. Math benchmarks (GSM8K, MATH, CMath) report exact-match accuracy of the final answer. HumanEval, LiveCodeBench and DS-1000 report Pass@1 with execution-based checking; CRUXEval-I and CRUXEval-O report exact match of the predicted input and output, respectively. The reported average is the unweighted mean over the fourteen benchmarks; the second aggregate in Table 1 excludes GPQA and SuperGPQA. The attention analyses of Figures 1, 2 and 6 and of Appendices G–J use a short English probe passage fed to the 524B-token checkpoints with attention weights exported before the output projection.

E Proofs and Derivations

Proof of Proposition 1. For any j, $\hat{a}'_{i,j} = a'_{i,j}/\sum_l a'_{i,l} = c_i a_{i,j}/(c_i \sum_l a_{i,l}) = a_{i,j}/\sum_l a_{i,l} = \hat{a}_{i,j}$, since $c_i > 0$ cancels. □

Remark 1 (Which mechanisms are covered)**.** *Any mechanism of the form $\mathbf{o}_i = c_i \sum_j a_{i,j}\mathbf{v}_j$ with c_i independent of j is covered: Sink Attention (Equation equation 2, $c_i = \sigma(\mathrm{LSE}(\mathbf{z}_i) - b)$), Gated Attention (Equation equation 3, $c_i = \gamma_i$), ASA ($c_i = m_i$, Equation equation 5) and SSA ($c_i = g\,m_i$). Mechanisms that change the shape of the distribution—Softpick (Zuhri et al., 2025),*

stick-breaking (Tan et al., 2025), sigmoid attention (Ramapuram et al., 2025), Differential Transformer (Ye et al., 2025), or any key-dependent bias—are not per-query rescalings and are not covered. The proposition is a statement about the forward pass for fixed logits; it does not say that the covered mechanisms are equivalent after training, and the empirical part of the paper shows that they are not.

The surviving-mass identity. With $E = \sum_{l \le i} \exp(z_{i,l})$ and $s = \exp(z_{i,\text{sink}})$, Equation equation 4 gives $m_i = \sum_j a_{i,j} = E/(s+E) = 1/(1+s/E) = 1/(1+\exp(z_{i,\text{sink}} - \mathrm{LSE}(\mathbf{z}_i))) = \sigma(\mathrm{LSE}(\mathbf{z}_i) - z_{i,\text{sink}})$, which is Equation equation 5; the sink probability is $p_{\text{sink}} = 1 - m_i = \sigma(z_{i,\text{sink}} - \mathrm{LSE}(\mathbf{z}_i))$. Thus the virtual sink implements a logistic comparison between the attention evidence $\mathrm{LSE}(\mathbf{z}_i)$—a smooth maximum of the logits, large when at least one key matches the query well—and the learned, token-specific threshold $z_{i,\text{sink}}$.

Gradient of the sink logit. Let $\mathcal{L}$ be the loss and $\mathbf{u}_i = \sum_j \hat{a}_{i,j}\mathbf{v}_j$ the normalized attention output. Since $\mathbf{o}_i = g\, m_i\, \mathbf{u}_i$ and $\partial m_i / \partial z_{i,\text{sink}} = -m_i(1-m_i)$,

$$\frac{\partial \mathcal{L}}{\partial z_{i,\text{sink}}} = -\, g\, m_i(1-m_i) \left\langle \frac{\partial \mathcal{L}}{\partial \mathbf{o}_i},\, \mathbf{u}_i \right\rangle, \qquad \nabla \mathbf{w}_{\text{sink}}^{(h)} = \sum_i \frac{\partial \mathcal{L}}{\partial z_{i,\text{sink}}^{(h)}}\, \mathbf{x}_i. \tag{8}$$

The sink logit of token i is therefore pushed up (more draining) exactly when the normalized attention output $\mathbf{u}_i$ is anti-aligned with the direction the loss wants the head to write, and pushed down when it is aligned—the gradient is a per-token "is this head's output useful here?" signal, modulated by $m_i(1-m_i)$, which vanishes when the decision is already saturated in either direction. The logits of the real tokens receive the complementary signal through the usual Softmax Jacobian. Because $\mathbf{W}_{\text{sink}}$ is zero-initialized, $z_{i,\text{sink}} = 0$ and $m_i = \sigma(\mathrm{LSE}(\mathbf{z}_i))$ at initialization: all heads start as a mild, uniform relaxation ($m_i \approx 1$ once LSE grows beyond a few units), and draining is switched on only where the gradient above accumulates consistently. The gain g multiplies this gradient but, as discussed in Appendix H, Muon's updates are invariant to such a uniform rescaling for matrix parameters. Note also that $\mathbf{W}_{\text{sink}}$ is itself a matrix updated by Muon with a fixed update RMS, so the sink logits move at an optimizer-determined speed once the gradient direction is consistent; the gradient above determines where the sink is switched on, not how fast.

Further remarks on the gain. The gain enters the forward pass only as a multiplier of the attention output, so g can be absorbed into $\mathbf{W}_o$: for any SSA model there is an ASA model with $\mathbf{W}'_o = g\,\mathbf{W}_o$ computing the same function. The benefit of g must therefore be a training-dynamics effect. Under Muon the update of $\mathbf{W}_o$ has a fixed RMS regardless of the gradient scale, and weight decay pulls $\|\mathbf{W}_o\|$ toward zero, so realizing a five-fold larger output scale through $\mathbf{W}_o$ would take many steps and be continuously counteracted, whereas a fixed gain provides it from the first step.

Expected mass conservation and the choice of ρ. This paragraph gives the sink-usage argument behind $\rho = 0.2$ (Section 4.2). Draining a fraction p_{sink} of the mass shrinks the attention output by $(1 - p_{\text{sink}})$; the gain that keeps the expected output magnitude of a head equal to that of vanilla Softmax is

$$g^\star = \frac{1}{\mathbb{E}_i\big[1 - p_{i,\text{sink}}\big]} = \frac{1}{1 - \bar{p}_{\text{sink}}}. \tag{9}$$

Appendix G reports the per-head mean sink mass $\bar{p}_{\text{sink}}$ for all $32 \times 16 = 512$ heads of the trained ASA model. Averaged over heads, $\bar{p}_{\text{sink}} = 0.644$, so conserving the mean head predicts $g^\star \approx 2.8$; but the distribution is broad: one third of the heads drain at least 80% of their mass (seven of the first eight layers have a median $\bar{p}_{\text{sink}}$ above 0.8), so these heads need $g^\star \ge 5$. A single gain must serve all heads. Setting $\rho = 0.2$ ($g = 5$) conserves these heavily draining heads in expectation and over-compensates lightly draining heads by at most a factor of five (implied per-head output scale $g(1 - \bar{p}_{\text{sink}})$: median 1.6, one third of heads below one, none above five; Figure 7); this over-compensation is the source of the positive $\Delta\mathcal{R}_i$ in Figure 2. The head that switches off on most tokens but fires on a few has its firing outputs amplified by g (Appendix H argues that this is what lets it fire with smaller logits), and once the gain is present the model drains more (head-averaged surviving mass $0.356 \to 0.303$, Appendix G), so the effective output scale of SSA is $g(1 - \bar{p}_{\text{sink}}) \approx 1.5$ times that of Softmax—a moderate boost rather than an explosion. The distribution brackets the useful range: $\rho = 0.1$ ($g = 10$), whose implied scale exceeds five for a quarter of the heads, leads early but falls behind $g = 5$ from 441B tokens and degrades at the end (Section 5.2).

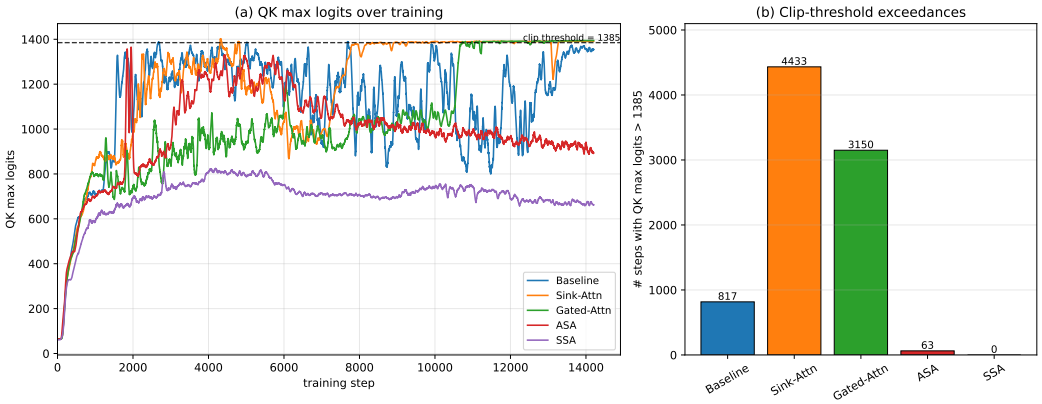


Figure 5: Attention-logit stability. **(a)** Maximum QK logit (smoothed) versus training tokens; the dashed line is the QK-clip threshold $\tau = 1385$. **(b)** Share of training iterations whose maximum logit exceeds τ (counts in parentheses).

Implied output scale and its spread. For a head with per-query surviving mass m_i, the ratio of the SSA output norm to the Softmax output norm with the same normalized distribution is $g\, m_i$. Averaged over queries this is $g(1 - \bar{p}_{\text{sink}})$, the quantity plotted in Figure 7(b); mass conservation in expectation (Equation equation 9) is the condition $g(1 - \bar{p}_{\text{sink}}) = 1$. Over the 512 heads of the trained ASA model, $g(1 - \bar{p}_{\text{sink}})$ has median 0.88 for $g = g^{\star}_{\text{mean}} = 2.8$ (with 56% of heads below one, i.e. attenuated), median 1.57 for $g = 5$ (33% below one, none above five), and median 3.13 for $g = 10$ (27% of heads above five). The reference-mass parameterization $g = 1/\rho$ makes this trade-off explicit: ρ is the surviving mass at which a head is exactly conserved, and heads that keep less than ρ are attenuated while heads that keep more are amplified.

