HBLLM: Wavelet-Enhanced High-Fidelity 1-Bit Quantization for LLMs

Ningning Chen
Sun Yat-sen University
chennn27@mail2.sysu.edu.cn

Weicai Ye
Sun Yat-sen University
cai_rcy@163.com

Ying Jiang
Sun Yat-sen University
jiangy32@mail.sysu.edu.cn

Content

Introduction

Why to propose 1-bit PTQ methods for LLMs?

- Large Language Models (**LLMs**) are difficult to deploy due to their memory size.
- Post-Training Quantization **(PTQ)** reduce memory size by compressing LLMs without additional training
- 1-bit PTQ methods quantize weights of LLMs into 1 bit. Compare to 16-bit models,
  - potential 75~90% memory of weights saving
  - potential faster inference speed
- BiLLM is the first 1-bit PTQ method without knowledge distillation.

Why not to use 1-bit PTQ methods for LLMs?

- ✗ large reconstruction error
- ✗ loss of critical information
- ✗ difficulty in adaptation to heterogeneous model structures

Introduction

HBLLM

- A 1-bit PTQ weight-only framework we propose.
- Integrating localized orthogonal transformations (i.e., Haar wavelets) into a BiLLM-style quantization process with other enhancements (Table 1).

Weaknessees in 1-bit PTQ methods	**Sources**	**Enhancements used in HBLLM**
Large reconstruction error	Limited expressiveness	**Quantization in Haar domain**
Loss of critical information	Inaccurate salient-column selection	**ℓ2-norm-based saliency-driven column selection**
Difficulty in adaptation to heterogeneous model structures	Lack of structure-aware grouping	**Frequency-aware multi-parameter intra-row grouping**
Extra memory cost is not small		**Intra-frequency-band mean sharing**

