Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free

Zihan Qiu et al.
Presented by: Fengnan Li

Duke B&B

Jan, 2026

Outline

Background: Attention Sink

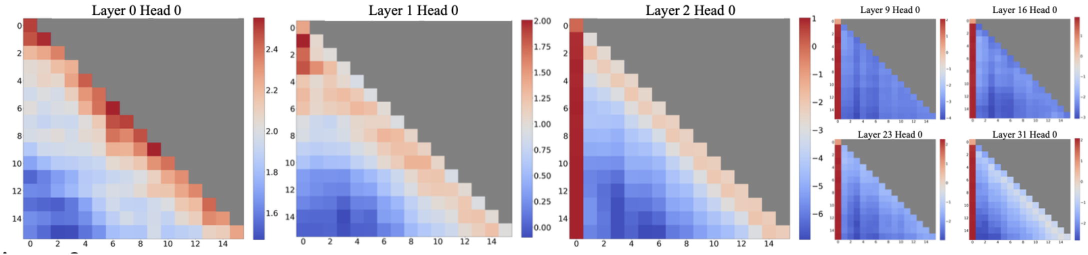


Figure: (1) The attention maps in the first two layers (layers 0 and 1) exhibit the "local" pattern, with recent tokens receiving more attention. (2) Beyond the bottom two layers, the model heavily attends to the initial token across all layers and heads. (Xiao, et al., ICLR 2024)

Background: Massive Activation

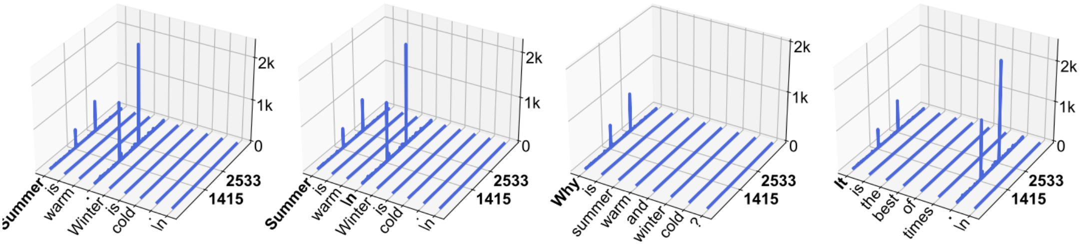


Figure: Outlier hidden state values:Activations with massive magnitudes appear in two fixed feature dimensions (1415, 2533), and two types of tokens—the starting token, and the first period (.) or newline token (\n)

Background: Evolution of Gating Mechanisms

- **Historical Context**: Gating mechanisms like LSTMs and Highway Networks pioneered controlled information flow.
- **Modern Usage**: Currently standard in FFN layers (SwiGLU) and token-mixers like State Space Models (SSMs).
- **The Research Gap**: Existing literature rarely examines the specific effects of gating within standard softmax attention.

Mathematical Formulation of Gated Attention

The gating mechanism is formalized to modulate an input Y using a secondary input X:

$$Y' = g(Y, X, W_\theta, \sigma) = Y \odot \sigma(XW_\theta) \tag{1}$$

Key Components:

- **Dynamic Filter**: The term $\sigma(XW_\theta)$ acts as a filter to preserve or erase features.
- **Activation Function**: Typically uses **sigmoid** to constrain scores in $[0, 1]$.
- **Granularity**: Explored headwise (scalar per head) vs. elementwise (vector per head) modulation.

Investigated Gating Positions

The authors systematically compared five positions (G_1 to G_5):

- G_1: Following SDPA output
- G_2: After the Value projection
- G_3: After the Key projection
- G_4: After the Query projection
- G_5: After the final dense output layer

Key Finding: Gating after SDPA (G_1) yields the most significant performance gains

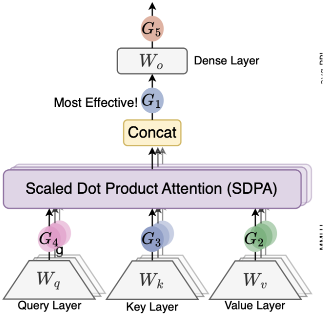


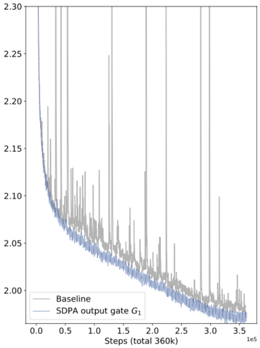
Figure: Illustration of investigated gating positions G_1 to G_5.