F Training Stability

This appendix expands on the stability results summarized in Section 5.1. We monitor the *maximum QK logit*, $\max_{i,j} \mathbf{q}_i^{\top} \mathbf{k}_j$ over all heads and causal pairs (before $1/\sqrt{d_k}$ scaling), throughout training. Large logits saturate the Softmax, induce massive-activation outliers, and force external clipping (Wortsman et al., 2024; Kimi Team, 2025); the QK-clip safeguard in our runs fires whenever the maximum exceeds $\tau = 1385$, and the less often it fires, the more self-regularizing the attention is.

In Figure 5(a) the Baseline, Sink Attention and Gated Attention grow quickly and spend long stretches pressed against τ (final-quarter medians 1165–1392); ASA stays much lower (945) and SSA lower still, peaking at 933 after 94B tokens and settling around 690, roughly half the level of Sink and Gated Attention and 40% below the Baseline (Appendix M, Table 13). Figure 5(b) counts clip events: the Baseline exceeds τ in 817 iterations (5.8% of training), and the prior mitigations are markedly worse, Sink Attention in 4,433 (31.2%) and Gated Attention in 3,150 (22.2%). This is consistent with the evidence-aware-gate view of Section 4.1: with a fixed threshold or a decoupled gate, the model can only separate informative from uninformative queries by inflating its logits. The input-dependent sink removes this pressure—ASA exceeds τ only 63 times (0.4%)—and the gain removes the residual events: **SSA never triggers the clip**. Most of the stability gain is thus due to the sink and the remainder to the gain.

Where the outliers come from. Appendix H shows that in the Baseline, Sink Attention and Gated Attention the heads with the largest logits are those whose query and key up-projection weights have the smallest L2 norms (Pearson r between -0.49 and -0.75 in the layers examined), and Appendix I that positions with large logits have low attention entropy in every method. Extreme logits are the price a weak head pays for sharp attention; our hypothesis, developed in Appendix H, is that by supplying output magnitude through the gain, SSA lowers the sharpness a head needs.

G How the Virtual Sink Is Used: Heads, Depth, and the Effect of the Gain

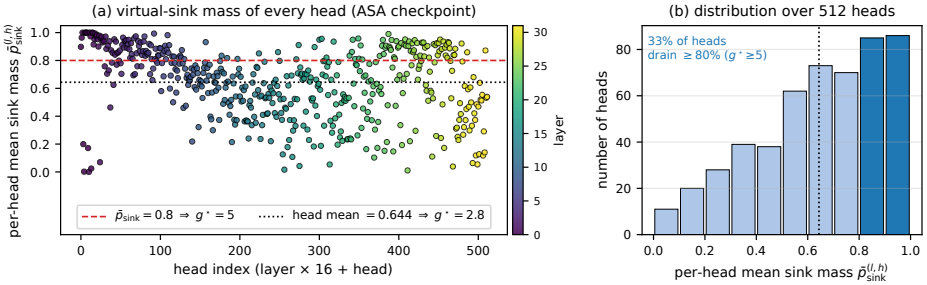


Figure 6: Learned virtual-sink mass of the trained ASA checkpoint (g=1, so no gain is involved). **(a)** Per-head mean sink mass $\bar{p}_{\text{sink}}^{(l,h)}$ for all $32 \times 16 = 512$ heads, colored by layer. **(b)** Its distribution: head mean 0.644 (median 0.687); 73% of heads drain more than half of their mass and one third drain at least 80%.

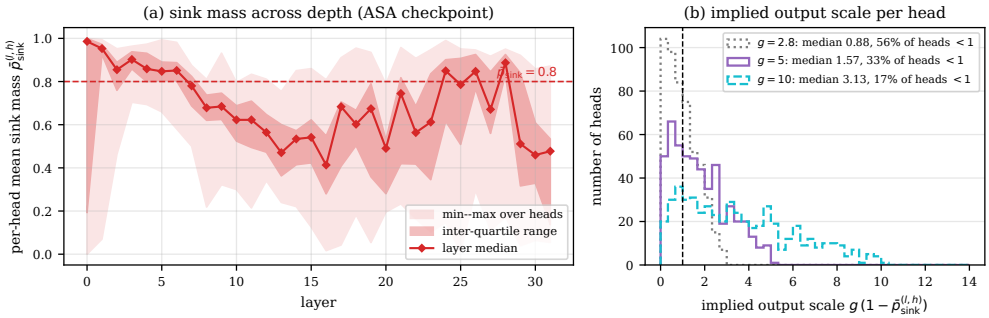


Figure 7: **(a)** Per-head mean sink mass $\bar{p}_{\text{sink}}^{(l,h)}$ of the ASA checkpoint across depth: layer median (line), inter-quartile range (dark band) and min–max range over the 16 heads of each layer (light band). **(b)** Implied per-head output scale $g(1 - \bar{p}_{\text{sink}}^{(l,h)})$ relative to Softmax for the head-mean-conserving $g^\star = 2.8$, the default $g = 5$ and $g = 10$; the dashed line marks exact conservation.

Heads. Figure 6 shows that the learned sink is strongly head-dependent: the per-head mean sink mass $\bar{p}_{\text{sink}}^{(l,h)}$ ranges from essentially zero to essentially one, with a head mean of 0.644 and a median of 0.687. There is no bimodality—heads do not split into "sink heads" and "content heads"—but instead form a broad continuum, so any fixed rule for which heads may drain would fit poorly; the learned, input-dependent threshold lets every head settle at its own operating point.

Depth. Figure 7(a) resolves the same data by layer. The early layers drain the most: the layer medians of the first eight layers are 0.99, 0.95, 0.85, 0.90, 0.86, 0.85, 0.85 and 0.78, and the ten layers with a median above 0.8 are layers 0–6, 24, 26 and 28. Draining decreases through the middle of the network (medians between 0.4 and 0.6 around layers 13–17), rises again in layers 22–28, and falls to about 0.5 in the last three layers. The within-layer spread is large everywhere except at the very first layer, where almost every head drains almost everything. This depth profile is consistent with the common observation that early layers of LLMs contain many heads that act as near no-ops on most tokens (Clark et al., 2019; Sun et al., 2024): the virtual sink gives such heads an explicit, input-dependent way to switch off without parking their mass on a real token.

Effect of the gain on sink usage. Figure 8 compares the per-head mean surviving mass $\bar{m} = 1 - \bar{p}_{\text{sink}}$ (before the gain) of the trained ASA and SSA checkpoints. The SSA distribution is shifted left (head mean 0.303 versus 0.356): with the gain acting as a safety net, the model drains more,

so the fixed gain changes how the sink is used rather than merely rescaling the output. Figure 7(b) shows the per-head implied output scale $g(1-\bar{p}_{\text{sink}})$ for three values of g (numbers in Appendix E).

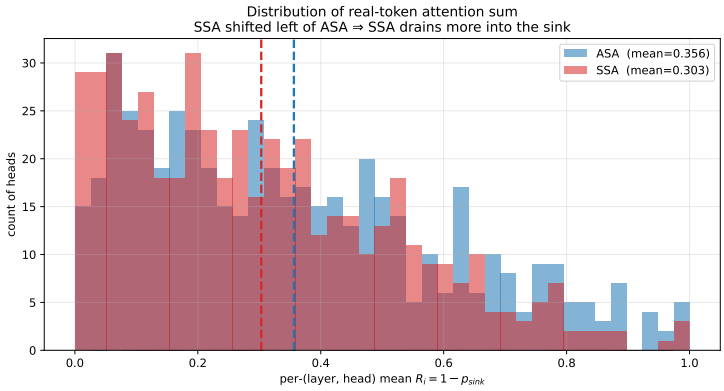
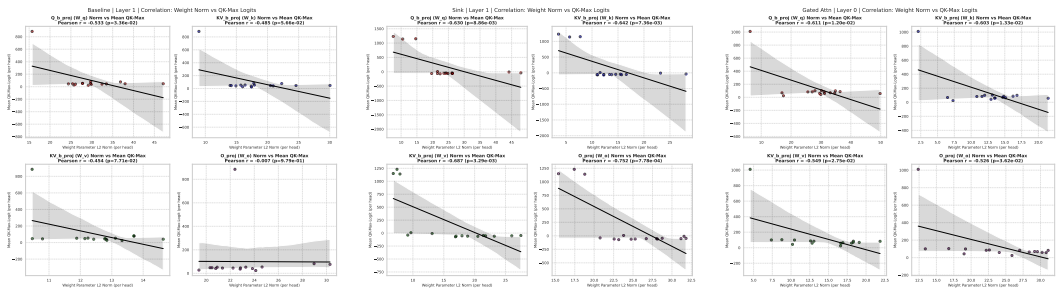


Figure 8: Distribution over the 512 heads of the per-head mean surviving real-token mass $\bar{m} = 1 - \bar{p}_{\text{sink}}$ for ASA (g=1) and SSA (g=5, gain divided out). Dashed lines mark the means.

H Why SSA Suppresses QK Max Logits: A Weight-Norm Perspective

This section details where the abnormally large QK logits of the Baseline and the prior mitigations come from. We first connect the head-wise maximum QK logit to the L2 norm of the per-head query and key projection weights (Figure 9), then state our hypothesis for why the gain of SSA lowers these logits, and finally remark on the role of the gradient magnitude under Muon.

