ReLU Strikes Back:
Exploiting Activation Sparsity in Large Language Models

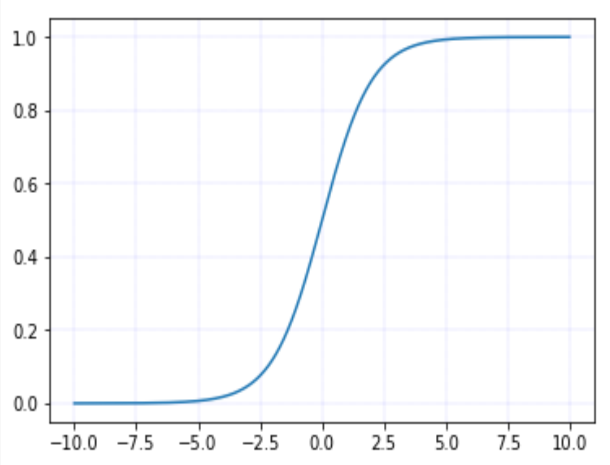
Iman Mirzadeh[†] **Keivan Alizadeh** **Sachin Mehta** **Carlo C Del Mundo**

Oncel Tuzel **Golnoosh Samei** **Mohammad Rastegari** **Mehrdad Farajtabar**[†]

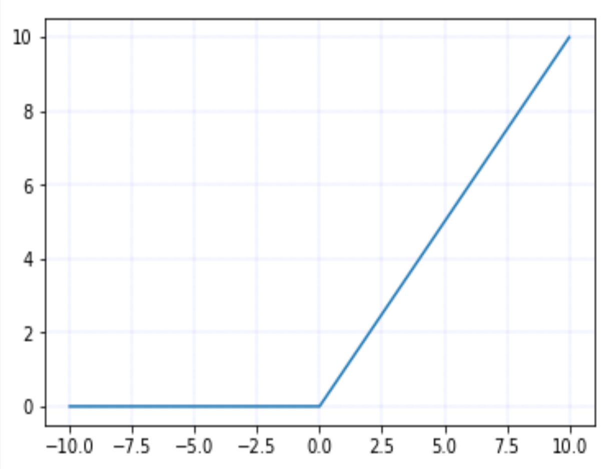
Apple

ICLR 2024

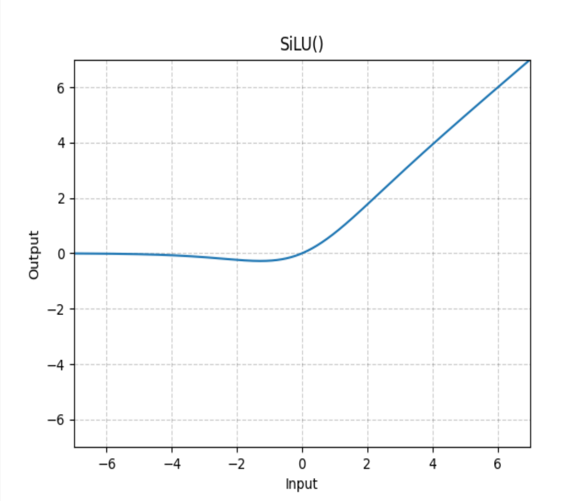
Ming Wang
2024.02.06



Sigmoid



ReLU