Results on MoE models

Method	Act Func	Score Shape	Added Param	Avg PPL	Hellaswag	MMLU	GSM8k	C-eval
Reference Baselines (Baseline uses $q = 32, k = 4$. All methods use $d_k = 128$.)								
(1) Baseline	-	-	0	6.026	73.07	58.79	52.92	60.26
(2) $k = 8$	-	-	50	5.979	73.51	59.78	52.16	62.26
(3) $q = 48$	-	-	201	5.953	73.59	58.45	53.30	59.67
(4) Add 4 Experts	-	-	400	5.964	73.19	58.84	52.54	**63.19**
Gating Position Variants								
(5) SDPA Elementwise G_1	sigmoid	$n \times q \times d_k$	201	**5.761**	74.64	**60.82**	**55.27**	62.20
(6) v Elementwise G_2	sigmoid	$n \times k \times d_k$	25	5.820	74.38	59.17	53.97	61.00
(7) k Elementwise G_3	sigmoid	$n \times k \times d_k$	25	6.016	72.88	59.18	50.49	61.74
(8) q Elementwise G_4	sigmoid	$n \times q \times d_k$	201	5.981	73.01	58.74	53.97	62.14
(9) Dense Output G_5	sigmoid	$n \times d_{\text{model}}$	100	6.017	73.32	59.41	50.87	59.43
Gating Granularity Variants								
(10) SDPA Headwise G_1	sigmoid	$n \times q$	1.6	5.792	74.50	60.05	54.44	62.61
(11) v Headwise G_2	sigmoid	$n \times q$	0.2	5.808	74.38	59.32	53.53	62.61
Head-Specific v.s. Head-Shared Gating								
(12) SDPA Head-Shared G_1	sigmoid	$n \times d_k$	201	5.801	74.34	60.06	53.15	61.01
(13) v Head-Shared G_2	sigmoid	$n \times d_k$	25	5.867	74.10	59.02	53.03	60.61
Multiplicative v.s. Additive								
(14) SDPA Additive G_1	SiLU	$n \times q \times d_k$	201	5.821	**74.81**	60.06	53.30	60.98
Activation Variants								
(15) SDPA Elementwise G_1	SiLU	$n \times q \times d_k$	201	5.822	74.22	60.49	54.59	62.34

- Training 15A2B MoE on 400B tokens: 1) SDPA and value output gating are effective. 2) Head-Specific Gating Matters. 3) Multiplicative Gating is Preferred. 4) Sigmoid Activation is Better.

Results on Dense Models



Method	Max LR	Avg PPL	HumanEval	MMLU	GSM8k	Hellaswag	C-eval	CMMLU
28 Layer, 1.7B Parameters, **400B Tokens**, Batch Size=1024								
(1) Baseline	4.0×10^{-3}	7.499	28.66	50.21	27.82	64.94	49.15	49.52
(2) SDPA Elementwise	4.0×10^{-3}	**7.404**	**29.27**	**51.15**	**28.28**	**65.48**	**50.72**	**50.72**
28 Layer, 1.7B Parameters, **3.5T Tokens**, Batch Size=2048								
(3) Baseline	4.5×10^{-3}	6.180	34.15	59.10	69.07	68.02	68.19	64.95
(4) SDPA Elementwise	4.5×10^{-3}	**6.130**	**37.80**	**59.61**	**70.20**	**68.84**	**68.52**	**65.76**
48 Layer, 1.7B Parameters, **400B Tokens**, Batch Size=1024								
(5) Baseline	4.0×10^{-3}	7.421	28.05	52.04	32.98	65.96	51.11	51.86
(6) Baseline	8.0×10^{-3}	9.195	21.34	44.28	15.24	57.00	43.11	42.63
(7) Baseline+Sandwich Norm	8.0×10^{-3}	7.407	30.49	52.07	32.90	66.00	52.04	51.72
(8) SDPA Elementwise	4.0×10^{-3}	**7.288**	**31.71**	52.44	32.37	66.28	52.06	52.29
(9) SDPA Headwise	4.0×10^{-3}	7.370	31.10	53.83	34.12	65.59	**55.07**	52.38
(10) SDPA Elementwise	8.0×10^{-3}	7.325	31.10	**54.47**	**36.62**	**66.40**	53.91	**53.80**
48 Layer, 1.7B Parameters, **1T Tokens**, Batch Size=4096								
(11) Baseline	5.3×10^{-3}	7.363	29.88	54.44	32.22	65.43	53.72	53.37
(12) Baseline	8.0×10^{-3}	-	-	-	-	-	-	-
(13) SDPA Elementwise	5.3×10^{-3}	7.101	**34.15**	55.70	36.69	67.17	54.51	54.68
(14) SDPA Elementwise	8.0×10^{-3}	**7.078**	31.71	**56.47**	**39.73**	**67.38**	**55.52**	**55.77**

