Mixture of Scales: Memory-Efficient Token-Adaptive Binarization for Large Language Models

Dongwon Jo[1], Taesu Kim[2], Yulhwa Kim[3]*, Jae-Joon Kim[3]*

The Thirty-Eighth Annual Conference on Neural Information Processing Systems (NeurIPS), 2024.

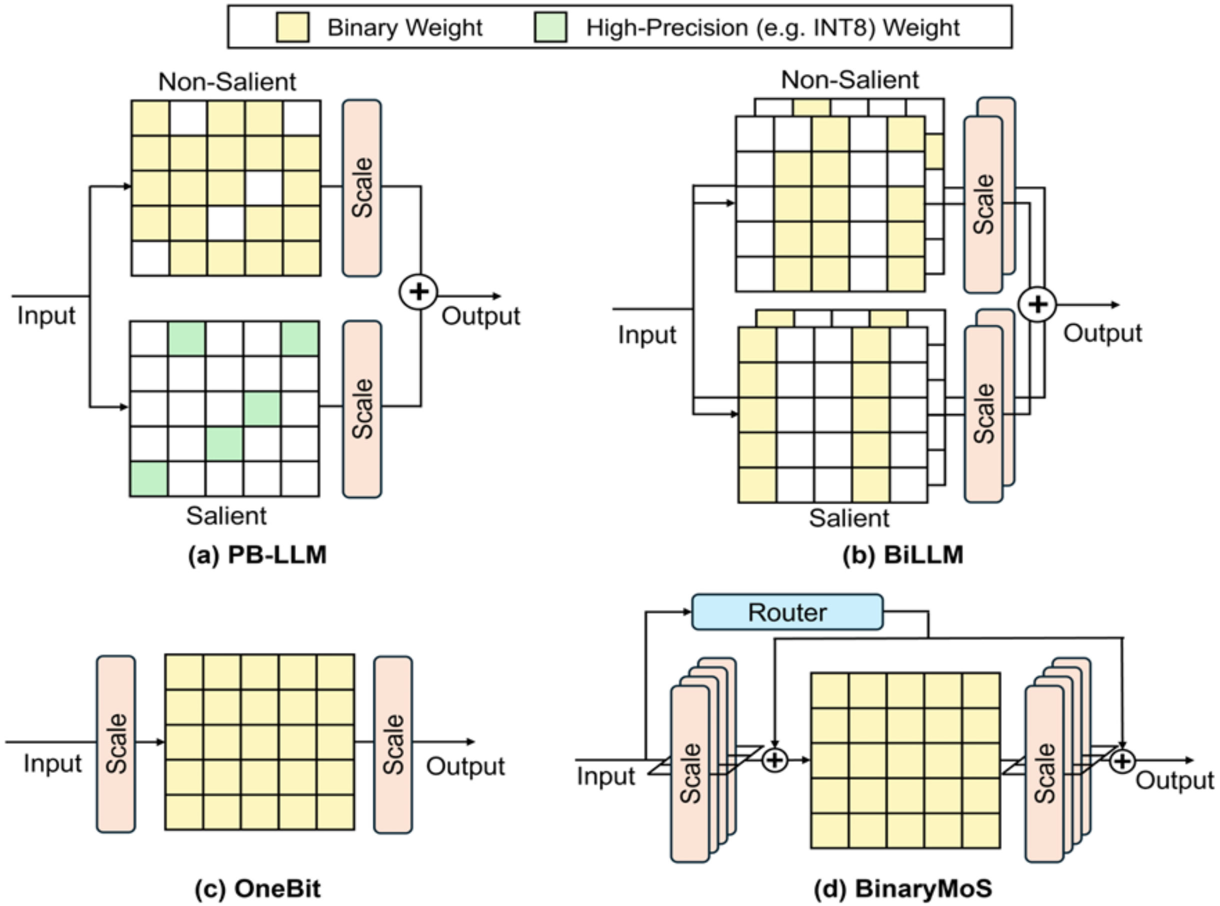
[1]Seoul National University, [2]SqueezeBits Inc., [3]Sungkyunkwan University, [*]Corresponding Author