Empirical observation: large QK logits coincide with small $\mathbf{W}_q, \mathbf{W}_k$ norms. For each head of one layer per model we measure (i) its mean maximum QK logit over a probe text and (ii) the per-head L2 norm of the up-projection weights $\mathbf{W}_q$ (the query up-projection, $\mathbf{W}_{qb}$ below) and $\mathbf{W}_k$ (the key part of the key–value up-projection, $\mathbf{W}_{kb_nope}$ below). Figure 9 plots these quantities for the Baseline, Sink Attention and Gated Attention. In all three, the heads with anomalously large logits are those with anomalously small $\mathbf{W}_q$ and $\mathbf{W}_k$ norms: the correlation is consistently negative (Pearson $r \approx -0.53$ for $\mathbf{W}_q$ and -0.49 for $\mathbf{W}_k$ in the Baseline layer, and between -0.53 and -0.75 in the Sink Attention and Gated Attention layers), whereas in the Baseline the output projection $\mathbf{W}_o$ is uncorrelated ($r \approx -0.01$).



(a) Baseline (layer 1). (b) Sink Attention (layer 1). (c) Gated Attention (layer 0).

Figure 9: Per-head correlation between weight L2 norm and mean maximum QK logit in one layer per model. The head with an anomalously large logit has anomalously small $\mathbf{W}_q$ and $\mathbf{W}_k$ norms.

Interpretation: sharpness and the norms of the query/key weights. When a head has very small $\mathbf{W}_q$ and $\mathbf{W}_k$, it can only realize a sharp preference by driving $\mathbf{q}_i^\top \mathbf{k}_j$ to very large values,

producing the outliers that trigger the QK-clip; Appendix I confirms that positions with large logits are positions with sharp, low-entropy attention. A head must be sharp whenever it needs an output of a given magnitude, because averaging many values cancels. Sink Attention and Gated Attention rescale only the output mass and leave the query/key geometry untouched—and, as Appendix F shows, they even encourage larger raw logits. SSA changes this trade-off: with a gain g, the same output magnitude is reached with g times less mass on the selected tokens, so less sharpness—and hence smaller logits—suffices. In the trained model this headroom is spent partly on draining more (head-averaged surviving mass $0.356 \to 0.303$, Appendix G) and partly on a net output gain of about $1.5\times$ over Softmax (Appendix E). First, a fixed gain is a reparameterization—Softmax attention could scale $\mathbf{W}_o$ by g—so the benefit must come from the optimizer's implicit bias: under Muon, whose updates have bounded RMS, and weight decay, growing $\|\mathbf{W}_o\|$ five-fold is slow and costly, whereas the gain supplies the scale for free (Appendix E). Second, it is not a gradient-magnitude argument (see the remark below). The stability data are consistent with it: the sink alone reduces QK-clip events from 817 to 63, and the gain removes the remaining 63 while lowering the maximum logit further (final-quarter median $945 \to 692$; Appendix M).

Remark on gradient magnitude. Because the output of SSA is $g \sum_j a_{i,j}\mathbf{v}_j$, the gradient reaching the scores, and with it $\nabla\mathbf{W}_q$ and $\nabla\mathbf{W}_k$ (the query up-projection and the content part of the key–value up-projection in MLA), carries the factor g, whereas in Sink Attention and Gated Attention it carries the factor $c_i < 1$. Under an optimizer whose step scales with the gradient (e.g. SGD) this would directly enlarge the updates of exactly the weights identified by the correlation above; under Muon, which we use, the update is invariant to a uniform rescaling of the gradient, so this factor does not by itself change the weight trajectory, and the explanation above rests on the sharpness argument combined with the optimizer's implicit bias against growing $\mathbf{W}_o$, not on gradient magnitude.

I Maximum QK Logit and Attention Sharpness

Figure 10 plots, for every layer, the Pearson correlation between the per-position maximum QK logit and the entropy of the attention distribution at that position (over a probe text, skipping the first four positions). The correlation is negative in almost every layer of every model: positions with large logits are positions with sharp, low-entropy attention. This supports the interpretation that extreme logits are the mechanism by which weak heads achieve sharp attention, rather than noise.

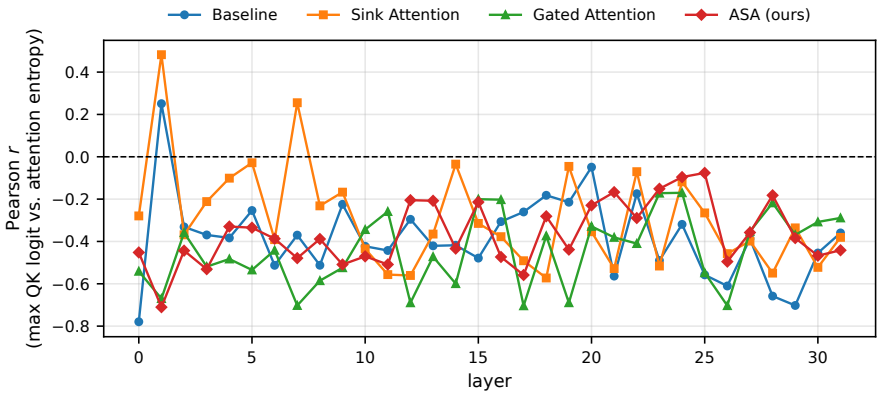


Figure 10: Per-layer Pearson correlation between the maximum QK logit and the attention entropy of a query position. Negative values indicate that large logits accompany sharp attention.

J Additional Attention Visualizations

Figure 11 shows the absolute remaining attention $\mathcal{R}_i = \sum_{j\geq 1} a_{i,j}$ whose difference to the Baseline is plotted in Figure 2. The Baseline leaves about half of its mass for tokens $j \geq 1$; Sink Attention and Gated Attention lie below it in almost every layer and at almost every query position, whereas

SSA (including the gain) lies above it and, in the layers that drain little, exceeds one—the "deliberate sharpening" of Section 4.2. Across query positions (panel b), all methods start low at the first few positions, where there is little context to attend to, and the gap between SSA and the prior methods is roughly constant beyond position 10, i.e. the gain does not favor early or late positions.

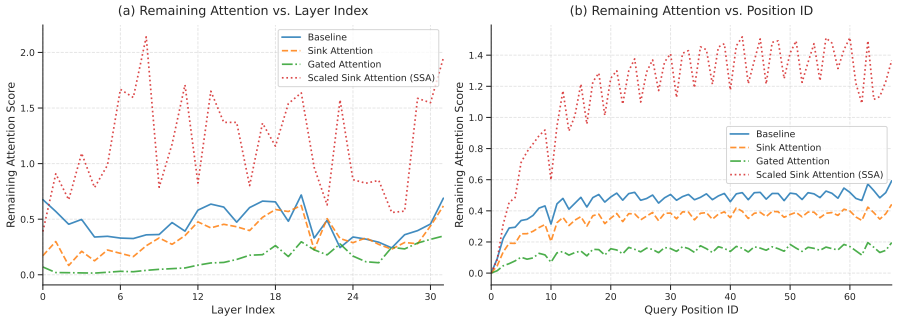


Figure 11: Absolute remaining attention $\mathcal{R}_i$ on real context tokens ($j \geq 1$) for the four methods, (left) averaged per layer and (right) averaged per query position. Values above one for SSA are due to the gain g.

K Ablation Details

Table 4: Ablations after 524B tokens. **Top:** design of the virtual sink without the gain (ASA). **Bottom:** the gain applied on top of the default virtual sink; input-dependent gates replace the fixed gain by a head-wise $g_i = f(\mathbf{w}_g^\top \mathbf{x}_i)$; the best of the five gates is shown, all five are in Table 5. **Avg** (shaded) is over all fourteen benchmarks; best per column within the gain block in **bold**. † g=10 scored 0.50/1.38 on CRUXEval-I/-O at this checkpoint (see Section 5.2); over the other twelve benchmarks it averages 34.87 versus 37.05 for g=5.

Variant	C-Eval	CMath	GSM8K	HumanEval	MATH	MMLU	**Avg**
Baseline (Softmax)	49.01	54.46	48.98	44.51	23.70	56.98	33.36
ASA: virtual-sink design (no gain, g=1)							
random init	50.11	56.65	51.18	42.68	25.40	56.84	34.02
zero init + learnable bias	48.67	51.91	51.25	46.95	16.68	59.00	33.27
zero init, no bias (default)	50.02	53.64	53.15	45.12	20.64	57.14	34.98
SSA: gain on top of the default virtual sink							
learnable per-head g (init 5)	47.19	59.65	54.74	46.95	**27.54**	57.11	35.73
input-dependent $g_i = \sigma(\cdot)$ (best of five)	48.49	56.01	50.49	41.46	24.88	**57.62**	34.61
fixed $g = 10$ †	47.17	56.01	53.75	42.68	26.96	56.56	30.03
fixed $g = 5$ (default)	**49.15**	**61.11**	**55.72**	**49.39**	26.88	57.48	**36.10**

Table 5: Input-dependent gain variants after 524B tokens (all on top of the default virtual sink; head-wise gate $g_i = f(\mathbf{w}_g^\top \mathbf{x}_i)$). **Avg** is over all fourteen benchmarks; best per column in **bold**. The fixed $g = 5$ row is repeated for reference.