Gating is effective across all settings

Non-linearity: Addressing the Low-Rank Problem

The output of the k-th head in standard multi-head attention:

$$o_i^k = \left(\sum_{j=0}^{i} S_{ij}^k \cdot X_j W_V^k \right) W_O^k = \sum_{j=0}^{i} S_{ij}^k \cdot X_j (W_V^k W_O^k) \tag{2}$$

- **Limitation**: Consecutive linear layers W_V and W_O merge into one low-rank mapping since $d_k < d_{model}$.
- **The Fix**: Introducing non-linearity via gating at G_1 or G_2 increases expressiveness:

$$o_i^k = \left(\sum_{j=0}^{i} S_{ij}^k \cdot \text{Non-Linearity-Map}(X_j W_V^k) \right) W_O^k \tag{3}$$

$$o_i^k = \text{Non-Linearity-Map} \left(\sum_{j=0}^{i} S_{ij}^k \cdot X_j W_V^k \right) W_O^k \tag{4}$$

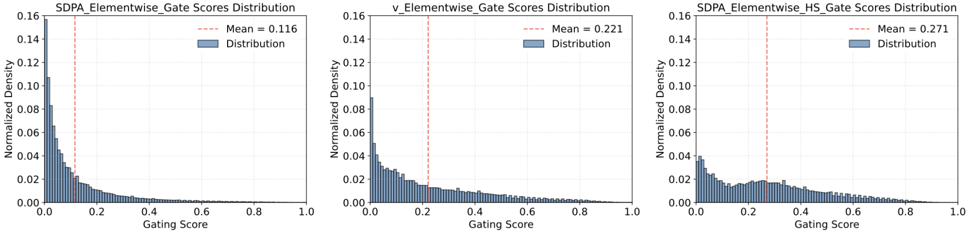
Non-linearity: Addressing the Low-Rank Problem

Method	Activation Function	Avg PPL	Hellaswag	MMLU	GSM8k	C-eval
(1) Baseline	-	6.026	73.07	58.79	52.92	60.26
(2) SDPA Elementwise Gate	Sigmoid	**5.761**	74.64	**60.82**	**55.27**	**62.20**
(3) v Elementwise Gate	Sigmoid	5.820	74.38	59.17	53.97	61.00
(4) SDPA Additive Gate	SiLU	5.821	**74.81**	60.06	53.30	60.98
(5) SDPA GroupNorm	RMSNorm	5.847	74.10	60.15	53.75	61.14
(6) SDPA SiLU	SiLU	5.975	73.34	59.55	53.19	60.90
(7) SDPA Additive Gate	Identity	5.882	74.17	59.20	52.77	59.86

Performance of different (non)-linearity augmentations.

Query-Dependent Sparsity

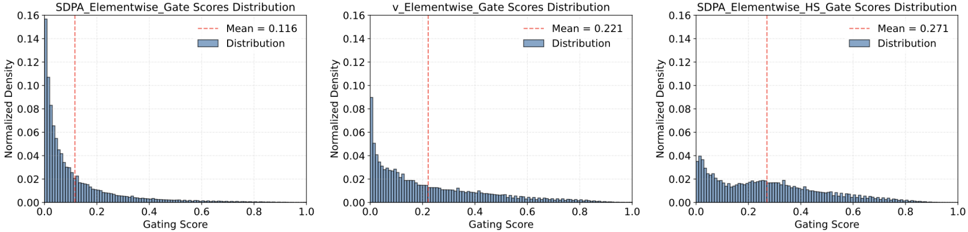
Method	Act-Func	Gate Score	M-Act	F-Attn	PPL	Hellaswag	MMLU	GSM8k
(1) Baseline	-	-	1053	0.467	6.026	73.07	58.79	52.92
(2) SDPA Elementwise Gate	Sigmoid	0.116	94	0.048	**5.761**	**74.64**	**60.82**	**55.27**
(3) SDPA Headwise Gate	Sigmoid	0.172	98	0.073	5.792	74.50	60.05	54.44
(4) SDPA Elementwise Head-shared Gate	Sigmoid	0.271	286	0.301	5.801	74.34	60.06	53.15
(5) v Elementwise Gate	Sigmoid	0.221	125	0.297	5.820	74.38	59.17	51.33
(6) SDPA Input Independent Gate	Sigmoid	0.335	471	0.364	5.917	73.64	59.02	52.40
(7) SDPA Elementwise Gate	NS-sigmoid	0.653	892	0.451	5.900	74.05	60.05	52.75