LLM Binarization

- Binarization is extreme version of quantization which transforms high-precision weight parameters into 1-bit
- Binarization is effective strategy to reduce the size of LLMs, but, typical binarization techniques show significant performance degradation
- Previous binarization using QAT or PTQ drastically limits the representational capacity of weights, struggling to achieve sufficient accuracy with binarized LLMs
- Previous works effort often compromise the inherent advantages of binarization by introducing high memory overhead
- Training binarized model from scratch requires high training cost and many GPU resources

Previous Binarization Methods

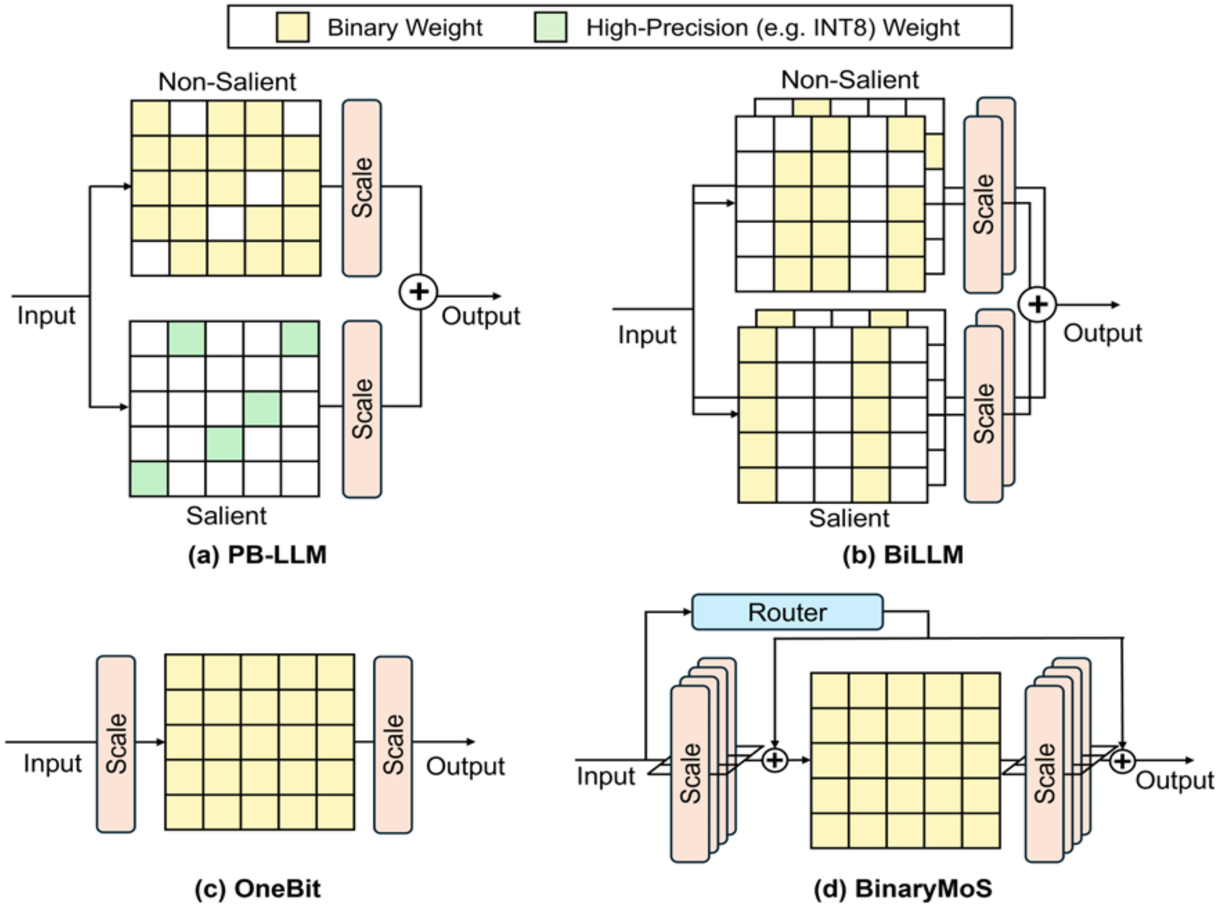
- **PB-LLM**
  - They maintain salient weight parameters as high-precision values (e.g., Float16 or INT8)
  - Index of salient weights is unstructured, requiring mask and indexing information
- **BiLLM**
  - They use the additional matrix as the residual matrix for salient weights
  - For non-salient weight, they categorize weight: concentrated weights close to the mean, and sparse weights



Zhihang Yuan, et al., "PB-LLM: Partially Binarized Large Language Models." ICLR, 2024.
Wei Huang, et al., "BiLLM: Pushing the Limit of Post-Training Quantization for LLMs." ICML, 2024.