Table 1. Enhancements used in HBLLM against weaknesses in 1-bit PTQ methods

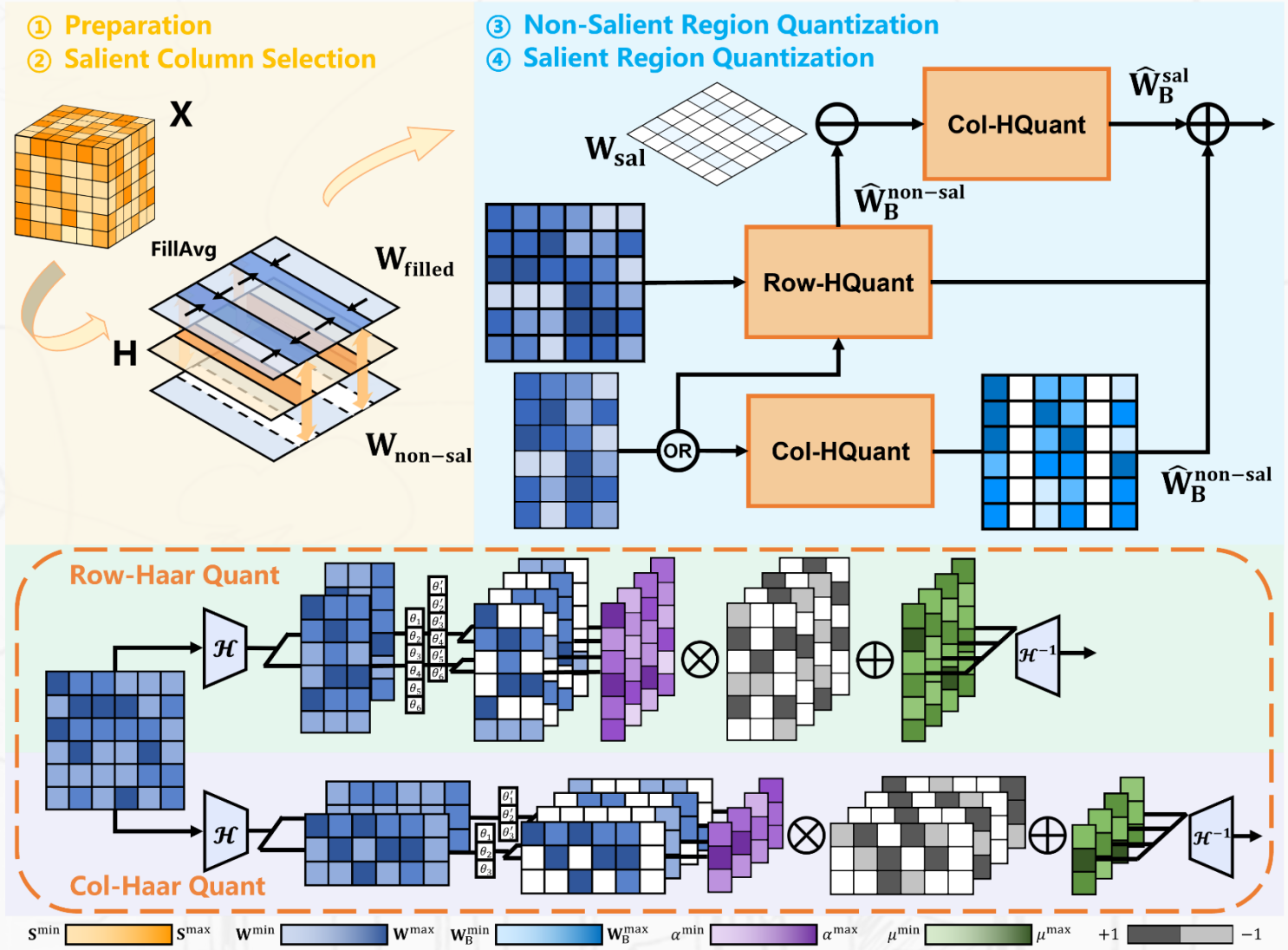
Method

Quantization Pipeline Overview:

HBLLM integrates the Haar transform into a BiLLM-style quantization pipeline.

1. Preparation Phase:

Compute the column-wise importance scores using a Hessian-based saliency metric.