SiLU

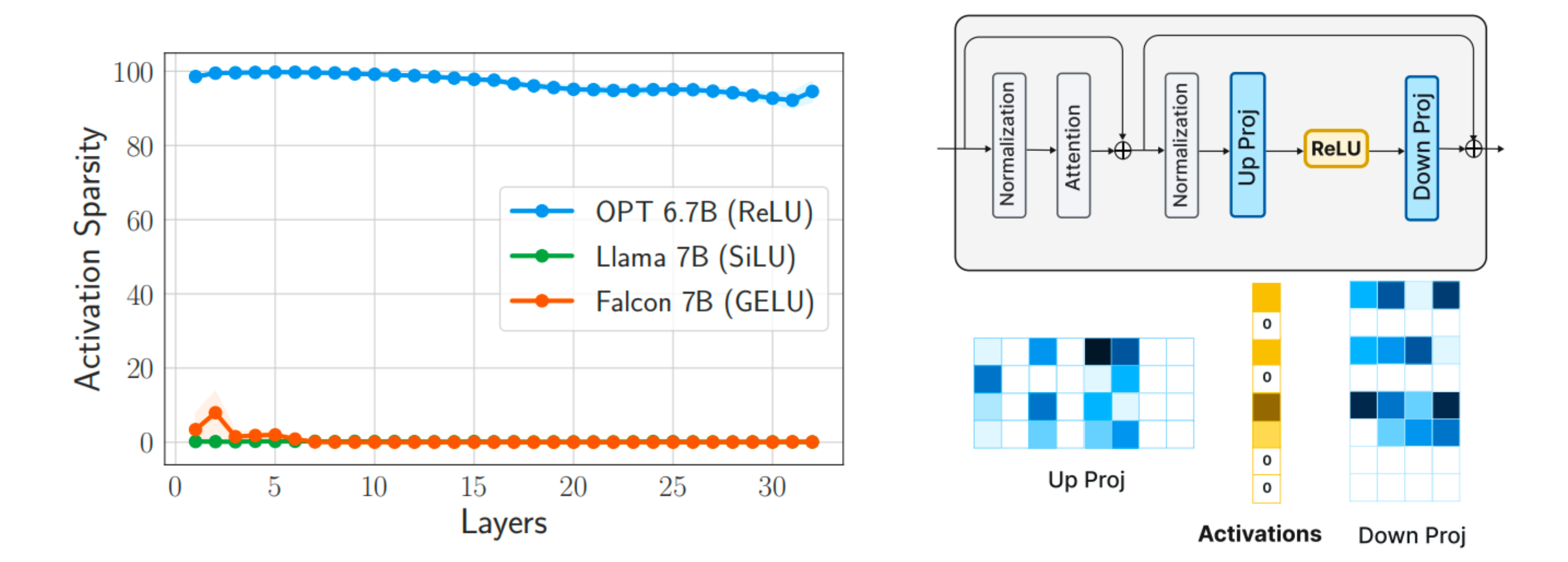
All these activation functions can be viewed as $f(x) = x \cdot \sigma(\beta x)$, where $\sigma(x) = \frac{1}{1+e^{-x}}$

SiLU can be viewed as $\beta = 1$, GeLU can be viewed as $\beta = 1.7$, and ReLU can be viewed as $\beta = \infty$

Does the activation function impact performance?

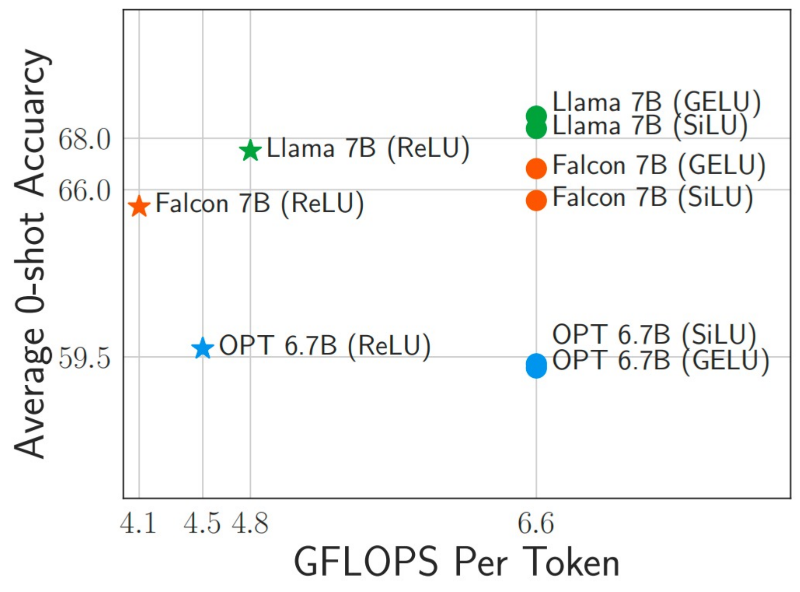
This Paper re-evaluate using ReLU for LLMs.