Previous Binarization Methods (cont'd)

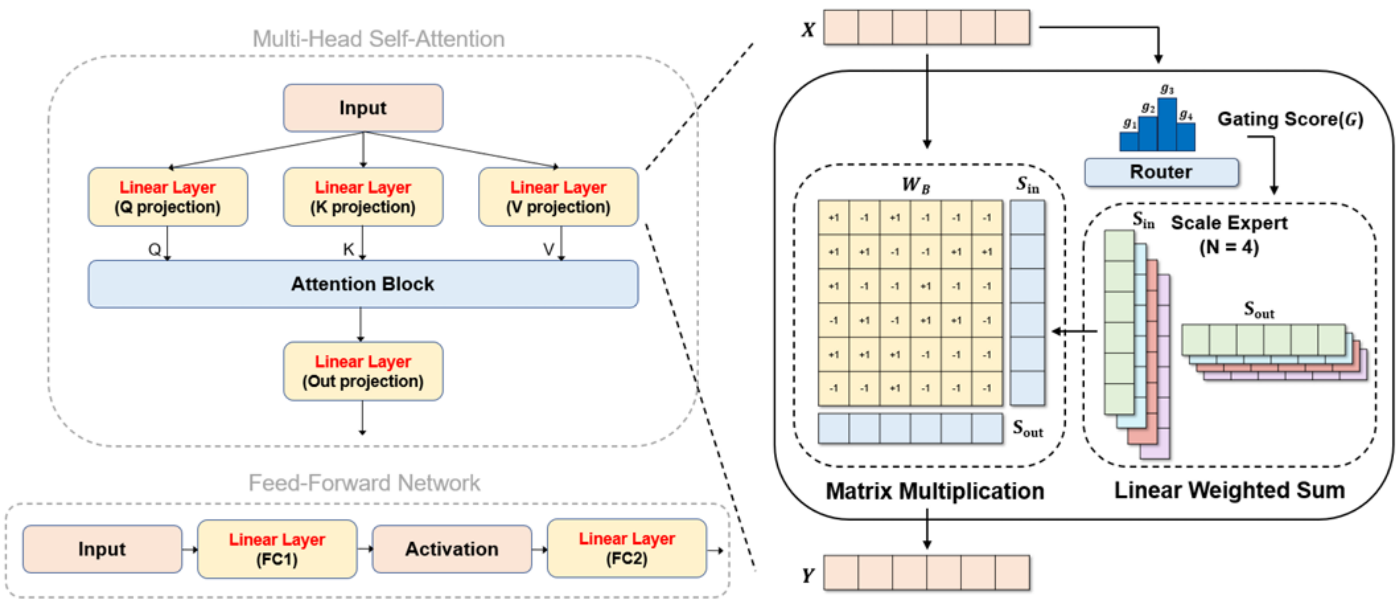
- **OneBit**
  - They incorporate scaling factors for both the input and output channel
  - They initialize scaling factor, decompose weights into rank-1 by SVD method
- **Limitation of Previous Works**
  - Not considering acceleration on HW such as GPUs in real scenario
  - Compromising the advantages of binarization by introducing high memory overhead
  - Low representational power, struggling to achieve sufficient accuracy compared to FP16



➡ We propose a novel binarization technique, **Mixture of Scales (BinaryMoS)**

Yuzhuang Xu, et al., "OneBit: Towards Extremely Low-bit Large Language Models." NeurIPS, 2024.