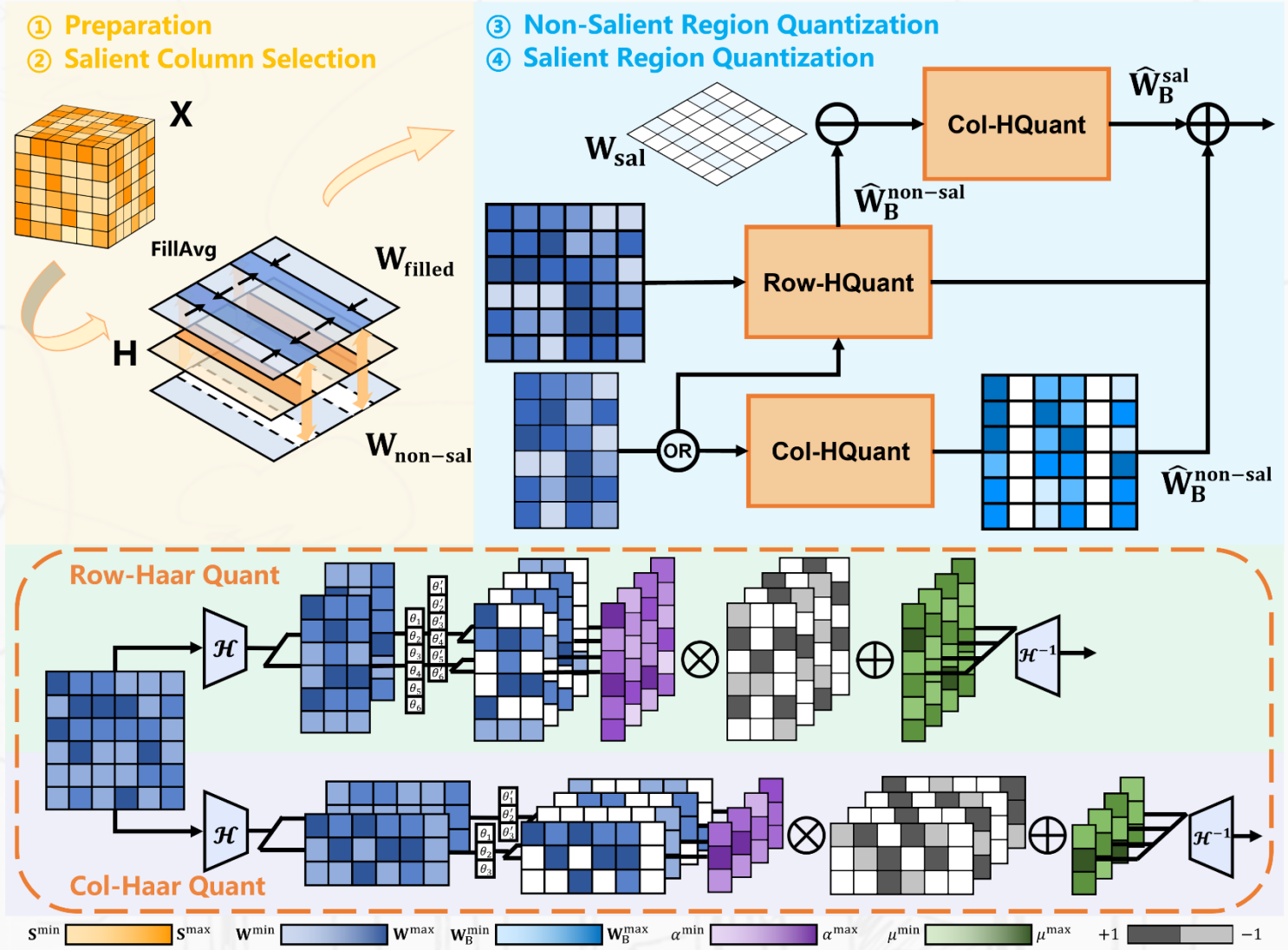
Method

Quantization Pipeline Overview:

HBLLM integrates the Haar transform into a BiLLM-style quantization pipeline.

2. Salient Column Selection and Quantization:

Select top-K salient columns → apply HaarQuant → keep minimal-error subset



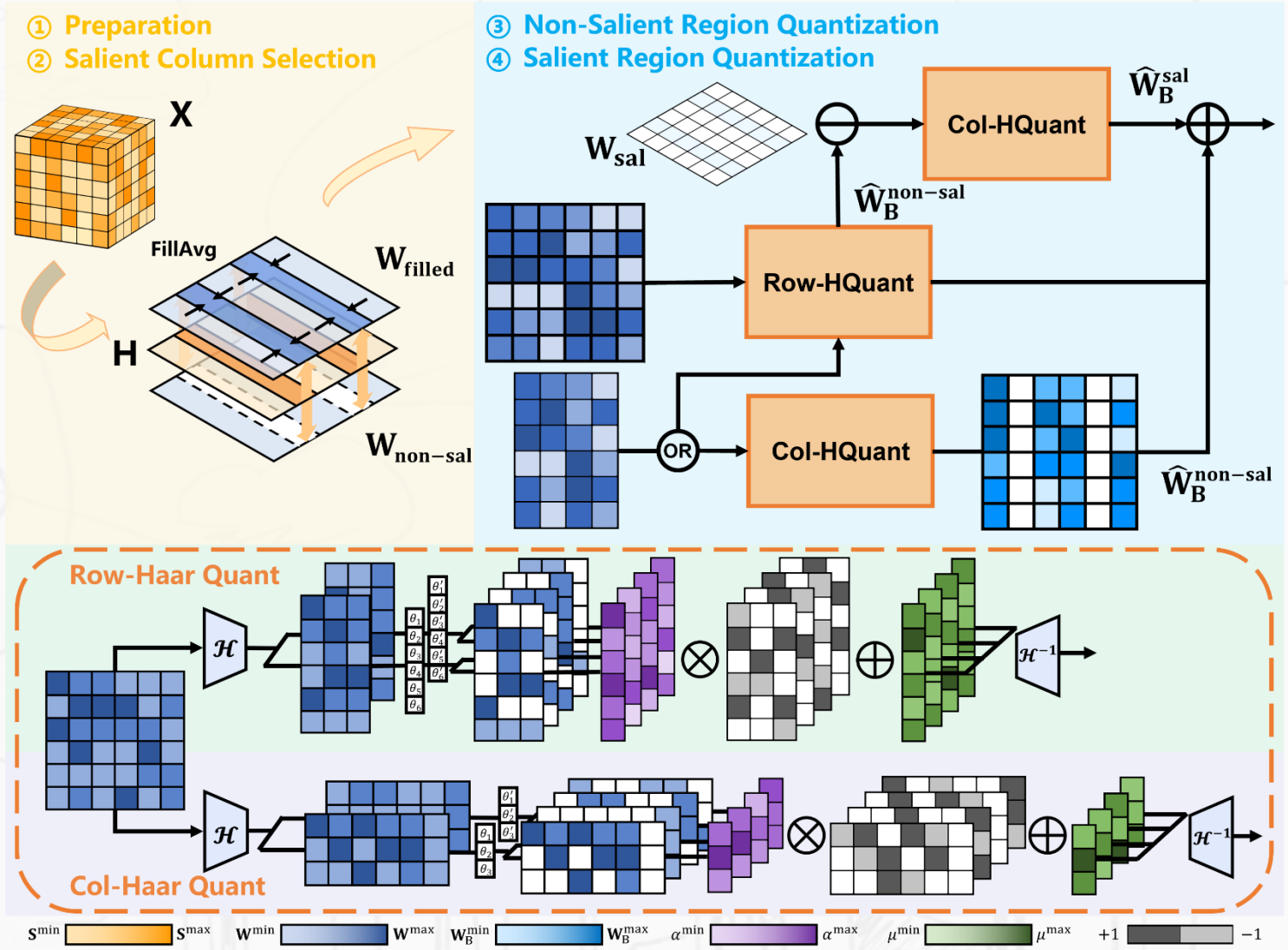
Method

Quantization Pipeline Overview:

HBLLM integrates the Haar transform into a BiLLM-style quantization pipeline.

3. Non-Salient Region Quantization:

Fill missing values (FillAvg) → apply HaarQuant again

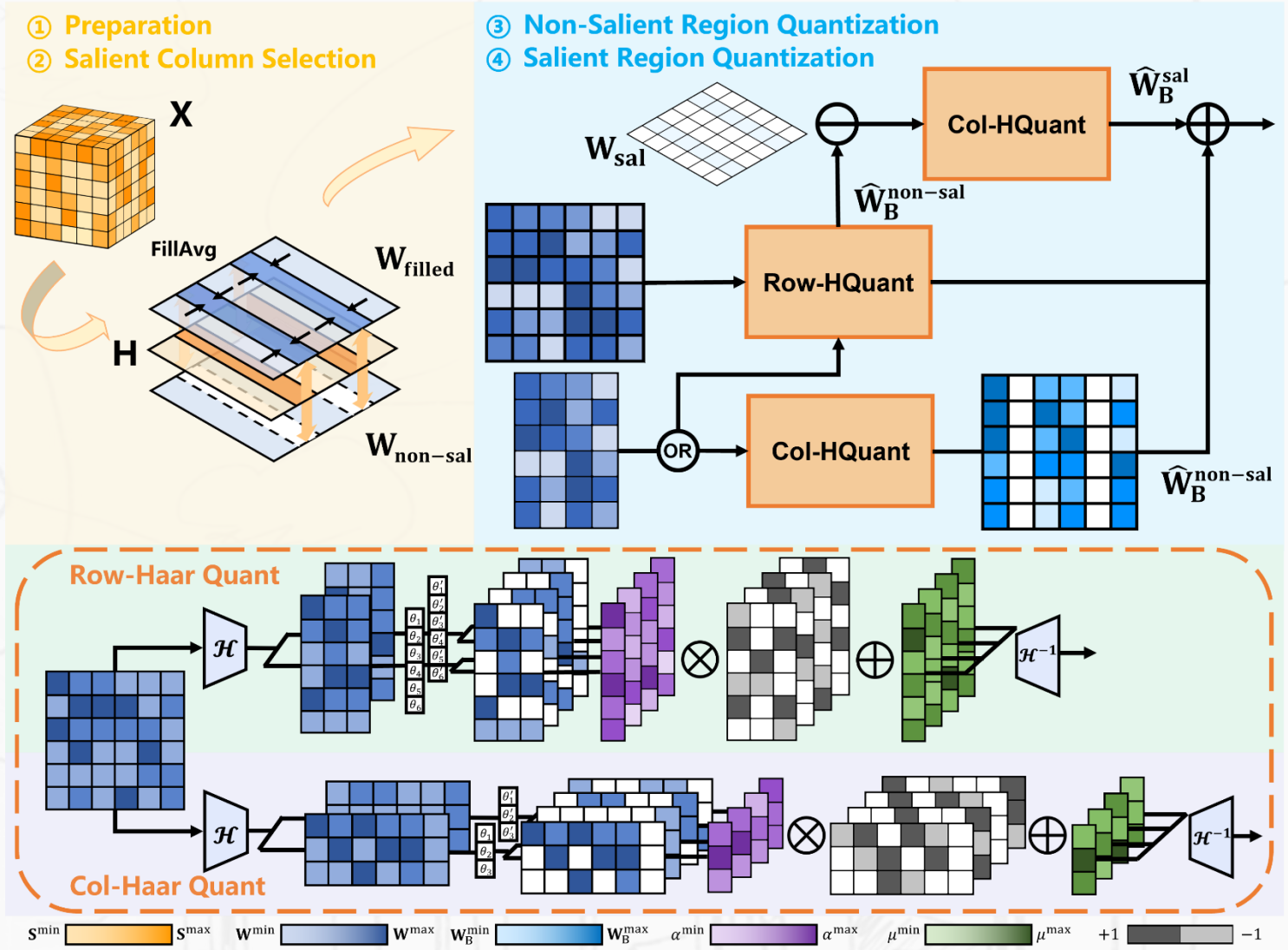


Method

Quantization Pipeline Overview:

HBLLM integrates the Haar transform into a BiLLM-style quantization pipeline.