- Methods to enhance inference efficiency: quantization, pruning, speculative decoding, weight sparsification.
- Activation sparsity results in substantial weight transfer (I/O) savings between the GPU and CPU.
- In OPT-6.7B, sparsity can impact 95% of the rows of the down projection layer’ s weights. And activation sparsity is more hardware-friendly than unstructured pruning.



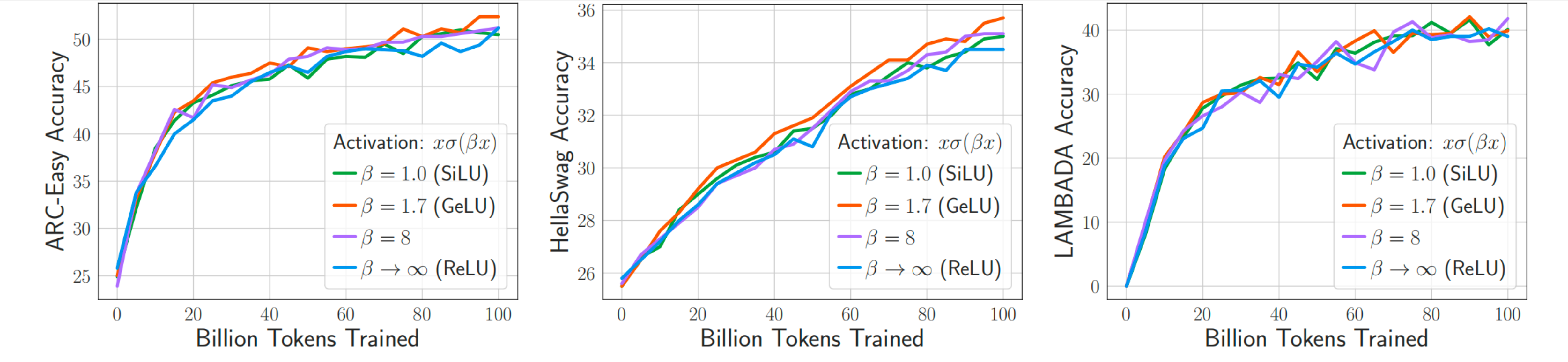
(a) Sparsity of different models (b) Sparsity for Efficiency

- When trained from scratch, there is no significant difference in terms of performance between different activation functions, but ReLU requires much less computation.
- Models quickly regain performance when fine-tuning from existing activation functions. Inserting additional ReLU layers after normalization layers, we can further reduce inference FLOPS

