Attribution-based Sparse Activation in Large Language Models

Jifeng Song*, Xiangyu Yin*, Boyuan Yang, Kai Huang, Weichen Liu, Wei Gao

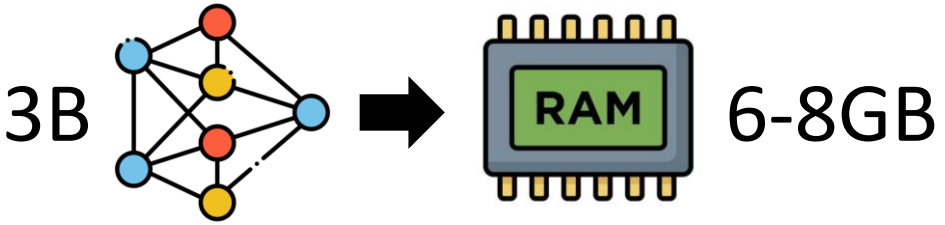
University of Pittsburgh

MLSys 2026

*Equal Contribution

LLM Inference Is Expensive

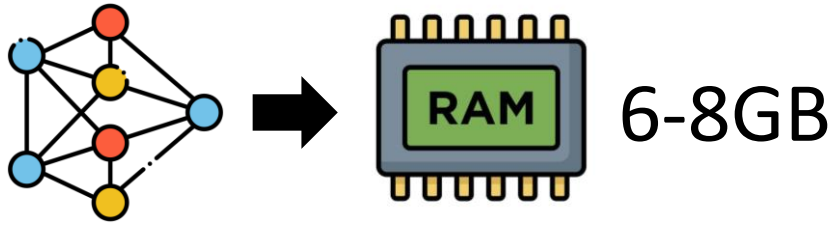
ChatGPT: daily energy use of tens of thousands US household



Out-of-box 3B model: won‘t fit on most phones; flagship phones – a few tokens per sec

LLM Inference Is Expensive

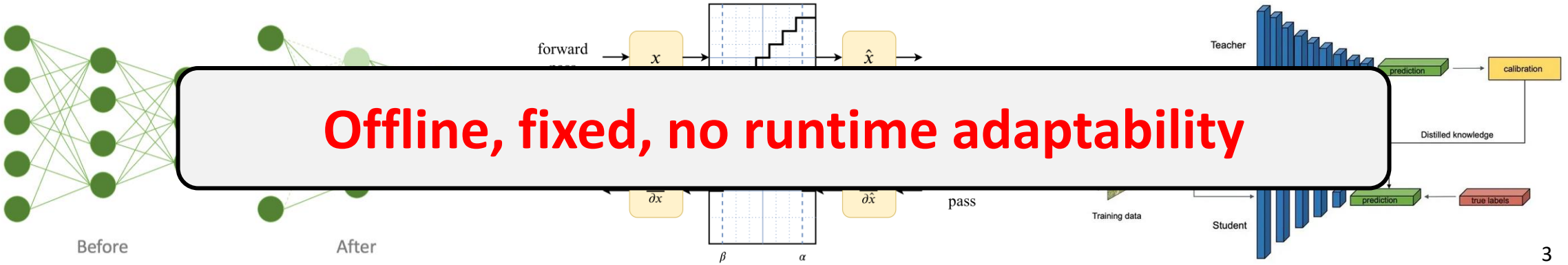
ChatGPT: daily energy use of tens of thousands US household



Out-of-box 3B model: won‘t fit on most phones; flagship phones – a few tokens per sec

Solutions?