- (i) **Effective Gating Scores are Sparse**. SDPA output gatings exhibit the lowest mean gating scores. Furthermore, the SDPA output gating score distribution shows a high concentration near 0.

Query-Dependent Sparsity

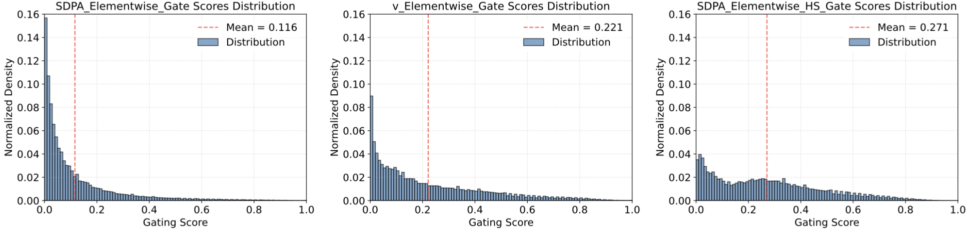
Method	Act-Func	Gate Score	M-Act	F-Attn	PPL	Hellaswag	MMLU	GSM8k
(1) Baseline	-	-	1053	0.467	6.026	73.07	58.79	52.92
(2) SDPA Elementwise Gate	Sigmoid	0.116	94	0.048	**5.761**	**74.64**	**60.82**	**55.27**
(3) SDPA Headwise Gate	Sigmoid	0.172	98	0.073	5.792	74.50	60.05	54.44
(4) SDPA Elementwise Head-shared Gate	Sigmoid	0.271	286	0.301	5.801	74.34	60.06	53.15
(5) v Elementwise Gate	Sigmoid	0.221	125	0.297	5.820	74.38	59.17	51.33
(6) SDPA Input Independent Gate	Sigmoid	0.335	471	0.364	5.917	73.64	59.02	52.40
(7) SDPA Elementwise Gate	NS-sigmoid	0.653	892	0.451	5.900	74.05	60.05	52.75



- **(ii) Query-Dependency Matters.** The scores for value gating (G_2) are higher than those for SDPA output gating (G_1), and the performance is inferior. Key difference: G_1 uses the hidden state of the current query (query-dependent), while G_2 uses those from past k and v.

Query-Dependent Sparsity

Method	Act-Func	Gate Score	M-Act	F-Attn	PPL	Hellaswag	MMLU	GSM8k
(1) Baseline	-	-	1053	0.467	6.026	73.07	58.79	52.92
(2) SDPA Elementwise Gate	Sigmoid	0.116	94	0.048	**5.761**	**74.64**	**60.82**	**55.27**
(3) SDPA Headwise Gate	Sigmoid	0.172	98	0.073	5.792	74.50	60.05	54.44
(4) SDPA Elementwise Head-shared Gate	Sigmoid	0.271	286	0.301	5.801	74.34	60.06	53.15
(5) v Elementwise Gate	Sigmoid	0.221	125	0.297	5.820	74.38	59.17	51.33
(6) SDPA Input Independent Gate	Sigmoid	0.335	471	0.364	5.917	73.64	59.02	52.40
(7) SDPA Elementwise Gate	NS-sigmoid	0.653	892	0.451	5.900	74.05	60.05	52.75