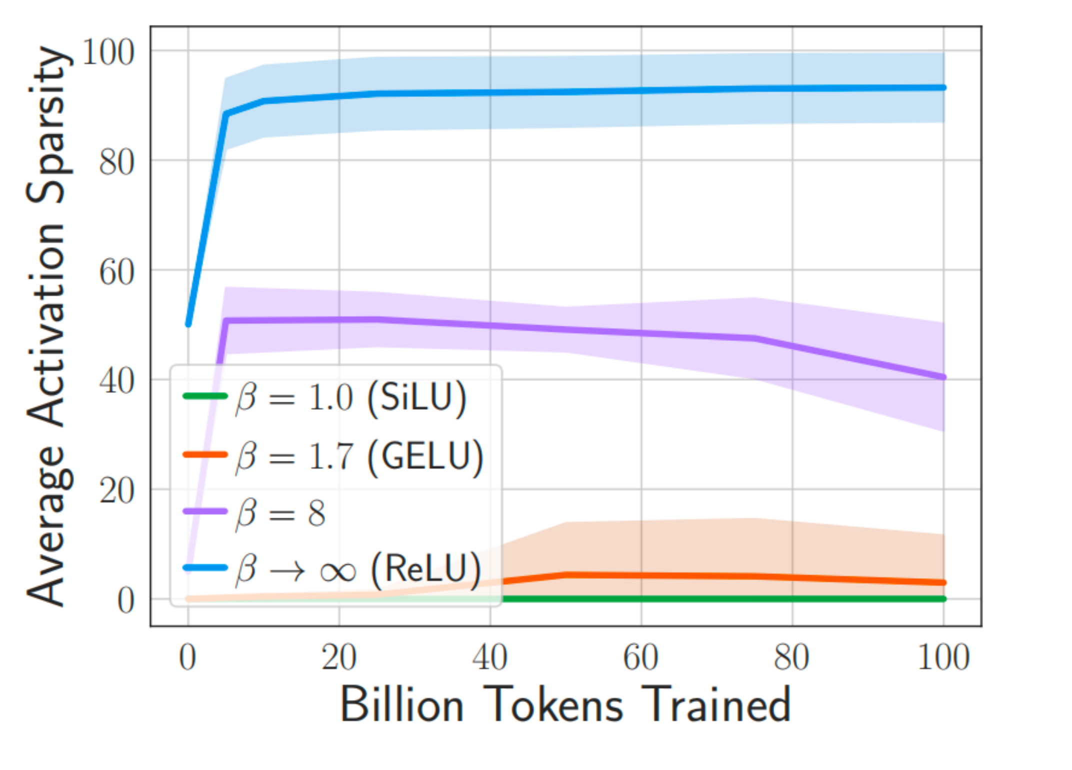


(c) Accuracy vs. Computation

- Train OPT-1.3B model from scratch with different activation function on a hundred billion tokens of the RefinedWeb datasets.
- The performance of the models is very similar when using different activation functions.