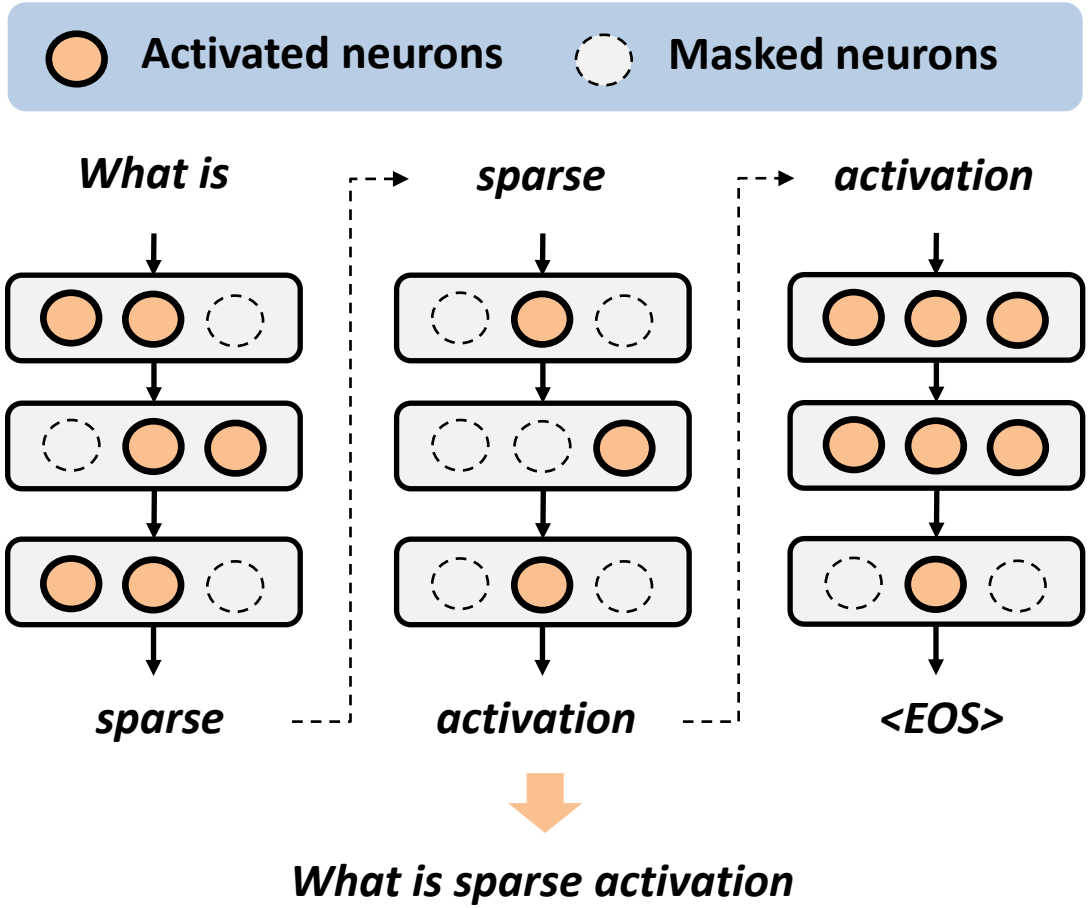
Pruning *Quantization* *Distillation*



Offline, fixed, no runtime adaptability

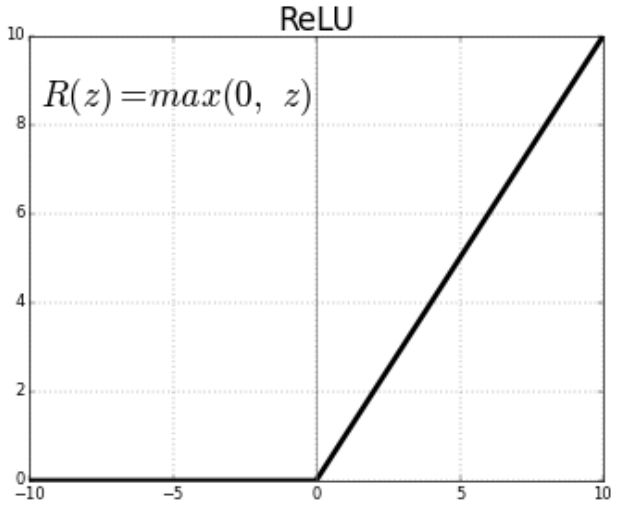
Sparse Activation

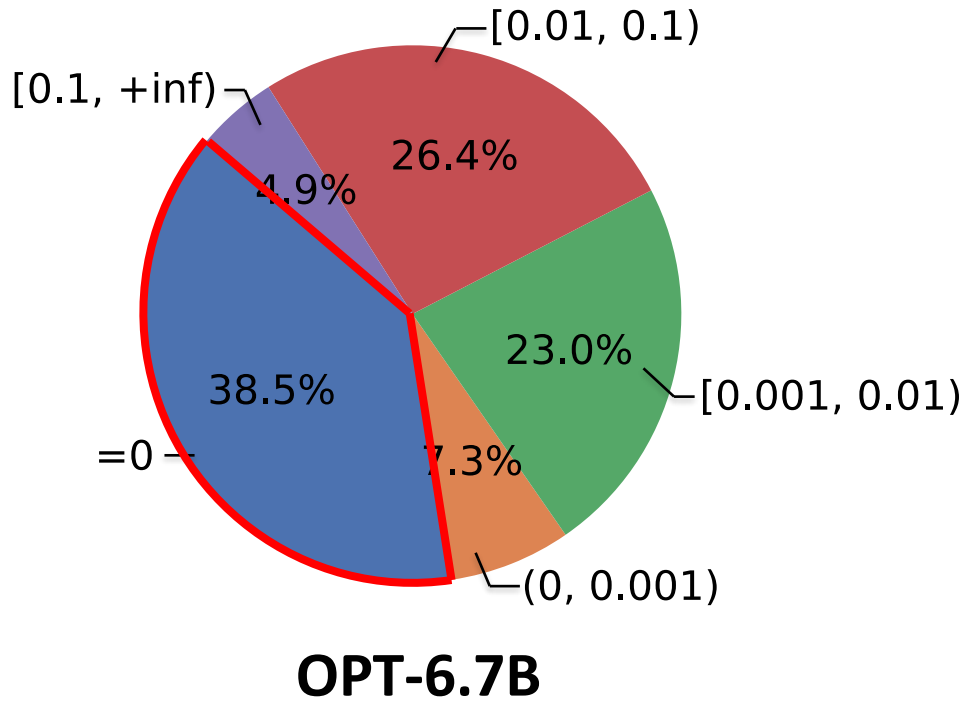
- **Per-input neuron selection**
  - Deactivate unimportant neurons at runtime
- **No model retraining is needed**
  - **Complementary** to techniques like quantization, distillation, etc.
- **Key question:** which neurons to deactivate?