BinaryMoS: Mixture of Scales for Binarization



Gating Score

$G = \mathrm{Softmax}(XW_R)$

$W_R \in \mathbb{R}^{m \times e}$ (Router Weight)

Token-Adaptive Scaling Factor

$\hat{S}_{in} = GS_{in}, \quad \hat{S}_{out} = GS_{out}$

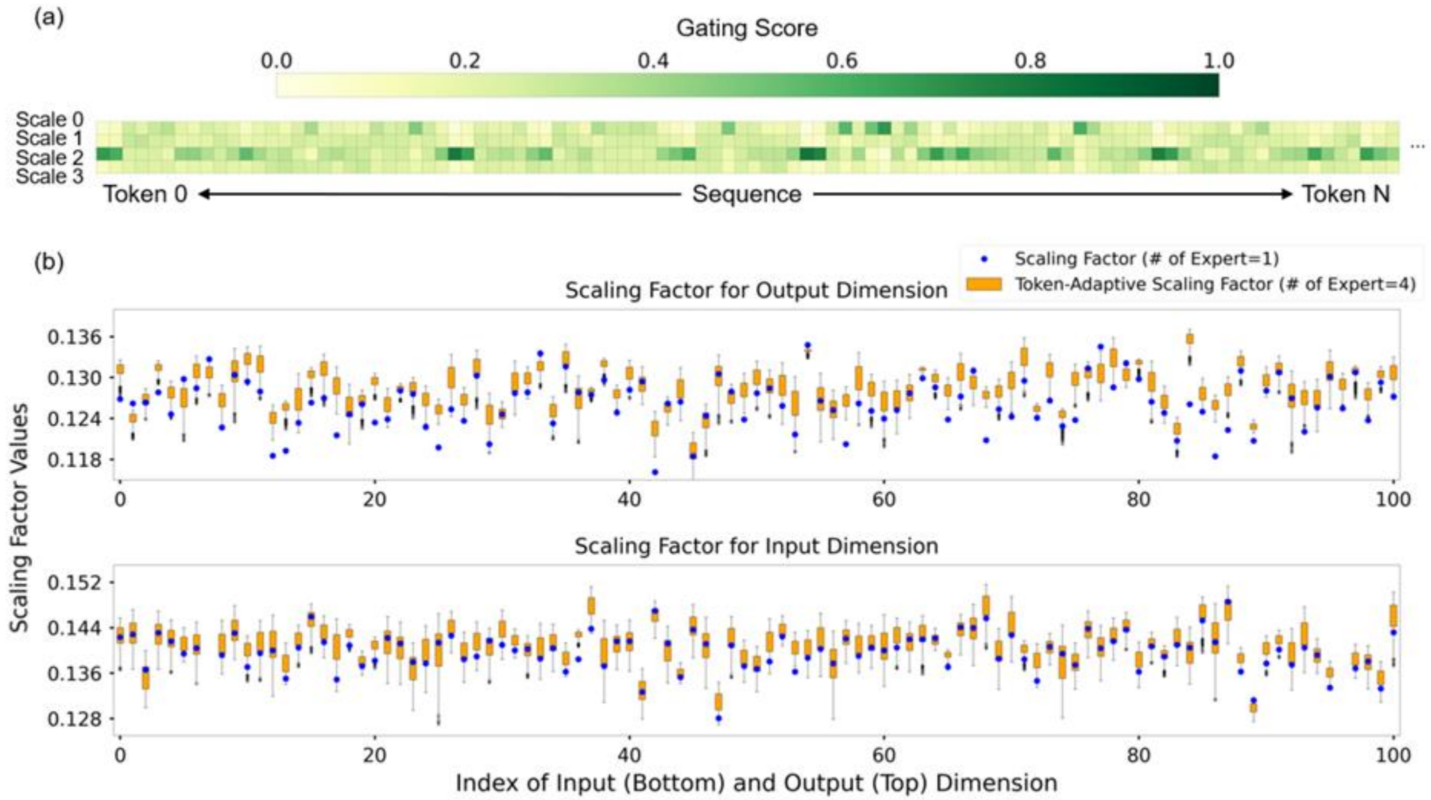
$S_{in} \in \mathbb{R}^{e \times m}$, $S_{out} \in \mathbb{R}^{e \times n}$ (Scaling Experts)