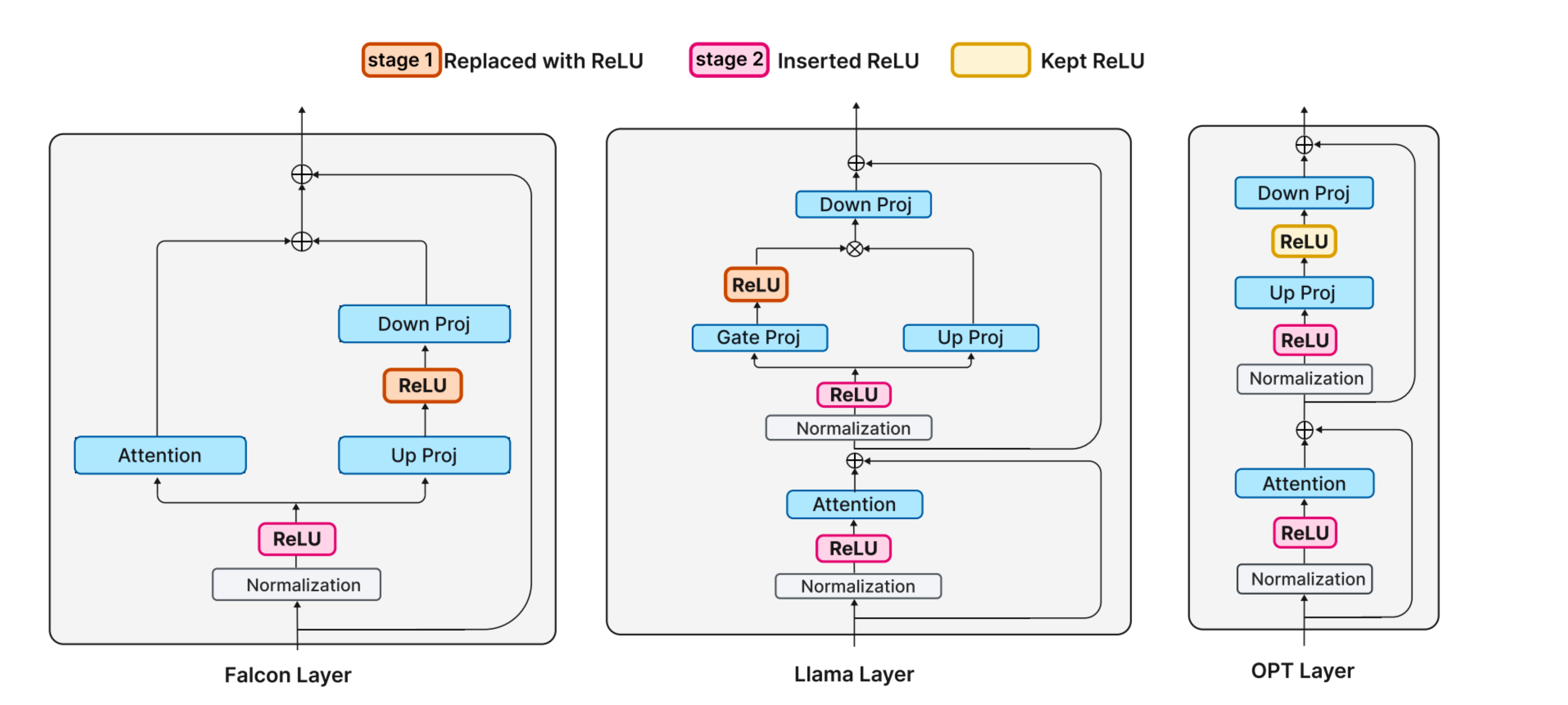


- non-ReLU activations result in a negligible performance gain (if any) but a substantial loss in sparsity and efficiency



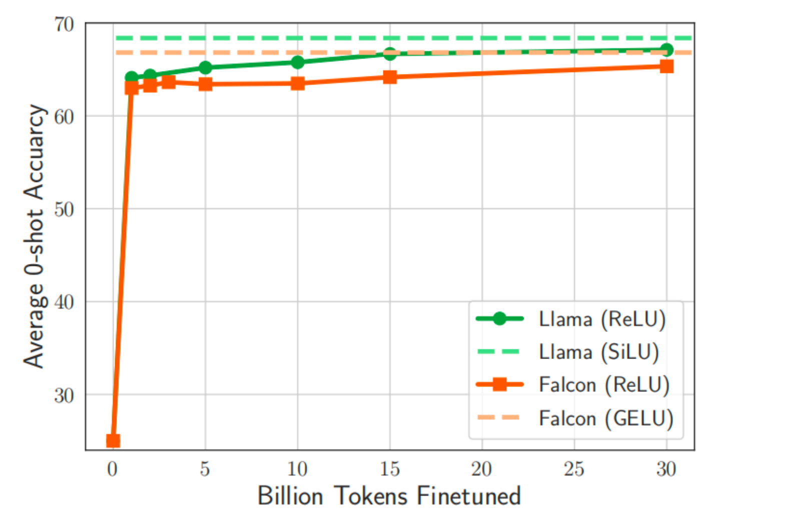
Relufication includes two stages:

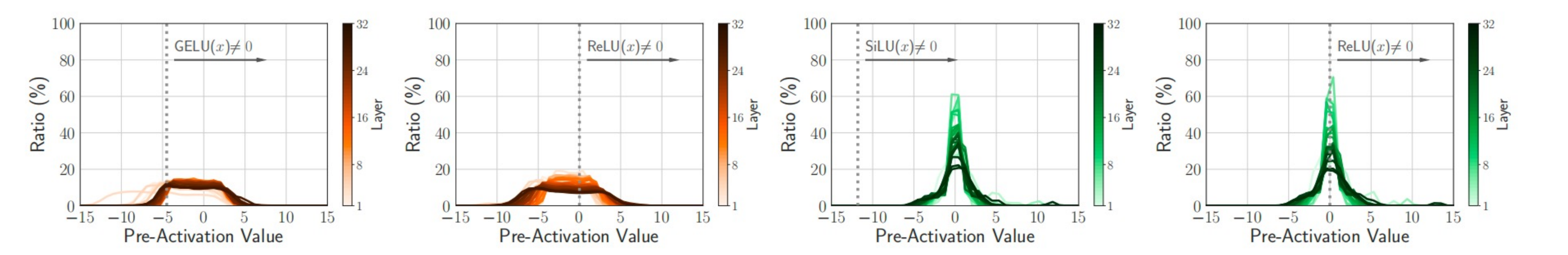
- Stage1: replace the activation function between up_proj and down_proj with ReLU.
- Stage2: insert new ReLUs after normalization layers.