Existing Work Skips Zero-Output Neurons

- **OPT + ReLU**
  - Many neurons output exactly zero
  - Zero contribution to inference



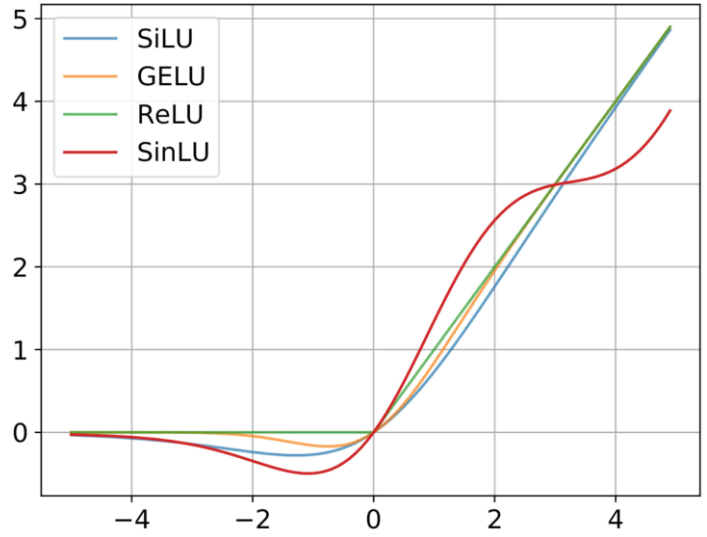


- Up to 70% sparsity, lossless on OPT [1] [2]

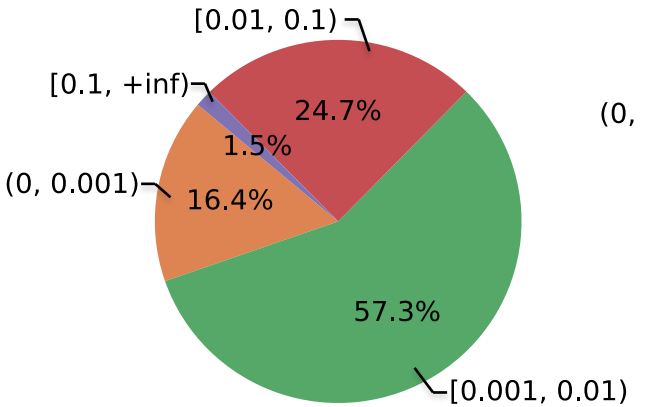
[1] Liu, Zichang, et al. "Deja vu: Contextual sparsity for efficient llms at inference time." *International Conference on Machine Learning*. PMLR, 2023.
[2] Song, Yixin, et al. "Powerinfer: Fast large language model serving with a consumer-grade gpu." Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles. 2024.

Modern LLMs are Different

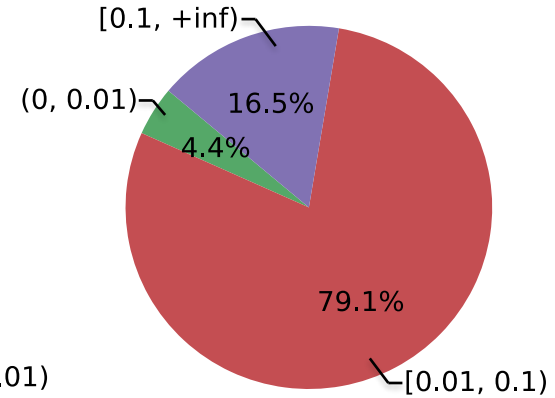
- **Modern LLMs (e.g., Llama, Phi, Gemma) use GeLU / SiLU**
  - Almost no zero-output neurons



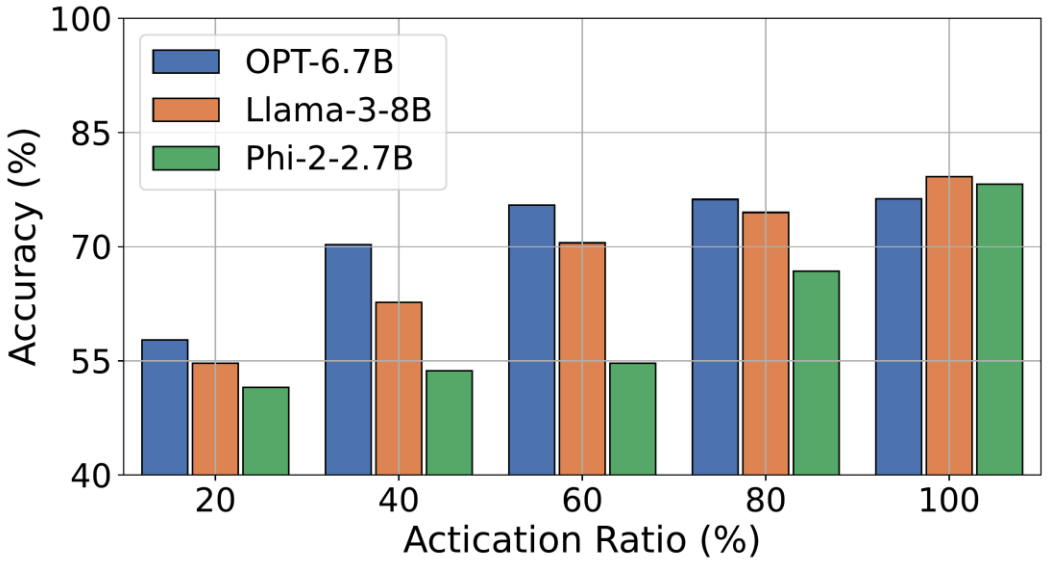
- Deactivating small-magnitude neurons results in large accuracy loss