Matrix Multiplication

$\hat{Y} = [(X \odot \hat{S}_{in})\mathrm{Sign}(W_{FP}^T)] \odot \hat{S}_{out}$

- Router computes the **gating score G**, which represents the significance of each scaling expert, using input tokens and router weights
- These gating scores are used to linearly combine the scaling experts, resulting in the creation of **token-adaptive scaling factors, $\hat{S}_{in}$ and $\hat{S}_{out}$**
- This Input-dependent method makes token-adaptive scaling factor dynamic scale, **increasing representation power with minimal memory and latency overhead**

Representational Power of Token Adaptive Scaling Factors



Analysis on Token-Adaptive Scaling Factors

- Router assigns gating score with substantial variation for each expert across token
- While conventional binarization methods with static scaling factors, offer a fixed scaling factor, the BinaryMoS successfully **generates a diverse range of scaling factors**

Experimental Results

		Perplexity ↓ (Wikitext2)					
Method	Wbits	OPT-125M	OPT-1.3B	LLaMA-1-7B	LLaMA-1-13B	LLaMA-2-7B	LLaMA-2-13B
GPTQ	2	660.52	125.29	45.73	15.20	40.23	32.87
OmniQuant	2	245.47	28.82	9.75	7.84	11.20	8.25
BinaryMoS	1	**36.46**	**18.45**	**7.97**	**7.16**	**7.88**	**7.08**
		Perplexity ↓ (C4)					
Method	Wbits	OPT-125M	OPT-1.3B	LLaMA-1-7B	LLaMA-1-13B	LLaMA-2-7B	LLaMA-2-13B
GPTQ	2	213.60	45.43	27.87	15.15	31.37	26.23
OmniQuant	2	390.30	33.81	13.01	10.43	15.46	11.06
BinaryMoS	1	**33.13**	**18.83**	**9.72**	**8.81**	**9.75**	**8.91**
		Average Zero-shot Accuracy ↑					
Method	Wbits	OPT-125M	OPT-1.3B	LLaMA-1-7B	LLaMA-1-13B	LLaMA-2-7B	LLaMA-2-13B
GPTQ	2	37.59	40.36	43.75	49.65	43.31	45.03
OmniQuant	2	36.54	46.43	51.58	56.42	49.54	54.24
BinaryMoS	1	**43.37**	**49.34**	**54.48**	**56.68**	**54.01**	**57.09**

Comparison to 2-bit Quantization Methods

- BinaryMoS **consistently outperforms other binarization methods** and narrows the performance gap with Float16 model
- BinaryMoS even **outperforms 2-bit quantization methods,** despite its lower memory requirement during inference