Stage1:

- The model performance quickly recovers during relatively short fine-tuning stage.
- The distribution of pre-activation doesn' t change significantly.





(a) Falcon 7B (GELU) (b) Falcon 7B (ReLU) (c) Llama 7B (SiLU) (d) Llama 7B (ReLU)

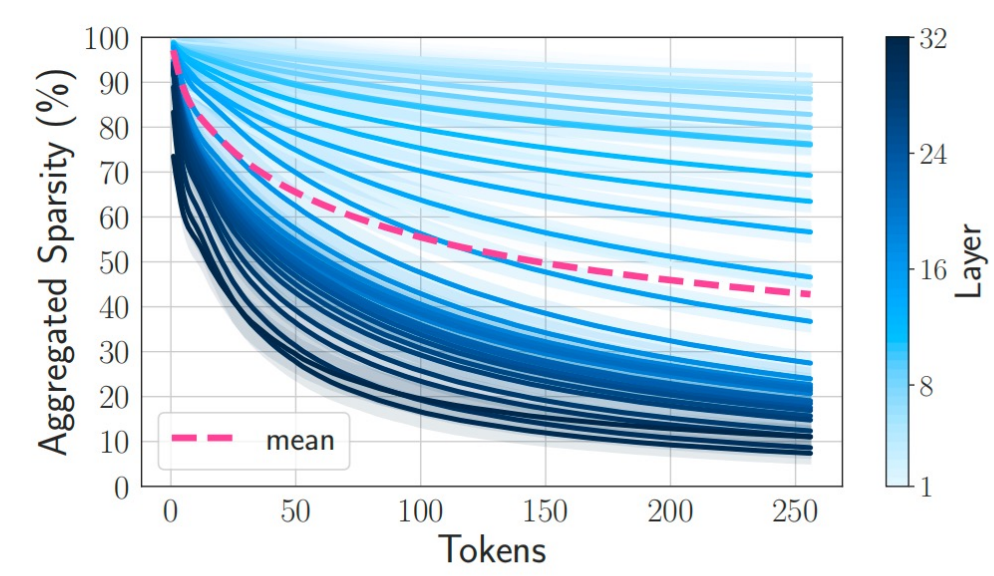
Stage2:

- While stage1 leads to the input of down_proj being sparse, QKV and up_proj remains.
- Inserting new ReLUs after normalization layers brings more sparsity.

Model (stage)	Input Sparsity (%)			FLOPS (G)	Zero-Shot Accuracy (%)									
	QKV	DownProj	UpProj		Avg	Arc-E	Arc-C	Hellaswag	BoolQ	PIQA	LAMBADA	TriviaQA	WinoGrande	SciQ
OPT 1.3B	0	96	0	1.3	50.7	57.3	22.9	41.3	57.0	71.8	56.0	6.1	58.9	84.6
OPT 2.7B (s2)	50	96	35	1.1	53.1	60.3	26.8	44.9	55.4	73.9	57.6	12.4	59.6	86.7
OPT 2.7B	0	96	0	1.8	54.5	63.3	29.2	45.8	57.6	74.2	61.4	12.3	60.8	85.9
OPT 6.7B (s2)	50	97	40	2.8	58.6	66.5	32.2	49.1	63.0	76.4	63.3	23.8	63.1	90.3
OPT 6.7B	0	97	0	4.5	59.8	68.0	32.4	50.2	68.4	75.8	67.2	20.9	65.3	90.2
Falcon 7B (s2)	56	95	56	2.2	64.8	73.6	38.6	55.3	68.4	78.9	67.6	40.4	67.1	93.4
Falcon 7B (s1)	0	94	0	4.1	65.2	72.2	39.1	55.4	70.6	78.4	69.2	40.5	67.5	93.1
Falcon 7B	0	1	0	6.6	66.8	74.6	40.2	57.7	73.5	79.4	74.5	40.4	67.2	94.0
Llama 7B (s2)	51	65	67	2.9	66.4	73.8	39.6	54.8	69.9	77.9	70.7	48.5	68.6	93.8
Llama 7B (s1)	0	62	0	4.8	67.1	75.2	40.1	55.2	73.4	77.7	71.5	49.6	67.1	94.2
Llama 7B	0	0	0	6.6	68.4	75.5	42.1	69.9	74.8	78.7	73.1	49.9	69.8	95.4