Llama-3-8B

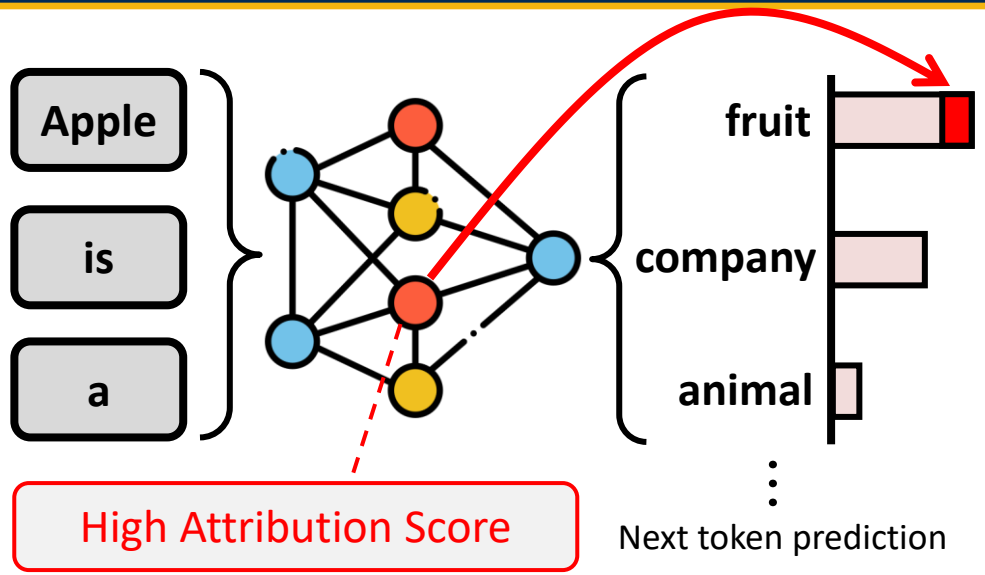


Phi-2-2.7B

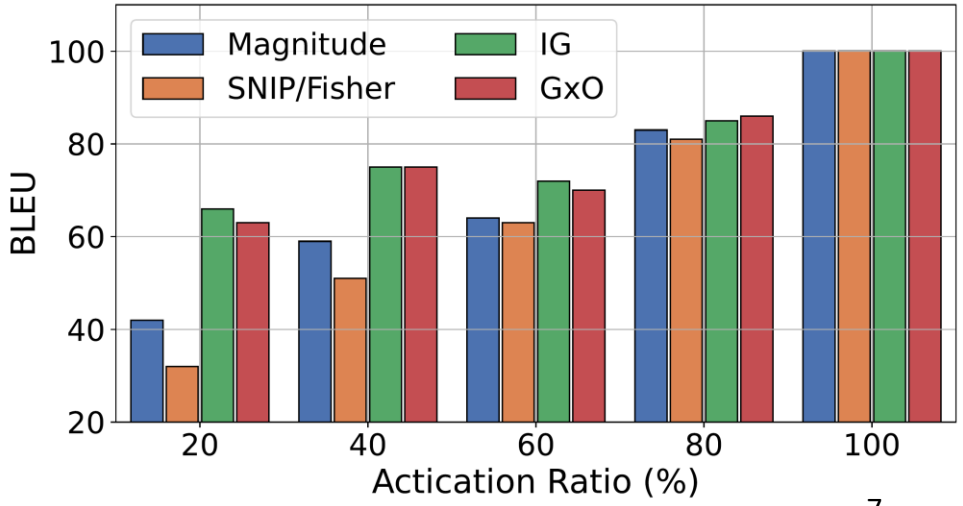


Our Approach: Attribution-based Sparse Activation

- **Attribution:** how much does this neuron contribute to the LLM output?
- Sparse activation ranks neurons by contribution, not magnitude