Model	Method	Wbits	Perplexity ↓		Zero-shot Accuracy ↑						
			Wiki2	C4	BoolQ	PIQA	Hella.	WinoG.	ARC-e	ARC-c	Average
OPT-125M	Float16	16	27.65	24.60	55.47	62.02	31.33	50.19	39.98	22.86	43.64
	PB-LLM	1	3233.63	1509.33	37.83	50.60	26.67	50.43	27.02	23.63	36.02
	BiLLM	1	2989.53	1769.26	37.82	50.59	25.75	51.30	27.65	23.63	36.12
	OneBit	1	39.45	35.58	61.92	60.01	27.01	50.43	35.81	21.84	42.84
	BinaryMoS	1	**36.46**	**33.13**	61.83	60.17	27.16	51.38	36.74	22.95	**43.37**
OPT-1.3B	Float16	16	14.62	14.72	57.82	72.42	53.70	59.51	50.97	29.52	53.99
	PB-LLM	1	272.83	175.42	62.17	54.24	27.25	50.27	27.98	23.72	40.94
	BiLLM	1	69.45	63.92	61.92	59.52	33.81	49.32	34.38	22.35	43.55
	OneBit	1	20.36	20.76	57.85	66.53	39.21	54.61	42.80	23.97	47.50
	BinaryMoS	1	**18.45**	**18.83**	60.34	68.66	41.99	53.99	44.87	26.19	**49.34**
LLaMA-1-7B	Float16	16	5.68	7.08	73.21	77.42	72.99	66.85	52.53	41.38	64.06
	PB-LLM	1	198.37	157.35	60.51	53.53	27.23	49.17	27.48	26.02	40.66
	BiLLM	1	41.66	48.15	62.23	58.65	34.64	51.14	33.08	25.68	44.24
	OneBit	1	8.48	10.49	62.50	70.40	54.03	55.32	41.07	30.88	52.36
	BinaryMoS	1	**7.97**	**9.72**	64.59	71.82	58.18	58.88	42.09	31.31	**54.48**
LLaMA-1-13B	Float16	16	5.09	6.61	68.47	79.05	76.24	70.17	59.85	44.54	66.39
	PB-LLM	1	35.83	39.79	62.17	58.70	33.97	52.17	31.86	23.63	43.75
	BiLLM	1	14.56	16.67	62.53	68.17	52.24	59.43	41.91	29.94	52.37
	OneBit	1	7.65	9.56	63.30	71.98	60.61	59.43	42.85	32.42	55.10
	BinaryMoS	1	**7.16**	**8.81**	63.82	73.88	64.05	60.93	44.28	33.11	**56.68**
LLaMA-2-7B	Float16	16	5.47	6.97	71.07	76.87	72.95	67.16	53.45	40.78	63.71
	PB-LLM	1	76.75	85.92	62.17	52.82	26.87	50.11	26.89	24.31	40.53
	BiLLM	1	27.72	36.34	62.14	59.19	35.18	53.11	34.22	26.54	45.06
	OneBit	1	8.60	10.74	63.06	70.40	54.24	56.67	40.82	29.35	52.42
	BinaryMoS	1	**7.88**	**9.75**	65.02	71.55	59.41	56.18	41.84	30.03	**54.01**
LLaMA-2-13B	Float16	16	4.88	6.47	68.99	79.05	76.62	69.77	57.95	44.20	66.10
	PB-LLM	1	155.25	151.15	37.82	53.26	28.89	49.48	28.28	23.72	36.91
	BiLLM	1	20.71	27.19	62.20	62.51	38.05	56.35	40.69	27.73	47.92
	OneBit	1	7.56	9.67	65.66	71.60	60.07	56.91	45.76	31.74	55.29
	BinaryMoS	1	**7.08**	**8.91**	66.12	73.72	63.80	58.98	45.71	33.19	**57.09**

Comparison to Other Binarization Methods

Memory Efficiency and Latency

Model	Float16	PB-LLM	BiLLM	OneBit	BinaryMoS
LLaMA-1/2-7B	13.51 GB	2.78 GB (4.86×)	2.28 GB (5.93×)	1.37 GB (9.86×)	1.40 GB (9.65×)
LLaMA-1/2-13B	26.20 GB	5.02 GB (5.22×)	4.06 GB (6.45×)	2.29 GB (11.44×)	2.33 GB (11.24×)

Comparison of Memory Footprint

- BinaryMoS significantly reduces the memory footprint of models, achieving compression ratios ranging from **9.65× to 11.24× with minimal memory overhead**
- Despite incorporating additional components for scaling experts, BinaryMoS increases by **only 2% compared to OneBit**

Model Config	**LLaMA-1/2-7B**			**LLaMA-1/2-13B**		
Weight Size	**4096 × 4096**	**4096 × 11008**	**11008 × 4096**	**5120 × 5120**	**5120 × 13824**	**13824 × 5120**
Float16	68.2	151.7	143.5	95.6	224.1	213.6
PB-LLM	96.1	177.5	168.3	122.7	243.7	234.7
BiLLM	87.1	96.4	104.2	95.2	124.2	131.0
OneBit	32.7	33.7	34.9	33.4	41.4	42.6
BinaryMoS	34.5	36.9	37.0	35.6	43.4	44.5

Latency (μsec) of Linear Layer

- BinaryMoS reduces latency compared to Float16 models by up to **5.2x**
- This demonstrates that the BinaryMoS **improves performance** in terms of perplexity and accuracy with **minimal latency overhead**

Summary

- BinaryMoS is a novel binarization technique designed to **enhance the representation capability** of binarized LLMs while **preserving the fundamental advantage of binarization**

- BinaryMoS adopts the mixture of scales approach to **dynamically adjust the scaling factors** of binary weight values in a **token-adaptive manner**

- This approach effectively mitigates information loss associated with binarization with **minimal memory and latency overhead**

- Our experimental results demonstrate that BinaryMoS **surpasses existing binarization approaches** and even **outperforms 2-bit quantization methods** in both perplexity and zero-shot tasks