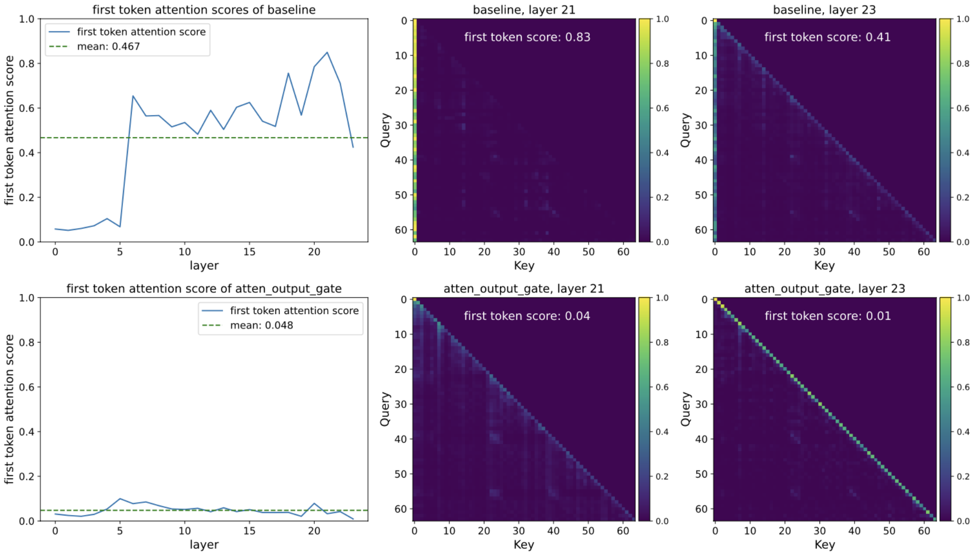
- **(iii) Less Sparse Gating is Worse.** To further validate the importance of gating sparsity, the authors reduce sparsity from the gating formulation by using a modified non-sparse (NS) sigmoid: $\text{NS-sigmoid}(x) = 0.5 + 0.5 \cdot \text{sigmoid}(x)$

Eliminating Attention Sinks and Massive Activations

- **Attention Sink**: Baseline models allocate $\approx 46.7\%$ of attention to the first token. Gating reduces this to **4.8%**.
- **Stability**: Sparse gating reduces massive activations ($M - Act$), preventing loss spikes during training.

Method	**M-Act**	**F-Attn**	**PPL**
Baseline	1053	0.467	6.026
SDPA Gate	**94**	**0.048**	**5.761**

Eliminating Attention Sinks and Massive Activations



Eliminating Attention Sinks and Massive Activations

- **Why can sparse gating reduce MA and attention sink?**
  - Attention sink *"depresses"* the scores of irrelevant tokens
    - it is difficult to make all irrelevant keys yield very negative logits
    - it is relatively easy to assign **very few** unimportant keys large positive logits
  - Gating after SDPA can filter out irrelevant information for the query
    - No need to *"sink"*
  - MA contributes to the emergence of attention sinks
    - Without attention sink, a large portion of MA would be unnecessary

- **The benefit of reducing MA and attention sink**
  - Less MA (large activations) and attention sink (large attention logits) can reduce the numerical error

Analysis: Gating Facilitates Context Length Extension

Method	4k	8k	16k	32k	64k	128k
Baseline	88.89	85.88	83.15	79.50	–	–
SDPA-Gate	90.56	87.11	84.61	79.77	–	–
YaRN Extended						
Baseline	82.90 (-6.0)	71.52 (-14.4)	61.23 (-21.9)	37.94 (-41.56)	37.51	31.65
SDPA-Gate	88.13 (-2.4)	80.01 (-7.1)	76.74 (-7.87)	72.88 (-6.89)	66.60	58.82

- Attention sink works input-independently to modify attention score distribution (the denominator of softmax function)
  - Increase context length $\rightarrow$ both the numerator and denominator of softmax changes $\rightarrow$ the logits corresponding to attention sink is fixed $\rightarrow$ Hard to generalize to new context length
- Attention output gate uses the input dependent gating scores

Final Summary

- **Position Matters**: Gating after SDPA (G_1) is the most effective modification for standard attention.
- **Dual Benefits**: It provides both **non-linearity** (improving capacity) and **sparsity** (filtering context).
- **Training Stability**: Nearly eliminates loss spikes and massive activations, enabling larger learning rates and batch sizes.
- **Innovation**: This work presents the first **attention-sink-free** models that generalize better to long sequences.

Thank You! Any Questions?