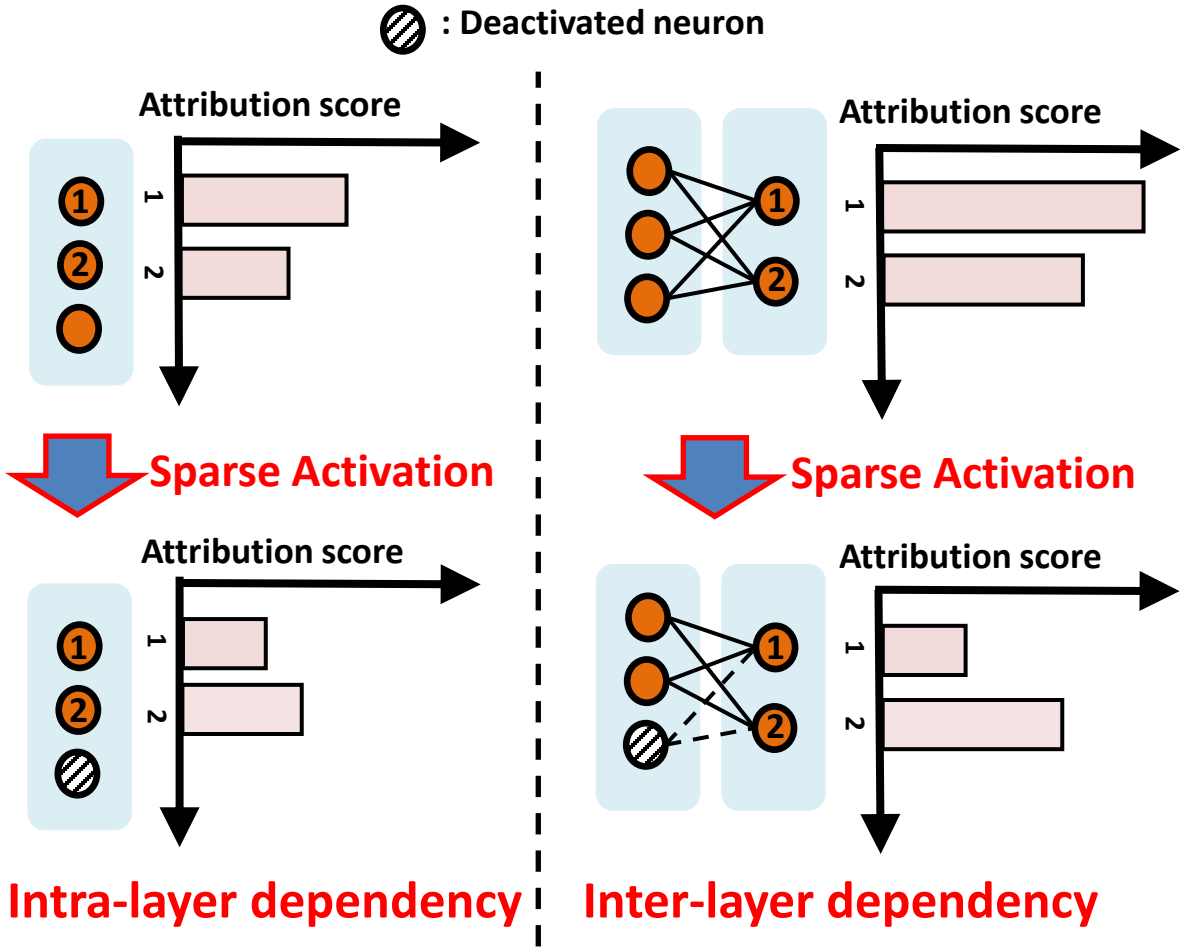


- **Choice of attribution metrics**
  - Integrated Gradients (IG): $\frac{1}{n}\sum_{k=1}^{n} \partial F(\frac{k}{n} \cdot x)/\partial x \cdot x$
    - Accurate, but expensive
  - **Gradient × Output (GxO):** $\frac{\partial F(x)}{\partial x} \cdot x$
    - Only need 1 forward + 1 backward pass
    - *Similar accuracy, better efficiency compared to IG*



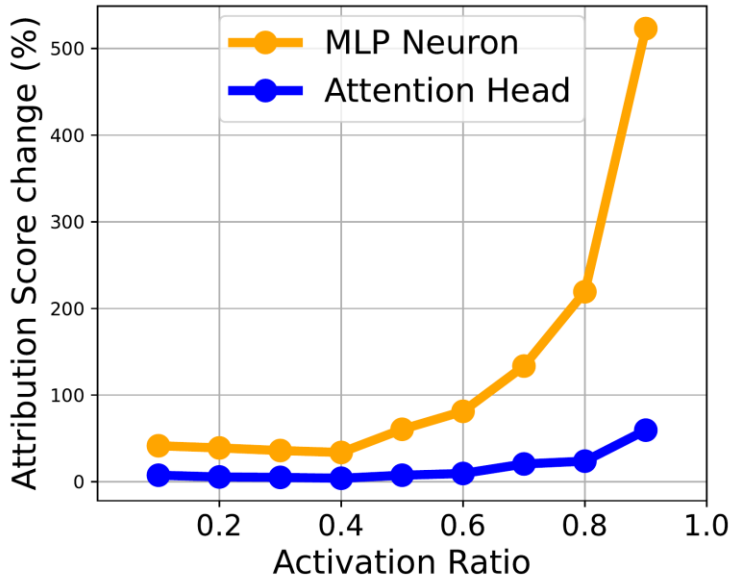
Attribution Scores Are Interdependent

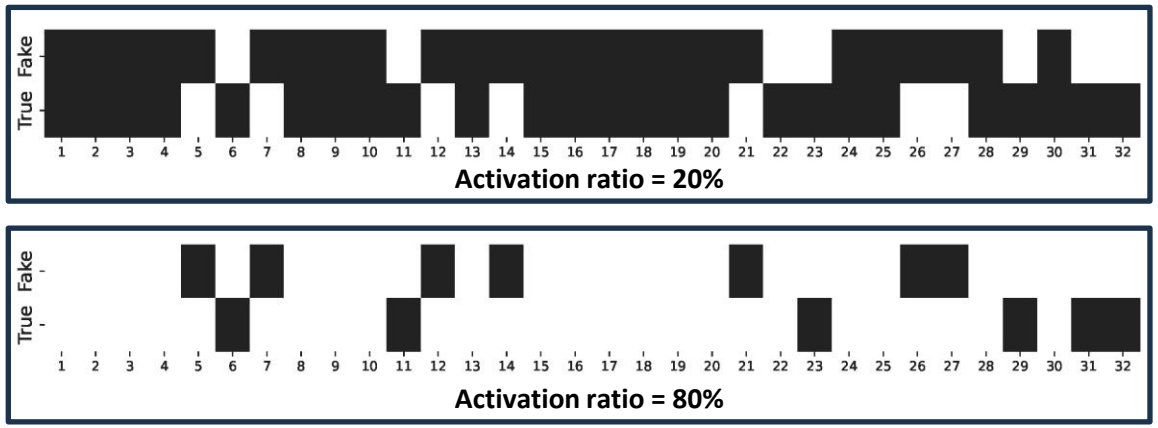
- **Deactivating one neuron**
  - → changes others' gradients and outputs
  - → changes others' attribution scores
- **Critical neurons** mislabeled as low-attribution → wrong neurons deactivated → wrong outputs



The Attribution Error Is Large

- **Grows with the activation ratio**
  - Ranking is fragile when few neurons are deactivated
- **Worse in MLP layers**
  - Thousands of neurons
  - Rankings are easily scrambled
- **Consequence**
  - Sub-optimal selection of neurons or attention heads to deactivate