➢ Aggregated Sparsity

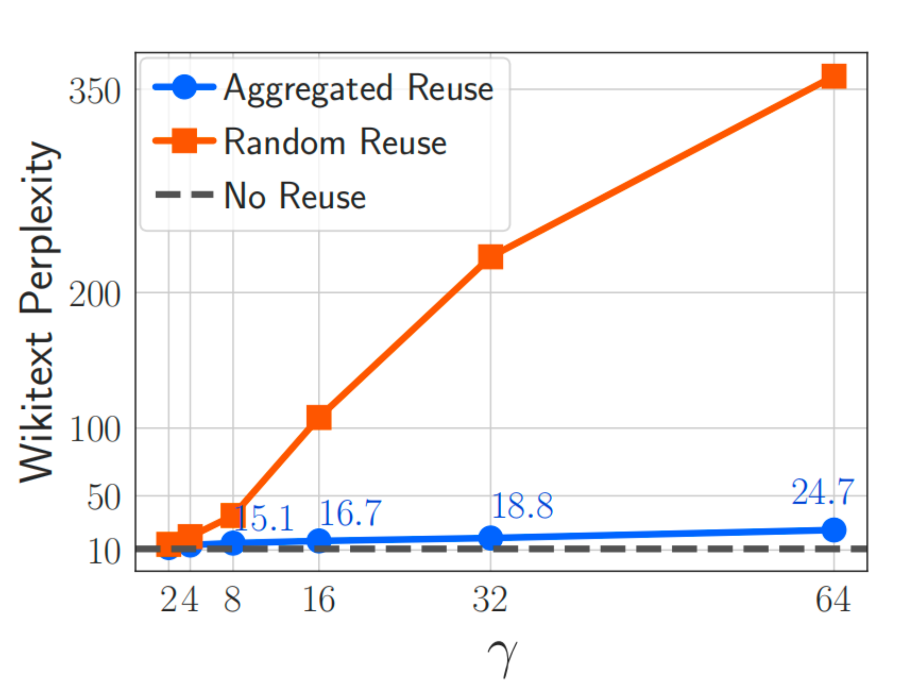
- Define **Aggregated Sparsity** as the ratio of neurons that have not been used up to processing the first *t* token
- We can benefit from the overlapping activations by utilizing previously loaded weights from the down projection layer for upcoming tokens.



OPT-6.7B

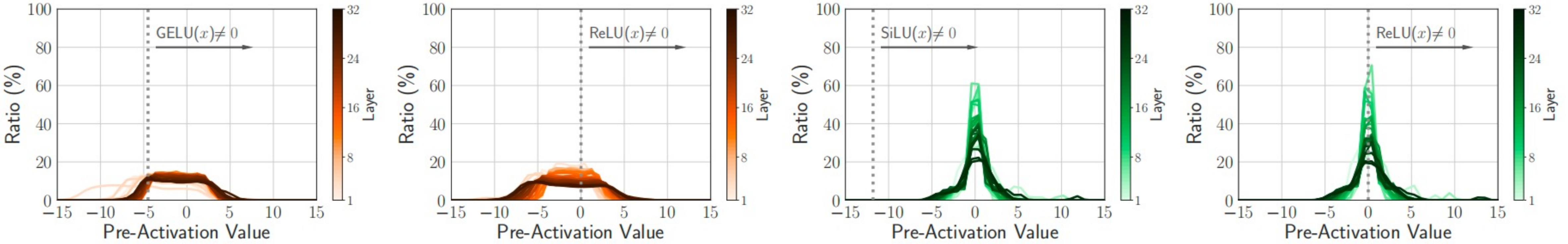
➢ Aggregated Sparsity

- Utilizing aggregated sparsity, we can intermittently avoid loading new weights for every γ token.
- Using $\gamma = 16$ as an example, tokens 129-145 are generated conventionally. However, for tokens 146-161, we retain the existing weight without introducing any new weight.