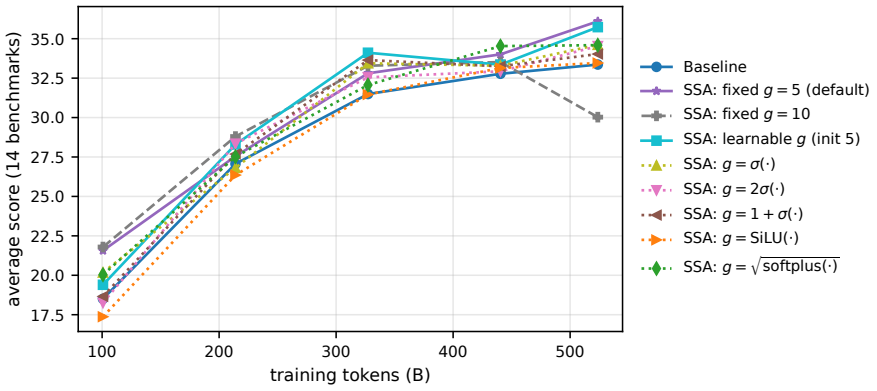
Variant	C-Eval	CMath	GSM8K	HumanEval	MATH	MMLU	**Avg**
SSA: $g_i = \sigma(\cdot)$	48.49	56.01	50.49	41.46	24.88	57.62	34.61
SSA: $g_i = 2\sigma(\cdot)$	48.94	50.00	52.08	40.85	**27.20**	59.40	34.53
SSA: $g_i = 1 + \sigma(\cdot)$	48.05	55.56	52.77	43.90	25.60	57.59	34.03
SSA: $g_i = \mathrm{SiLU}(\cdot)$	47.60	50.73	48.22	46.95	23.34	54.85	33.46
SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	**49.64**	56.65	52.16	36.59	25.30	**59.50**	34.59
SSA: fixed $g = 5$ (default)	49.15	**61.11**	**55.72**	**49.39**	26.88	57.48	**36.10**

Table 4 is the ablation table discussed in Section 5.2, Table 5 lists all input-dependent gain variants, Figure 12 shows the fourteen-benchmark average of every ablation variant at the five evaluated

checkpoints (thirteen benchmarks at 327B, see Appendix P), and Figure 17(b) the per-benchmark picture of the virtual-sink designs. The default zero-initialized, bias-free virtual sink pulls ahead of the other sink designs from 441B tokens and keeps climbing, whereas the bias-augmented variant flattens toward the Baseline. Among the gain choices, the fixed $g = 5$ finishes highest; $g = 10$ and the learnable gain are ahead of it between 214B and 327B tokens but fall behind from 441B tokens.



(a) Design of the virtual sink (no gain).



(b) Choice of the gain g.

Figure 12: Average score over the fourteen benchmarks versus training tokens for the ablation variants of Table 4.

Every ablation variant at every checkpoint. Tables 6–9 report all fourteen benchmarks for all twelve ablation variants at the five checkpoints. Three regularities are worth noting. First, the ordering among the sink designs is not stable early on: random initialization is ahead of the default at 214B and 327B tokens (at 101B its average is over thirteen benchmarks and not comparable) and falls behind from 441B onward, so short ablation runs would have picked the wrong design. Second, the input-dependent gates are never far from ASA at any checkpoint, i.e. they add little on top of the virtual sink, consistent with the interpretation of Section 5.2 that an adaptive gate collapses toward the uniform-rescaling regime. Third, the learnable per-head gain and $g = 10$ share the same trajectory shape—ahead of the fixed $g = 5$ from 214B to 327B tokens ($g = 10$ already at 101B) and behind it from 441B—which suggests that an over-large effective gain helps early optimization but is not what the converged model wants; the CMMLU score of the learnable gain at 441B (34.00, against 44.5 and 49.7 at the adjacent checkpoints) is an isolated outlier that recovers at the next checkpoint.

Table 6: Knowledge and reasoning benchmarks of all ablation variants at 101, 214 and 327B tokens. Best per column within each checkpoint in **bold**. ‡At 327B the ASA checkpoint has no SuperGPQA score; the random-init variant also lacks SuperGPQA at 101B (shown as "–"; its average at that checkpoint is over thirteen benchmarks).

Tokens	Method	MMLU	MMLU-Pro	CMMLU	C-Eval	GPQA	SuperGPQA
101B	Baseline (Softmax)	38.28	13.31	31.08	25.71	**21.21**	10.57
	ASA: random init	37.57	14.44	28.17	27.45	18.97	–
	ASA: zero init + learnable bias	34.94	13.29	27.51	28.08	14.29	10.09
	ASA: zero init, no bias (default)	34.90	15.32	30.70	28.30	18.08	10.19
	SSA: learnable per-head g (init 5)	33.05	14.01	27.52	30.74	14.29	9.56
	SSA: $g_i = \sigma(\cdot)$	35.94	15.35	26.03	28.32	20.54	**10.85**
	SSA: $g_i = 2\sigma(\cdot)$	38.12	15.56	27.62	29.22	17.41	3.56
	SSA: $g_i = 1 + \sigma(\cdot)$	34.12	14.40	27.11	25.77	20.31	9.51
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	33.78	11.98	25.28	27.15	15.18	9.29
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	**40.51**	14.93	29.15	29.61	16.96	9.63
	SSA: fixed $g = 10$	38.45	**17.51**	30.59	**33.08**	20.54	10.71
	SSA: fixed $g = 5$ (default)	37.05	16.31	**31.86**	30.66	20.09	10.67
214B	Baseline (Softmax)	48.89	19.60	40.58	39.67	23.21	12.10
	ASA: random init	48.14	20.19	37.45	39.89	19.64	11.76
	ASA: zero init + learnable bias	49.09	20.43	38.13	41.70	20.54	12.97
	ASA: zero init, no bias (default)	48.84	20.77	39.90	39.66	17.63	12.73
	SSA: learnable per-head g (init 5)	48.81	20.62	41.73	**43.42**	19.87	12.56
	SSA: $g_i = \sigma(\cdot)$	48.53	20.93	41.49	43.15	17.63	12.54
	SSA: $g_i = 2\sigma(\cdot)$	**50.56**	22.88	41.08	41.81	21.21	8.74
	SSA: $g_i = 1 + \sigma(\cdot)$	47.97	19.46	38.81	40.52	20.54	11.61
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	46.98	21.43	39.49	41.03	17.19	12.02
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	49.78	22.75	40.75	40.28	20.76	**13.02**
	SSA: fixed $g = 10$	48.34	**24.31**	**41.90**	40.59	**23.88**	12.70
	SSA: fixed $g = 5$ (default)	49.37	21.70	40.29	40.84	17.63	12.78
327B‡	Baseline (Softmax)	53.22	23.58	42.76	43.49	22.54	13.06
	ASA: random init	53.19	24.75	44.04	46.10	20.09	**14.29**
	ASA: zero init + learnable bias	53.69	23.46	42.11	46.91	20.54	13.46
	ASA: zero init, no bias (default)	51.81	23.56	44.38	**47.36**	**24.11**	–
	SSA: learnable per-head g (init 5)	51.88	24.33	44.51	44.98	23.88	13.67
	SSA: $g_i = \sigma(\cdot)$	53.98	**25.85**	45.61	45.65	22.54	14.21
	SSA: $g_i = 2\sigma(\cdot)$	**54.82**	24.26	45.56	46.05	21.43	8.54
	SSA: $g_i = 1 + \sigma(\cdot)$	53.79	23.56	43.03	44.68	23.88	13.08
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	50.66	23.50	42.70	44.23	23.88	14.08
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	53.72	25.33	**45.73**	45.34	21.21	13.22
	SSA: fixed $g = 10$	52.48	24.34	42.25	45.79	20.54	12.65
	SSA: fixed $g = 5$ (default)	52.28	23.75	44.24	44.40	21.21	12.97

Table 7: Knowledge and reasoning benchmarks of all ablation variants at 441 and 524B tokens. Best per column within each checkpoint in **bold**.

Tokens	Method	MMLU	MMLU-Pro	CMMLU	C-Eval	GPQA	SuperGPQA
441B	Baseline (Softmax)	55.36	25.03	47.20	47.33	22.10	13.93
	ASA: random init	56.59	26.83	45.72	47.65	24.55	14.11
	ASA: zero init + learnable bias	55.88	**27.57**	48.38	**48.85**	20.98	15.26
	ASA: zero init, no bias (default)	55.84	26.95	46.23	46.20	22.10	14.97
	SSA: learnable per-head g (init 5)	55.52	26.82	34.00	47.10	18.53	14.02
	SSA: $g_i = \sigma(\cdot)$	55.24	26.59	48.70	45.15	18.97	**15.50**
	SSA: $g_i = 2\sigma(\cdot)$	55.80	26.34	46.83	46.94	16.52	13.81
	SSA: $g_i = 1 + \sigma(\cdot)$	55.81	24.04	45.31	47.98	22.10	13.08
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	53.70	25.72	46.38	46.13	20.31	13.34
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	**57.27**	27.56	48.60	46.87	**28.12**	13.63
	SSA: fixed $g = 10$	54.86	26.70	45.49	45.90	23.66	13.41
	SSA: fixed $g = 5$ (default)	55.15	26.09	**49.17**	47.19	17.19	15.13
524B	Baseline (Softmax)	56.98	26.04	48.55	49.01	21.88	14.17
	ASA: random init	56.84	28.30	49.50	**50.11**	23.88	15.09
	ASA: zero init + learnable bias	59.00	28.40	47.61	48.67	23.21	**16.17**
	ASA: zero init, no bias (default)	57.14	**29.27**	49.85	50.02	**25.89**	15.16
	SSA: learnable per-head g (init 5)	57.11	28.07	49.67	47.19	20.98	14.02
	SSA: $g_i = \sigma(\cdot)$	57.62	29.20	50.21	48.49	20.31	15.79
	SSA: $g_i = 2\sigma(\cdot)$	59.40	28.82	50.72	48.94	20.76	15.16
	SSA: $g_i = 1 + \sigma(\cdot)$	57.59	25.69	47.59	48.05	22.54	13.26
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	54.85	27.62	45.98	47.60	21.21	14.67
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	**59.50**	28.44	**51.14**	49.64	21.43	14.21
	SSA: fixed $g = 10$	56.56	27.35	48.66	47.17	23.88	12.67
	SSA: fixed $g = 5$ (default)	57.48	27.91	50.92	49.15	22.99	14.92

Table 8: Math and code benchmarks and the fourteen-benchmark average (shaded; thirteen at 327B‡) of all ablation variants at 101, 214 and 327B tokens. CRUX-I/-O denote CRUXEval-Input/-Output, LCB denotes LiveCodeBench. Best per column within each checkpoint in **bold**.