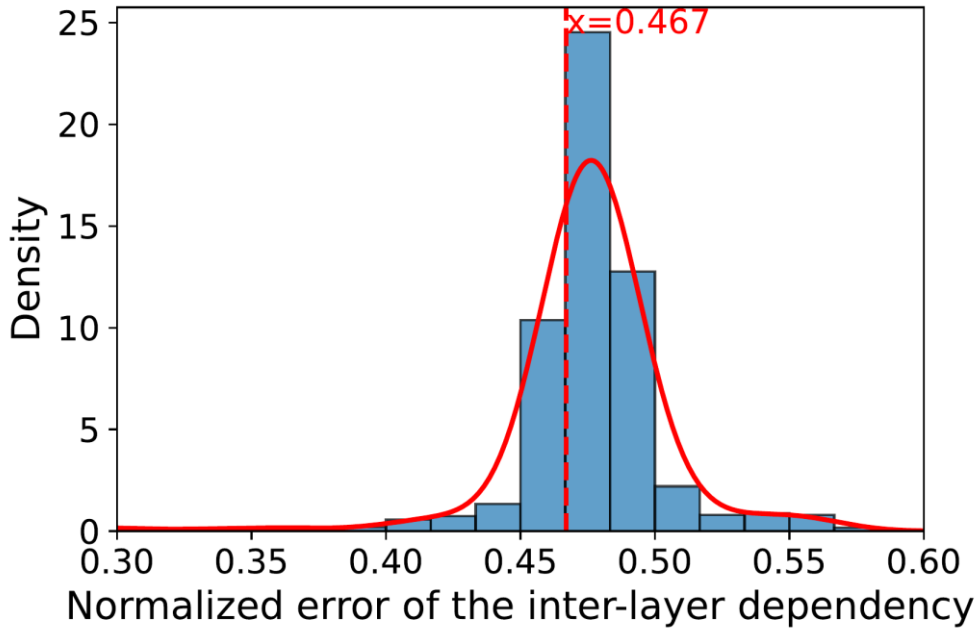




Our Fix: A Closed-Form Corrective Term

- **Naïve fix:** re-computing attribution after each deactivation → too slow

- **Instead**, we analytically bound and fix the inter-layer attribution error
  - Tight upper bound from neuron magnitudes & gradients
  - Expectation under truncated-normal distribution



$$Corrected\ \mathrm{GxO} = \underbrace{\frac{\partial F}{\partial x_i} \cdot x_i}_{\text{GxO}} + \underbrace{\frac{1}{2}}_{\text{scaling factor}} \underbrace{|x_i|}_{\text{magnitude}} \underbrace{\sqrt{\sum_{k=1}^{N_1} \left(\frac{\partial F}{\partial x_k}\right)^2}}_{\text{L2-norm of the layer's gradients}}$$