Shifted ReLU

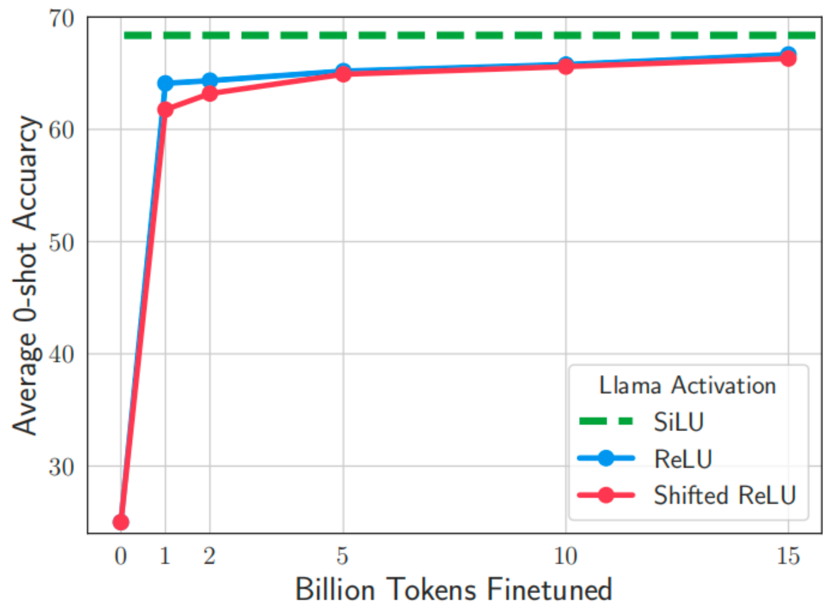
- Compared to Falcon, relufied Llama has much less sparsity(65%). We may be able to shift the pre-activation distribution to the left to put more volume before the cutoff at 0.
- The ReLU we used will be $\mathrm{Re}LU(x) \rightarrow \mathrm{Re}LU(x-b)$

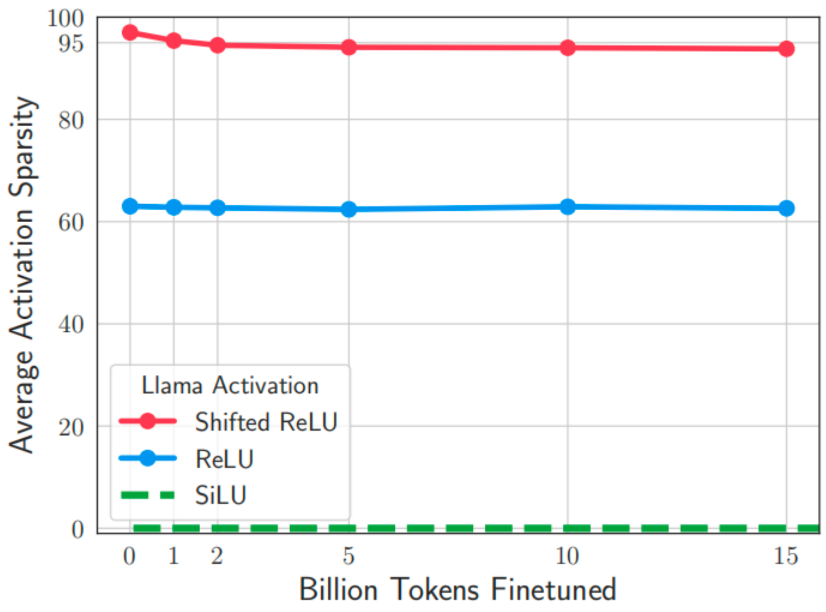


(a) Falcon 7B (GELU) (b) Falcon 7B (ReLU) (c) Llama 7B (SiLU) (d) Llama 7B (ReLU)

➢ Shifted ReLU

- For Llama, set b = 1 can attain more sparsity while maintaining on-par accuracy with the ReLU activation function.





哈爾濱工業大學

HARBIN INSTITUTE OF TECHNOLOGY

Thanks