Tokens	Method	GSM8K	MATH	CMath	HumanEval	CRUX-I	CRUX-O	LCB	DS-1000	**Avg**
101B	Baseline (Softmax)	25.17	12.62	20.95	21.95	11.25	18.50	2.56	5.90	18.50
	ASA: random init	24.26	13.38	20.58	27.44	13.12	16.88	3.03	6.30	19.35
	ASA: zero init + learnable bias	24.18	11.98	24.59	28.05	9.62	17.00	3.32	4.60	17.97
	ASA: zero init, no bias (default)	26.91	14.24	20.49	26.83	13.25	17.12	3.41	8.00	19.12
	SSA: learnable per-head g (init 5)	25.25	13.50	23.13	28.05	21.62	18.12	3.41	9.30	19.40
	SSA: $g_i = \sigma(\cdot)$	25.32	12.70	25.59	29.27	**25.00**	16.38	3.13	8.00	20.17
	SSA: $g_i = 2\sigma(\cdot)$	19.71	12.12	25.32	27.44	16.00	13.25	3.41	6.80	18.25
	SSA: $g_i = 1 + \sigma(\cdot)$	24.26	12.78	23.86	**31.10**	10.00	15.38	2.65	10.00	18.66
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	21.00	11.10	22.04	26.83	13.63	17.88	2.46	5.60	17.37
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	23.35	**15.04**	26.05	25.00	20.00	18.12	3.51	8.60	20.03
	SSA: fixed $g = 10$	24.03	13.16	28.60	29.27	24.00	**21.25**	**4.08**	9.80	**21.79**
	SSA: fixed $g = 5$ (default)	**27.67**	14.88	**31.06**	28.66	19.50	17.75	3.98	**11.80**	21.57
214B	Baseline (Softmax)	38.89	17.56	40.44	33.54	22.38	21.88	5.88	**14.30**	27.07
	ASA: random init	36.16	18.76	36.70	36.59	32.75	22.25	6.54	9.40	26.87
	ASA: zero init + learnable bias	38.36	19.04	38.52	35.98	29.38	21.38	6.73	9.70	27.28
	ASA: zero init, no bias (default)	36.47	13.12	36.70	34.76	24.88	20.75	5.59	9.50	25.81
	SSA: learnable per-head g (init 5)	41.02	18.70	43.81	35.37	28.12	23.00	5.88	13.20	28.29
	SSA: $g_i = \sigma(\cdot)$	38.13	17.86	35.79	31.71	31.37	20.62	5.40	10.20	26.81
	SSA: $g_i = 2\sigma(\cdot)$	38.82	19.18	39.98	37.20	**35.38**	21.12	6.35	12.50	28.34
	SSA: $g_i = 1 + \sigma(\cdot)$	40.64	19.68	42.26	35.98	28.12	23.75	6.07	12.20	27.69
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	35.41	15.44	33.52	**38.41**	35.12	21.38	4.83	6.70	26.35
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	40.33	18.50	39.07	34.76	28.38	18.50	**7.49**	10.90	27.52
	SSA: fixed $g = 10$	**42.30**	**20.92**	43.81	36.59	31.50	**24.00**	7.11	5.20	**28.80**
	SSA: fixed $g = 5$ (default)	41.85	19.86	**45.90**	32.32	23.62	21.75	5.59	12.10	27.54
327B‡	Baseline (Softmax)	42.53	19.34	49.09	39.63	25.87	24.38	6.54	16.40	31.49
	ASA: random init	42.84	22.76	46.81	43.29	37.12	23.75	6.64	14.20	32.74
	ASA: zero init + learnable bias	43.97	20.94	44.17	**46.34**	20.62	22.62	6.92	10.60	30.99
	ASA: zero init, no bias (default)	46.17	17.94	43.90	40.85	**37.38**	26.62	6.73	13.70	32.65
	SSA: learnable per-head g (init 5)	**51.48**	**23.78**	**52.28**	42.68	35.38	26.75	7.39	14.10	**34.11**
	SSA: $g_i = \sigma(\cdot)$	47.76	20.10	49.82	38.41	36.00	26.62	7.11	15.30	33.44
	SSA: $g_i = 2\sigma(\cdot)$	47.46	21.44	45.63	37.20	32.62	22.38	8.06	16.20	32.55
	SSA: $g_i = 1 + \sigma(\cdot)$	47.38	22.66	48.27	44.51	33.00	26.75	**8.15**	**17.70**	33.64
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	41.24	20.30	44.99	39.02	35.00	25.12	5.78	12.60	31.46
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	45.26	21.34	50.18	33.54	27.62	24.88	7.58	15.10	32.06
	SSA: fixed $g = 10$	45.79	22.64	47.72	44.51	33.75	**29.25**	8.06	15.60	33.29
	SSA: fixed $g = 5$ (default)	49.05	22.96	51.00	42.68	26.12	25.87	7.01	15.90	32.81

Table 9: Math and code benchmarks and the fourteen-benchmark average (shaded) of all ablation variants at 441 and 524B tokens. Best per column within each checkpoint in **bold**.

Tokens	Method	GSM8K	MATH	CMath	HumanEval	CRUX-I	CRUX-O	LCB	DS-1000	**Avg**
441B	Baseline (Softmax)	49.36	24.22	54.46	44.51	26.88	23.75	8.53	16.20	32.78
	ASA: random init	48.98	25.34	51.28	43.29	34.62	25.12	8.06	12.50	33.19
	ASA: zero init + learnable bias	**52.77**	24.30	51.00	45.12	27.75	28.25	9.29	14.20	33.54
	ASA: zero init, no bias (default)	51.18	25.78	52.00	43.29	38.00	27.38	7.01	12.80	33.55
	SSA: learnable per-head g (init 5)	51.55	24.54	55.65	46.34	37.75	30.50	8.82	15.80	33.35
	SSA: $g_i = \sigma(\cdot)$	49.51	21.26	53.37	42.07	37.00	29.00	8.72	14.90	33.28
	SSA: $g_i = 2\sigma(\cdot)$	49.73	24.02	50.46	44.51	32.00	28.12	**9.57**	16.40	32.93
	SSA: $g_i = 1 + \sigma(\cdot)$	49.05	23.08	53.55	41.46	39.88	27.62	8.72	13.80	33.25
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	48.29	23.88	49.64	45.12	**42.88**	29.62	7.58	11.40	33.14
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	49.05	23.58	56.28	40.85	35.38	29.50	8.25	**18.50**	**34.53**
	SSA: fixed $g = 10$	49.66	**26.12**	53.64	**46.95**	35.88	**32.12**	9.48	5.40	33.52
	SSA: fixed $g = 5$ (default)	51.93	24.98	**59.29**	45.73	31.62	26.62	9.10	16.90	34.01
524B	Baseline (Softmax)	48.98	23.70	54.46	44.51	25.12	28.12	8.72	16.80	33.36
	ASA: random init	51.18	25.40	56.65	42.68	27.51	26.00	8.63	14.50	34.02
	ASA: zero init + learnable bias	51.25	16.68	51.91	46.95	26.75	24.50	9.19	15.50	33.27
	ASA: zero init, no bias (default)	53.15	20.64	53.64	45.12	**38.94**	26.14	8.72	16.00	34.98
	SSA: learnable per-head g (init 5)	54.74	**27.54**	59.65	46.95	35.88	**29.88**	9.19	**19.40**	35.73
	SSA: $g_i = \sigma(\cdot)$	50.49	24.88	56.01	41.46	36.38	29.12	9.19	15.40	34.61
	SSA: $g_i = 2\sigma(\cdot)$	52.08	27.20	50.00	40.85	36.62	27.25	8.44	17.20	34.53
	SSA: $g_i = 1 + \sigma(\cdot)$	52.77	25.60	55.56	43.90	32.33	23.50	9.00	19.00	34.03
	SSA: $g_i = \mathrm{SiLU}(\cdot)$	48.22	23.34	50.73	46.95	38.88	26.12	8.15	14.10	33.46
	SSA: $g_i = \sqrt{\mathrm{softplus}(\cdot)}$	52.16	25.30	56.65	36.59	36.00	26.11	9.19	17.90	34.59
	SSA: fixed $g = 10$	53.75	26.96	56.01	42.68	0.50	1.38	9.29	13.50	30.03
	SSA: fixed $g = 5$ (default)	**55.72**	26.88	**61.11**	**49.39**	33.12	27.75	**9.57**	18.50	**36.10**

L Additional Downstream Analyses

Per-checkpoint radar charts. Figure 13 shows the score difference of every method to the Baseline on every benchmark at each of the five checkpoints. Three patterns emerge. (i) SSA's early lead (Figure 3a) is broad rather than driven by one task: at 101B tokens it is ahead of the Baseline on eleven of fourteen benchmarks, with the largest gains on CMath, CRUXEval-I, HumanEval and DS-1000. (ii) By 524B tokens the polygons of all methods have contracted toward the Baseline, as expected when all runs approach the same data-limited regime, but SSA is the only method with a positive gain on all three math benchmarks at every checkpoint (Sink Attention and Gated Attention achieve this only at 524B). (iii) The benchmarks on which the prior methods gain most (CRUXEval-I for Sink and Gated Attention, MMLU-Pro for Gated Attention) are also those on which their gains fluctuate most between checkpoints, whereas SSA's CMath gain grows monotonically from 327B onward and its GSM8K gain never falls below $+2.5$ points.