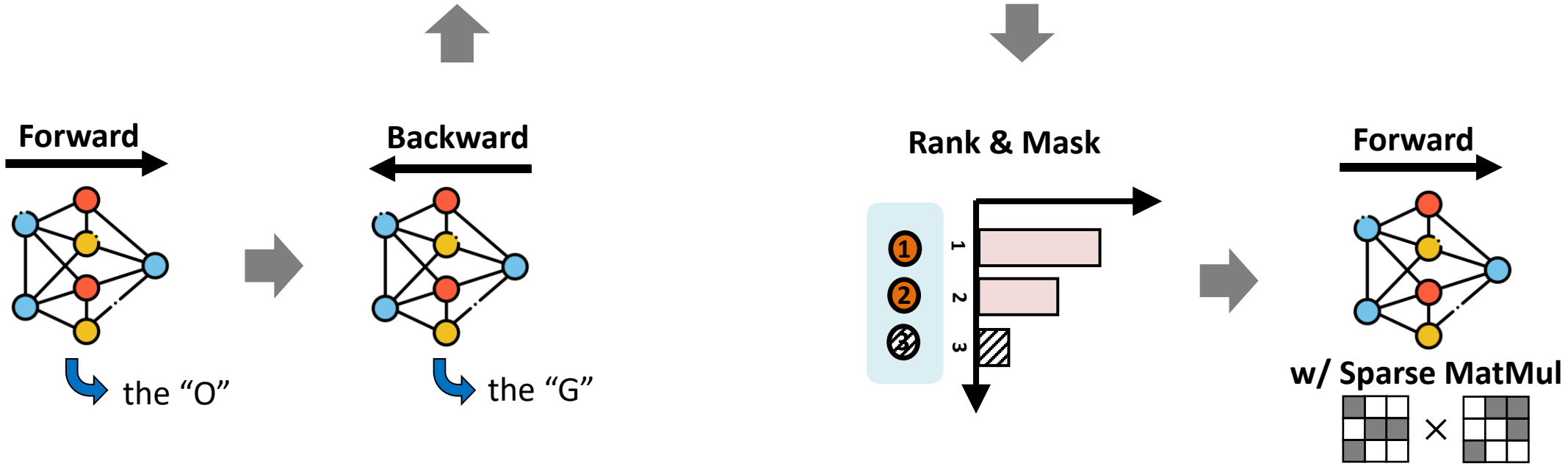
# of elements in the layer

Implementation

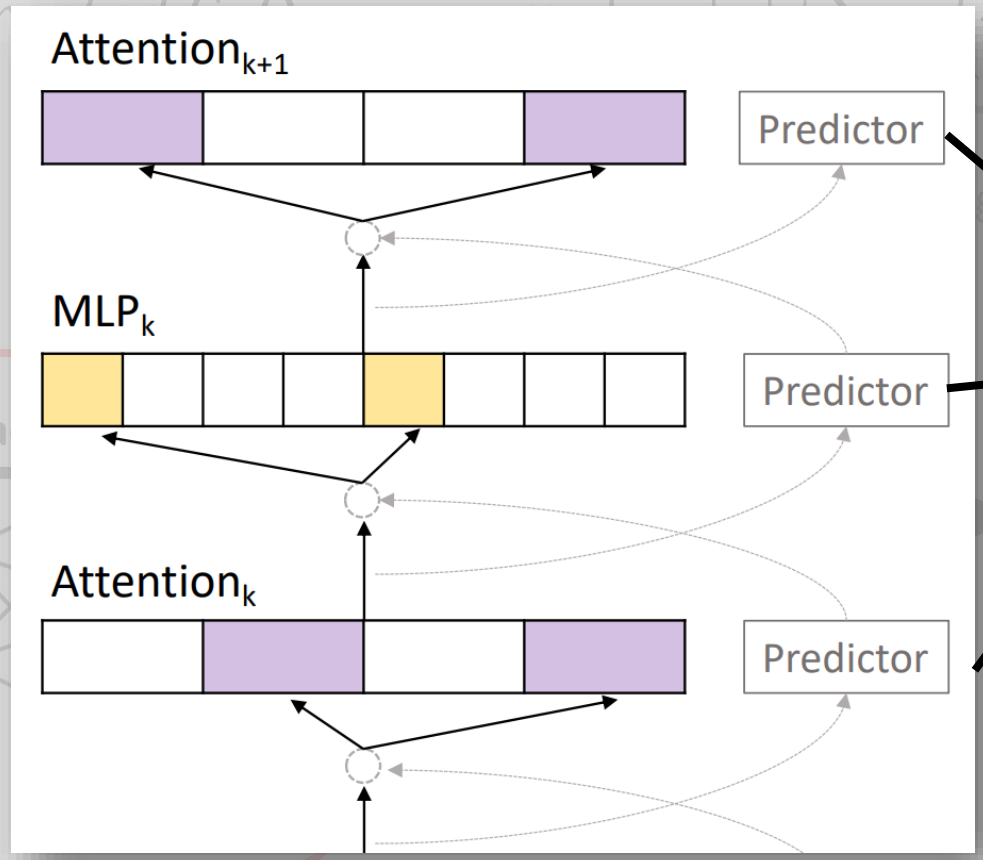
$$Corrected\ \mathrm{GxO} = \frac{\partial F}{\partial x_i} \cdot x_i + \frac{1}{2} |x_i| \sqrt{\sum_{k=1}^{N_1} \left(\frac{\partial F}{\partial x_k}\right)^2}$$

neurons' outputs

gradients



Implementation



Light-weight MLPs that predicts the activation masks

Liu, Zichang, et al. "Deja vu: Contextual sparsity for efficient llms at inference time." *International Conference on Machine Learning*. PMLR, 2023.

60–70% Sparsity and <5% Accuracy Loss on Modern LLMs

- **Benchmarks:** TruthfulQA, Gigaword, Quora QP
  - *Open-ended text-generation*, measured in BLEU and ROUGE scores
- **Models:**
  - **Llama-3-8B:** 60% of sparsity, <5% BLEU loss
  - **Phi-2 (2.7B):** 60% sparsity, <5% loss
  - **Gemma-2B, MobiLlama-0.5B:** 70% sparsity, <5% loss
- 30-40% accuracy advantage over baselines

Model & Metric	AR=10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Phi-2-2.7B										
Magnitude	10.5	19.1	21.5	23.5	25.3	24.9	24.2	24.5	24.0	33.9
Gradient	3.8	4.3	5.1	5.0	5.3	5.4	5.5	7.5	8.9	33.9
SNIP/Fisher	10.8	13.3	18.5	20.7	23.7	24.1	23.1	24.3	24.8	33.9
IG	13	16.4	18.6	22.8	23.4	22.7	23.8	25.2	30.7	33.9
GxO	15.7	19.7	19.0	21.0	22.5	23.6	22.7	24.1	31.4	33.9
Corrected GxO	**17.8**	**18.3**	**26.8**	**28.3**	**29.6**	**31.5**	**30.7**	**32.2**	**36.7**	33.9
Gemma-2B										
Magnitude	0	0.2	2.75	5.57	6.95	7.58	8.19	8.79	10.71	10.72
Gradient	0	0	0	0.37	0.61	3.04	6.6	8.14	10.03	10.72
SNIP/Fisher	0	0.11	1.86	2.47	3.06	4.41	6.77	8.07	10.32	10.72
IG	0	0	0.27	1.16	1.56	4.7	6.24	8.64	8.73	10.72
GxO	0	0.02	0.31	0.94	0.96	4.83	5.74	6.86	11.8	10.72
Corrected GxO	0	**0.55**	**5.29**	**5.63**	**8.04**	**10.2**	**10.71**	**11.58**	**13.83**	10.72
MobiLlama-0.5B										
Magnitude	0.26	1.21	1.73	2.33	2.86	2.65	3.6	3.98	5.39	5.45
Gradient	0.45	0.52	0.68	0.76	0.76	0.61	0.93	1.53	2.31	5.45
SNIP/Fisher	0.45	1.12	1.66	2.33	2.55	3.23	**4.82**	**5.13**	**5.5**	5.45
IG	0.84	1.61	1.84	1.75	1.48	0.94	1.19	1.3	1.28	5.45
GxO	0.68	1.25	1.19	1.04	1.07	0.68	0.95	1.04	1.23	5.45
Corrected GxO	**1.41**	**3.8**	**4.07**	**4.48**	**4.63**	**4.63**	4.39	4.66	5.01	5.45
Llama-3-8B										
Magnitude	1.59	2.76	6.89	11.97	13.58	15.8	18.7	16.97	14.95	26.52
Gradient	0.75	0.6	1.11	0.93	1.39	1.61	1.63	2.56	10.75	26.52
SNIP/Fisher	3.07	3.22	4.17	6.89	10.08	12.86	18.13	16.48	18.35	26.52
GxO	1.93	1.71	2.53	3.59	4.2	5	6.42	7.78	20.73	26.52
Corrected GxO	**4.6**	**7.97**	**11.84**	**21.66**	**24.48**	**26.08**	**26.94**	**27.8**	**29.3**	26.52