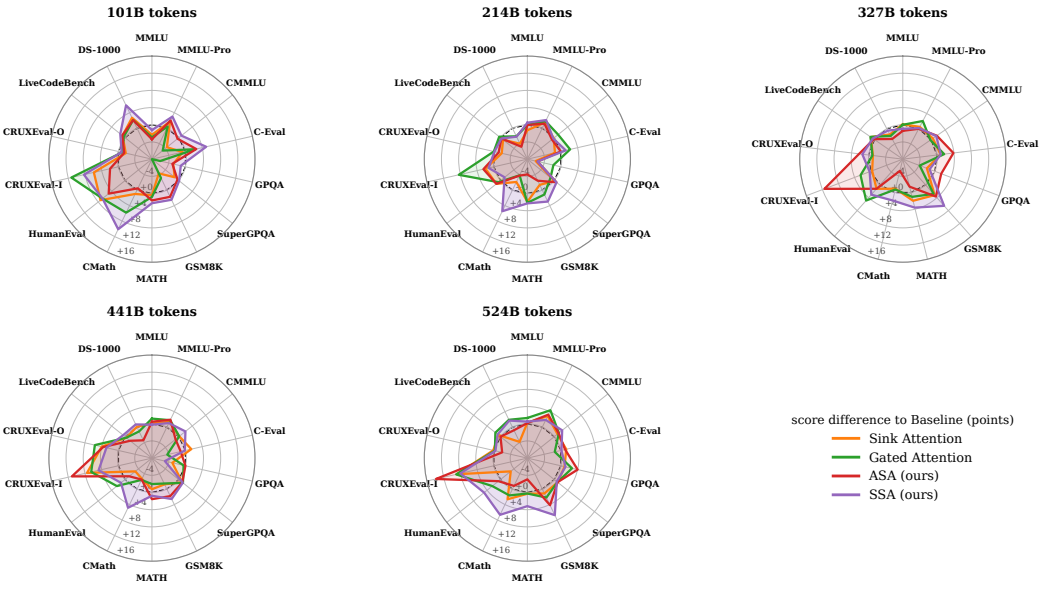


Figure 13: Score difference to the Softmax Baseline (points) on every benchmark at each evaluated checkpoint (radius clipped to $[-8, +16]$; SuperGPQA is omitted at 327B, where the ASA checkpoint has no score). The dashed circle marks zero.

Per-benchmark trajectories. Figure 14 plots every benchmark against training tokens for the five main methods. Most curves are still rising at 524B tokens, so none of the models is saturated. The math benchmarks show the cleanest separation: SSA is the top curve on GSM8K and CMath at every checkpoint and on MATH at four of five.

Domain aggregates. Table 10 aggregates the fourteen benchmarks into the three domains of Table 1 at every checkpoint. SSA leads the math aggregate at every checkpoint, with a margin over the best other method between $+2.4$ and $+4.2$ points.

Rank statistics and bootstrap intervals. We quantify how robust the aggregate comparison is to the choice of benchmarks: Table 11 reports, at 524B tokens, how often each method is best or top-two among the five, its mean rank, the number of benchmarks on which it beats the Baseline, and the mean gap to the Baseline and to Gated Attention with a 95% percentile bootstrap interval obtained by resampling the fourteen benchmarks with replacement (20,000 resamples) (Efron & Tibshirani, 1994). SSA is best on five benchmarks and beats the Baseline on thirteen of fourteen (two-sided sign test $p = 0.002$); its mean gap to the Baseline, $+2.74$, has interval $[+1.45, +4.19]$, and Gated Attention's $+2.00$ has interval $[+0.91, +3.35]$, so both are robust to the benchmark selection.

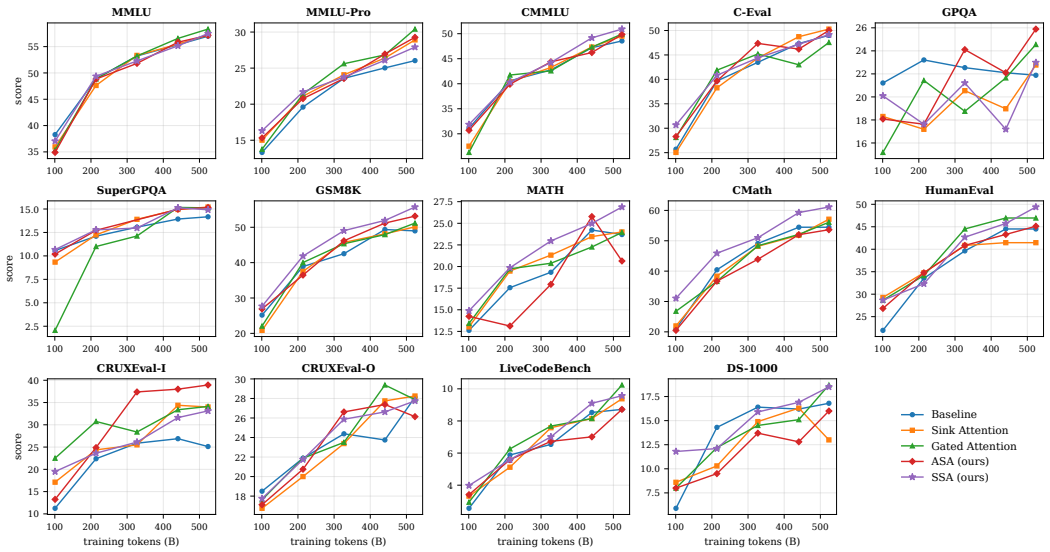


Figure 14: Every benchmark versus training tokens for the five main methods.

Table 10: Domain-level aggregates (unweighted means of the benchmarks in each domain) at every checkpoint; the number of benchmarks per domain is in parentheses. Best per column within each checkpoint in **bold**. ‡Knowledge over five benchmarks and All over thirteen at 327B (no SuperGPQA score for ASA).

Tokens	Method	Knowledge (6)	Math (3)	Code (5)	**All (14)**
101B	Baseline	23.36	19.58	12.03	18.50
	Sink Attention	21.86	18.61	15.01	18.72
	Gated Attention	20.12	20.75	15.94	18.76
	ASA (ours)	22.92	20.55	13.72	19.12
	SSA (ours)	**24.44**	**24.54**	**16.34**	**21.57**
214B	Baseline	30.68	32.30	19.60	27.07
	Sink Attention	29.47	31.78	18.91	26.19
	Gated Attention	**31.05**	32.13	**21.05**	**27.71**
	ASA (ours)	29.92	28.76	19.10	25.81
	SSA (ours)	30.43	**35.87**	19.08	27.54
327B‡	Baseline	37.12	36.99	22.56	31.49
	Sink Attention	37.11	38.41	22.44	31.77
	Gated Attention	37.07	38.03	23.71	32.15
	ASA (ours)	**38.24**	36.00	**25.06**	32.65
	SSA (ours)	37.18	**41.00**	23.52	**32.81**
441B	Baseline	35.16	42.68	23.97	32.78
	Sink Attention	35.31	41.14	25.61	33.09
	Gated Attention	35.09	40.76	**26.59**	33.27
	ASA (ours)	**35.38**	42.99	25.70	33.55
	SSA (ours)	34.99	**45.40**	25.99	**34.01**
524B	Baseline	36.10	42.38	24.65	33.36
	Sink Attention	37.32	43.74	25.22	34.37
	Gated Attention	37.65	43.74	27.58	35.36
	ASA (ours)	**37.89**	42.48	26.98	34.98
	SSA (ours)	37.23	**47.90**	**27.67**	**36.10**

Checkpoint-to-checkpoint volatility. Table 12 reports the mean absolute change of the per-benchmark scores between consecutive checkpoints and, for the last interval, the standard deviation of the per-benchmark changes. Between 441B and 524B tokens the average benchmark moves by

Table 11: Rank statistics over the fourteen benchmarks at 524B tokens and percentile-bootstrap 95% intervals (resampling benchmarks, one shared set of 20,000 resamples for all cells) for the mean gaps, computed from unrounded scores; tied scores receive the average rank. Best per column in **bold**.

Method	Best on (of 14)	Top-2 on (of 14)	Mean rank	Wins vs. Baseline	Mean gap vs. Baseline [95% CI]	Mean gap vs. Gated Attention [95% CI]
Baseline	0	1	4.32	–	–	−2.00 [−3.35, −0.91]
Sink Attention	3	5	2.93	12/14	+1.01 [−0.33, +2.59]	−0.98 [−2.19, +0.08]
Gated Attention	4	**8**	**2.29**	12/14	+2.00 [+0.91, +3.35]	–
ASA (ours)	2	6	3.11	9/14	+1.62 [−0.14, +3.88]	−0.38 [−1.46, +0.82]
SSA (ours)	**5**	**8**	2.36	**13/14**	**+2.74 [+1.45, +4.19]**	**+0.74 [−0.35, +1.93]**

1.0 (Baseline) to 2.3 (Gated Attention, ASA) points, with a standard deviation of 1.4–2.3 points across benchmarks. The fourteen-benchmark average has a noise level of roughly 0.4–0.6 points ($\text{std}/\sqrt{14}$). The SSA–Baseline gap of +2.74 is several times this level, and SSA is the best method at four of five checkpoints.