Table 1. Accuracy of sparsely activated LLMs with different activation ratios (AR) on TruthfulQA

35% Lower Latency and 40% Less Memory Used

- **Cold-start setup:** model loading included in latency
- **torch.sparse weights + sparse matmul**
- **At AR = 30% on Phi-2:**
  - 35% latency reduction
  - 43% GPU memory reduction

Model & Benchmark	AR=10%	30%	40%	60%	70%	80%	90%	100%	Dense
Phi-2 & TruthfulQA									
Latency (s)	0.68	1.06	1.26	1.78	2.02	2.250	2.51	2.86	1.59
Memory (GB)	4.70	7.91	9.06	12.33	15.23	18.29	21.30	24.39	13.76
Phi-2 & Gigaword									
Latency (s)	0.68	1.08	1.32	1.86	2.14	2.37	2.64	2.99	1.58
Memory (GB)	4.72	7.74	9.12	12.49	15.20	18.30	21.49	24.71	13.79
MobiLlama & TruthfulQA									
Latency (s)	0.86	1.11	1.30	1.85	2.03	2.31	2.50	2.91	1.73
Memory (GB)	3.16	5.89	7.16	10.13	12.07	13.98	16.24	18.18	10.69
MobiLlama & Gigaword									
Latency (s)	0.86	1.22	1.43	1.83	1.90	2.30	2.55	3.00	1.77
Memory (GB)	3.17	5.91	7.27	10.18	12.26	14.07	16.37	18.38	10.72
Llama-3-8B & TruthfulQA									
Latency (s)	1.16	3.53	4.30	6.09	7.39	8.29	10.44	-	6.22
Memory (GB)	7.84	19.36	24.46	34.29	39.29	42.93	48.55	-	30.67
Llama-3-8B & Gigaword									
Latency (s)	1.26	3.81	4.73	6.40	6.94	8.48	9.98	-	6.14
Memory (GB)	11.02	17.93	22.17	29.39	35.37	38.20	42.59	-	30.70

Table 5. Computing latency and memory savings with sparse activation

Generality Across Models and Tasks

- **Models:** Llama-3 · Phi-2 · Gemma · MobiLlama · Qwen2.5

- **Tasks:**
  - Question answering (TruthfulQA)
  - Summarization (Gigaword)
  - Paraphrase (Quora QP)
  - Translation (WMT16 DE→EN)
  - Natural language inference (GLUE-MNLI)

- **Best or tied-best metric in every case**

Benchmark & Metric	AR=10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Gigaword										
Magnitude	1.27	1.86	2.34	2.95	2.95	3.1	3.08	3.35	3.41	3.35
Gradient	2.45	2.48	2.45	2.41	2.41	2.46	2.4	2.58	2.47	3.35
SNIP/Fisher	0.6	1.33	1.86	2.57	3.09	3.23	3.4	3.55	3.76	3.35
IG	0.34	1.64	1.76	2.24	2.41	2.82	3.17	3.15	3.98	3.35
GxO	1.32	1.53	1.89	2.45	2.32	2.75	2.87	3.32	3.47	3.35
Cor-GxO	**2.44**	**2.42**	**2.60**	**2.74**	**2.82**	**3.14**	**4.17**	**4.04**	**4.25**	3.35
QP										
Magnitude	6.6	9.2	9.6	11.2	11.7	11	11.1	10.8	11.1	11.2
Gradient	2.76	3.38	3.6	3.63	3.67	4.85	5.21	6.32	7.85	11.2
SNIP/Fisher	6.5	8.9	10.2	10.6	11	11.1	11.8	11.8	11.3	11.2
IG	7.2	6.2	8.9	10.1	10.4	9.4	11.9	10.5	10.4	11.2
GxO	7.5	7.5	9.5	10.2	10.2	10.4	11.4	10.6	10.1	11.2
Cor-GxO	**8.3**	**9.2**	**10.8**	**10.6**	**10.7**	**11.5**	**12.4**	**12.8**	**13.4**	**11.2**

Benchmark & Metric	50%	60%	70%	80%	90%	100%
WMT16-DE-EN Metric: BLEU						
Magnitude	0.36	1.53	7.56	15.82	23.75	48.68
SNIP	0.20	1.12	2.22	6.19	13.37	48.68
GxO	0.09	1.12	2.22	6.19	13.37	48.68
Corrected GxO	**0.19**	**1.96**	**13.76**	**28.15**	**40.60**	48.68
GLUE-MNLI Metric: Accuracy						
Magnitude	0.24	0.28	0.39	0.47	0.57	0.77
SNIP	0.25	0.31	0.38	0.40	0.66	0.77
GxO	0.29	0.30	0.32	0.35	0.41	0.77
Corrected GxO	**0.35**	**0.42**	**0.55**	**0.67**	**0.77**	0.77

Takeaways

- Magnitude-based sparse activation does not apply on modern LLMs
- Attribution is the right signal, but errors caused by interdependency need to be fixed
- We introduced a closed-form corrective term on the attribution metric
- Achievements on modern LLM families
  - 70% sparsity with <5% accuracy loss
  - 35–40% latency reduction & memory savings

Thank you!

- Questions?

Intelligent System Laboratory @ Pitt

http://pittisl.github.io