Table 12: Mean absolute change of the per-benchmark scores between consecutive checkpoints (points), and the standard deviation across benchmarks of the change over the last interval. Changes involving 327B exclude SuperGPQA.

Method	101B→214B	214B→327B	327B→441B	441B→524B	Std. of per-benchmark change, 441B→524B
Baseline	8.56	3.38	2.93	1.00	1.38
Sink Attention	7.63	4.50	2.96	1.81	1.95
Gated Attention	8.94	3.94	2.85	2.31	1.70
ASA (ours)	6.91	5.84	2.95	2.34	2.28
SSA (ours)	6.33	4.13	3.27	2.12	1.44

M Additional Loss and Stability Analyses

Loss gap and token speed-up. Figure 15(a) plots the smoothed loss of each method minus the smoothed Baseline loss (201-iteration centered window). Sink Attention's gap is within ±0.003 of zero for the entire run and Gated Attention's between −0.003 and −0.008; ASA averages −0.003 after 100B tokens. SSA's gap is an order of magnitude larger: it reaches −0.030 after about 40B tokens, narrows to −0.019 by 200B and to −0.014 at 524B tokens, but never closes—the gap shrinks in absolute terms as all losses flatten, while the token speed-up grows. Figure 15(b) makes this explicit: for every target loss reached by the Baseline after at least 60B tokens, we plot the ratio of the tokens the Baseline needed to the tokens SSA needed. The ratio rises from about 1.12× for early targets to 1.2–1.3× for targets in the second half of training (1.23× at the final Baseline loss with this window, 1.30× with the 51-iteration window of Figure 3b). The speed-up is therefore not a transient warm-up effect but increases with the training budget over the range we studied.

QK-logit statistics. Table 13 complements Figure 5 with the raw (unsmoothed) per-iteration maximum QK logit: its peak and when it occurred, the number and share of iterations above the clip threshold $\tau = 1385$ and above 0.9τ, the first iteration above τ, and the median and 95th percentile over the final quarter of training. Figure 16 shows the distribution over the final quarter. The Baseline, Sink Attention and Gated Attention are all clipped for the first time within the first 135B tokens and spend the rest of training near τ: their final-quarter medians (1165, 1389, 1392) show that Sink and Gated Attention are essentially pinned to the threshold—the clip is doing the work of keeping their logits bounded. ASA is clipped 63 times, all within the first 205B tokens, and settles at a median of 945; SSA peaks at 933 after 94B tokens—below τ by a third—and settles at a median of 692 with a 95th percentile of 754, i.e. its maximum logit is not only lower but also much less dispersed than that of any other method.

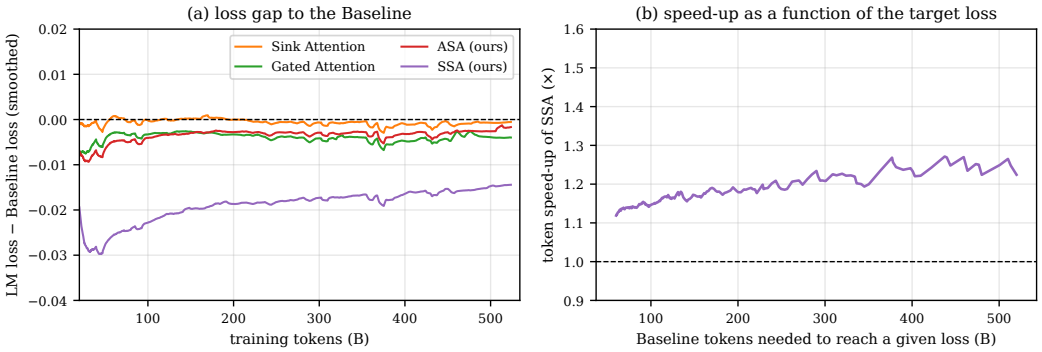


Figure 15: **(a)** Smoothed next-token loss minus the smoothed Baseline loss (201-iteration window). **(b)** Token speed-up of SSA as a function of the target loss, expressed by the number of tokens the Baseline needed to reach it.

Table 13: Statistics of the raw per-iteration maximum QK logit over the 524B-token runs. $\tau = 1385$ is the QK-clip threshold; "final 25%" refers to the last quarter of training iterations. Best values in **bold**.

Method	Peak	Peak at (B tokens)	Iterations $> \tau$	Share $> \tau$ (%)	Share $> 0.9\tau$ (%)	First $> \tau$ (B)	Median (final 25%)	95th pct. (final 25%)
Baseline	1810	65	817	5.8	39.4	47	1165	1381
Sink Attention	1579	146	4433	31.2	59.7	68	1389	1400
Gated Attention	1660	179	3150	22.2	25.9	131	1392	1396
ASA (ours)	1443	136	63	0.4	8.9	56	945	1061
SSA (ours)	**933**	94	**0**	**0.0**	**0.0**	–	**692**	**754**

N Larger-Scale Results

Table 14 gives the per-benchmark results summarized in Section 5.3, and Figure 17(a) the corresponding radar chart. Both runs use the "Large" configuration of Table 3. SSA is ahead on every math and code benchmark and on 11 of the 12 benchmarks overall.

Table 14: Larger-scale model (≈264B total / ≈12B activated parameters) after 44B training tokens; both runs share data, initialization and hyper-parameters. Best per row in **bold**; our method is shaded.

Benchmark	Gated Attention	SSA (ours)	Benchmark	Gated Attention	SSA (ours)
MMLU	25.25	**27.70**	GSM8K	11.83	**13.12**
MMLU-Pro	11.05	**11.62**	MATH	7.64	**9.92**
CMMLU	**21.12**	21.07	CMath	14.30	**14.57**
C-Eval	21.21	**21.85**	HumanEval	21.34	**24.39**
GPQA	16.07	**20.98**	LiveCodeBench	1.52	**2.56**
SuperGPQA	9.05	**9.69**	DS-1000	1.70	**5.60**
Avg (12 benchmarks)				13.51	**15.26**

O Long-Context Results

Table 15 lists every RULER task at 128K context for the three models of Section 5.4. The needle-in-a-haystack family comprises three single-needle tasks (S-1–S-3, with haystacks of increasing difficulty), three multi-key tasks (MK-1–MK-3, in which distractor needles share the key format of the target), a multi-value task (MV, several values stored under one key) and a multi-query task (MQ, several needles queried at once); VT is variable tracking, CWE and FWE are common- and frequent-words extraction, and QA-1 and QA-2 are the two question-answering tasks.

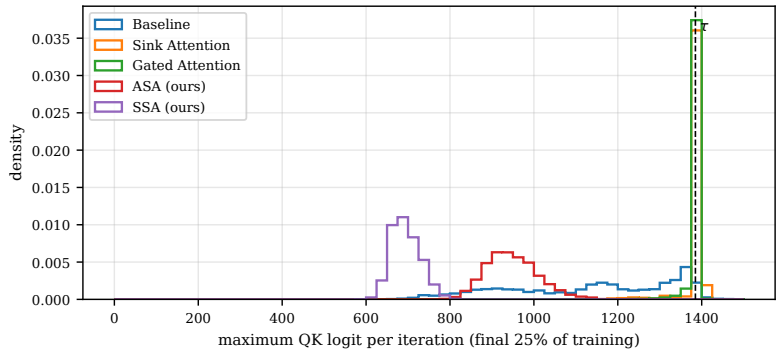


Figure 16: Distribution of the raw per-iteration maximum QK logit over the final quarter of training. The Sink Attention and Gated Attention distributions are concentrated at the clip threshold τ.

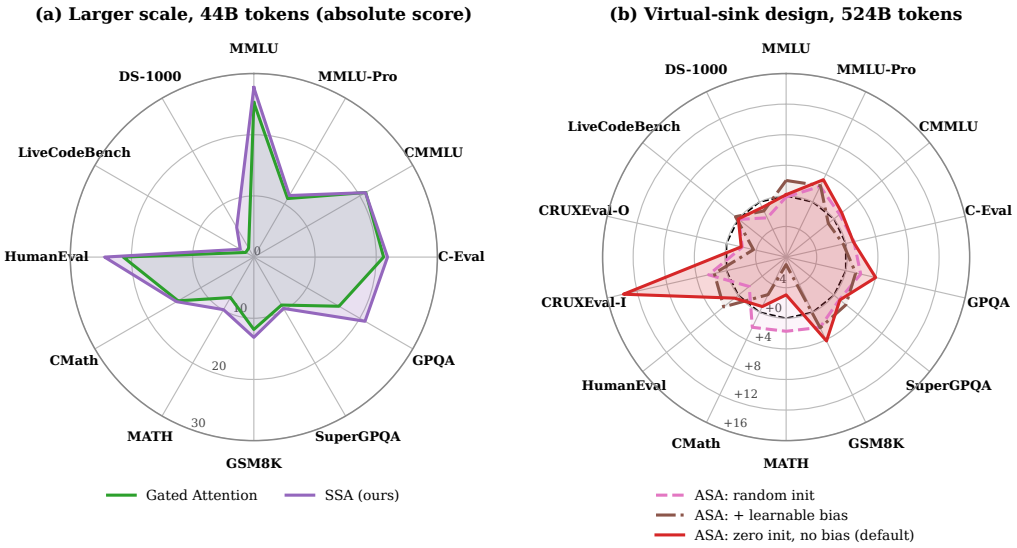


Figure 17: **(a)** Absolute scores of Gated Attention and SSA at the larger scale after 44B tokens. **(b)** Score difference to the Baseline (points) of the three virtual-sink designs of Table 4 after 524B tokens (20B model); the dashed circle marks zero.

Where SSA gains. SSA is the best or tied-best model on eight of the thirteen tasks and has the highest average by 4.5 points over Gated Attention and 5.7 points over the Baseline. Its advantage grows with the difficulty of the retrieval. The single-needle tasks are saturated for every model ($\geq$ 94%). On the multi-key tasks, where the model must reject distractors that look like the target, SSA scores 87.6, 94.2 and 52.6 against 79.2, 75.0 and 36.4 for the Baseline (+8.4, +19.2 and +16.2 points), and it is +10.4 and +11.4 points ahead of Gated Attention on MK-1 and MK-3. Variable tracking, which requires following a chain of variable assignments through the whole context, shows the same pattern (75.44 versus 71.04 and 61.76). The clearest separation is on question answering: SSA reaches 33.4 and 28.2 on QA-1 and QA-2, against 18.4 and 25.4 for the Baseline and 7.8 and 14.0 for Gated Attention. Question answering is the only task family in which the answer must be composed from the retrieved passage rather than copied, and it is the family on which the prior remedy falls furthest below the Baseline; SSA improves on the Baseline instead, nearly doubling its QA-1 accuracy.

Where the models are close. On multi-value and multi-query retrieval the three models lie within a few points of one another (the Baseline is ahead of SSA by 3.2 and 4.9 points), as they do on the hardest single-needle task (at most 5.6 points apart); these tasks are close to solved at this context

Table 15: Per-task RULER accuracy (%) at 128K context after long-context continued pre-training. **Avg** (shaded) is the unweighted mean over all thirteen tasks. Best per column in **bold**.

	Needle-in-a-haystack									Aggregation		QA		
Method	S-1	S-2	S-3	MK-1	MK-2	MK-3	MV	MQ	VT	CWE	FWE	QA-1	QA-2	**Avg**
Baseline	**100.00**	**100.00**	**99.78**	79.20	75.00	36.40	**96.30**	**95.45**	61.76	0.16	59.53	18.40	25.40	65.18
Gated Attention	**100.00**	**100.00**	99.31	77.20	**97.40**	41.20	87.15	92.05	71.04	**3.70**	71.33	7.80	14.00	66.32
SSA (ours)	**100.00**	**100.00**	94.20	**87.60**	94.20	**52.60**	93.10	90.55	**75.44**	0.36	**71.53**	**33.40**	**28.20**	**70.86**

length and do not separate the methods. Common-words extraction is near zero for every model (≤ 3.7), so the aggregation group is decided by frequent-words extraction, on which SSA is best (71.53 versus 71.33 and 59.53).

Interpretation. The pattern matches the collateral-damage analysis of Section 3. Gated Attention improves over the Baseline on tasks that reward suppressing irrelevant context (MK-2, VT, FWE) but loses half of the Baseline's question-answering accuracy, as expected from a gate that shrinks the attention on informative tokens together with the sink. SSA obtains the same or larger gains on the retrieval and tracking tasks without this regression, because its virtual sink drains mass only where the attention evidence is weak and the gain restores the mass on the evidence that remains. The sink projection itself needs no adaptation to the $32\times$ longer context: the per-(token, head) sink logit is attached to the query shard under context parallelism (Appendix C), and the continued pre-training recipe is identical for all three models.

P Full Results at All Checkpoints

Tables 16 and 17 report every benchmark for the five main methods at all five checkpoints.

Table 16: Knowledge and reasoning benchmarks of all main methods at the five evaluated checkpoints. Best per column within each checkpoint in **bold**. ‡The ASA checkpoint at 327B tokens has no SuperGPQA score; the fourteen-benchmark averages at this checkpoint (Table 17, Figures 3a and 12) are therefore taken over the remaining thirteen benchmarks for all methods.

Tokens	Method	MMLU	MMLU-Pro	CMMLU	C-Eval	GPQA	SuperGPQA
101B	Baseline	**38.28**	13.31	31.08	25.71	**21.21**	10.57
	Sink Attention	35.95	15.01	27.53	25.05	18.30	9.34
	Gated Attention	35.35	13.75	26.28	28.11	15.18	2.06
	ASA (ours)	34.90	15.32	30.70	28.30	18.08	10.19
	SSA (ours)	37.05	**16.31**	**31.86**	**30.66**	20.09	**10.67**
214B	Baseline	48.89	19.60	40.58	39.67	**23.21**	12.10
	Sink Attention	47.62	21.10	40.39	38.27	17.19	12.24
	Gated Attention	48.86	21.40	**41.74**	**41.88**	21.43	11.00
	ASA (ours)	48.84	20.77	39.90	39.66	17.63	12.73
	SSA (ours)	**49.37**	**21.70**	40.29	40.84	17.63	**12.78**
327B‡	Baseline	53.22	23.58	42.76	43.49	22.54	13.06
	Sink Attention	**53.33**	24.07	43.25	44.37	20.54	**13.90**
	Gated Attention	53.22	**25.61**	42.57	45.20	18.75	12.12
	ASA (ours)	51.81	23.56	**44.38**	**47.36**	**24.11**	–
	SSA (ours)	52.28	23.75	44.24	44.40	21.21	12.97
441B	Baseline	55.36	25.03	47.20	47.33	**22.10**	13.93
	Sink Attention	55.45	26.40	47.34	**48.73**	18.97	14.95
	Gated Attention	**56.56**	26.83	47.32	42.99	21.65	**15.17**
	ASA (ours)	55.84	**26.95**	46.23	46.20	**22.10**	14.97
	SSA (ours)	55.15	26.09	**49.17**	47.19	17.19	15.13
524B	Baseline	56.98	26.04	48.55	49.01	21.88	14.17
	Sink Attention	57.22	28.91	49.52	**50.29**	22.77	**15.21**
	Gated Attention	**58.33**	**30.41**	49.93	47.55	24.55	15.12
	ASA (ours)	57.14	29.27	49.85	50.02	**25.89**	15.16
	SSA (ours)	57.48	27.91	**50.92**	49.15	22.99	14.92

Table 17: Math and code benchmarks of all main methods at the five evaluated checkpoints, and the average over all fourteen benchmarks (shaded; thirteen at 327B‡). CRUX-I/-O denote CRUXEval-Input/-Output, LCB denotes LiveCodeBench. Best per column within each checkpoint in **bold**.

Tokens	Method	GSM8K	MATH	CMath	HumanEval	CRUX-I	CRUX-O	LCB	DS-1000	**Avg**
101B	Baseline	25.17	12.62	20.95	21.95	11.25	**18.50**	2.56	5.90	18.50
	Sink Attention	20.85	13.02	21.95	**29.27**	17.12	16.75	3.32	8.60	18.72
	Gated Attention	22.06	13.42	26.78	28.66	**22.50**	17.62	2.94	8.00	18.76
	ASA (ours)	26.91	14.24	20.49	26.83	13.25	17.12	3.41	8.00	19.12
	SSA (ours)	**27.67**	**14.88**	**31.06**	28.66	19.50	17.75	**3.98**	**11.80**	**21.57**
214B	Baseline	38.89	17.56	40.44	33.54	22.38	**21.88**	5.88	**14.30**	27.07
	Sink Attention	37.53	19.48	38.34	**34.76**	24.38	20.00	5.12	10.30	26.19
	Gated Attention	40.03	19.74	36.61	34.15	**30.75**	**21.88**	**6.26**	12.20	**27.71**
	ASA (ours)	36.47	13.12	36.70	**34.76**	24.88	20.75	5.59	9.50	25.81
	SSA (ours)	**41.85**	**19.86**	**45.90**	32.32	23.62	21.75	5.59	12.10	27.54
327B‡	Baseline	42.53	19.34	49.09	39.63	25.87	24.38	6.54	**16.40**	31.49
	Sink Attention	45.72	21.32	48.18	40.85	25.50	23.38	7.58	14.90	31.77
	Gated Attention	45.34	20.38	48.36	**44.51**	28.38	23.50	**7.68**	14.50	32.15
	ASA (ours)	46.17	17.94	43.90	40.85	**37.38**	**26.62**	6.73	13.70	32.65
	SSA (ours)	**49.05**	**22.96**	**51.00**	42.68	26.12	25.87	7.01	15.90	**32.81**
441B	Baseline	49.36	24.22	54.46	44.51	26.88	23.75	8.53	16.20	32.78
	Sink Attention	48.14	23.46	51.82	41.46	34.38	27.75	8.15	16.30	33.09
	Gated Attention	47.92	22.26	52.09	**46.95**	33.38	**29.38**	8.15	15.10	33.27
	ASA (ours)	51.18	**25.78**	52.00	43.29	**38.00**	27.38	7.01	12.80	33.55
	SSA (ours)	**51.93**	24.98	**59.29**	45.73	31.62	26.62	**9.10**	**16.90**	**34.01**
524B	Baseline	48.98	23.70	54.46	44.51	25.12	28.12	8.72	16.80	33.36
	Sink Attention	50.11	24.00	57.10	41.46	34.00	**28.25**	9.38	13.00	34.37
	Gated Attention	51.18	23.94	56.10	46.95	34.12	27.88	**10.24**	**18.70**	35.36
	ASA (ours)	53.15	20.64	53.64	45.12	**38.94**	26.14	8.72	16.00	34.98
	SSA (ours)	**55.72**	**26.88**	**61.11**	**49.39**	33.12	27.75	9.57	18.50	**36.10**