4. Adjustment and Refinement



Experiments

Experimental Settings

- **Dataset:** see the following figure
- **Baselines:**

Full Precision、 BiLLM[2]、ARB-LLM[3]、PB-LLM[4]、FrameQuant[5]

- **Backbone models:**

OPT, LLaMA-1, LLaMA-2, and LLaMA-3

- **Evaluation Metircs:**

Perplexity (↓), Accuracy (↑)

Perplexity↓
WikiText2
C4 PTB

CommonSenseQA↑
PIQA BoolQ ARC-e ARC-c
HellaSwag Winogrande
COPA OBQA LAMABADA

[1] Frantar E, Ashkboos S, Hoefler T, et al. OPTQ: Accurate Post-training Quantization for Generative Pre-trained Transformers.*11th International Conference on Learning Representations(ICLR)*. 2023.
[2] Huang W, Liu Y, Qin H, et al. BiLLM: Pushing the Limit of Post-training Quantization for LLMs. *Proceedings of the 41st International Conference on Machine Learning(ICML).* 2024: 20023-20042.
[3] Li Z, Yan X, Zhang T, et al. ARB-LLM: Alternating Refined Binarizations for Large Language Models. *arXiv preprint arXiv:2410.03129*, 2024.
[4] Shang Y, Yuan Z, Wu Q, et al. PB-LLM: Partially Binarized Large Language Models. *The Twelfth International Conference on Learning Representations(ICLR).* 2024.
[5] Adepu H, Zeng Z, Zhang L, et al. FrameQuant: Flexible Low-Bit Quantization for Transformers. *Forty-first International Conference on Machine Learning(ICML).* 2024.

Experiments

Summary of Experiment results:

HBLLM Achieves SOTA Performance in 1-bit PTQ methods

- The ppl ratio between HBLLM and the original FP16 model remains within the range of 1.2–2.2 (in Fig.1)
- 73.8~88.8% of the original model’ s accuracy in QA tests
- 77.1~80.5% memory of weights saving
- The expected inference latency is approximately 31.8% of the FP16 baseline inference time

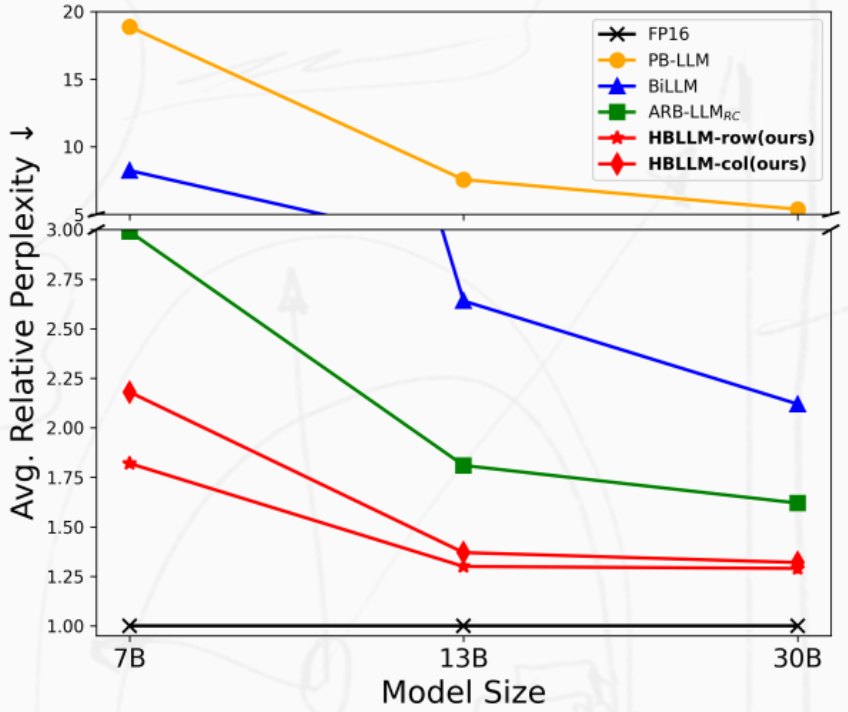


Figure 1: Average relative perplexity (normalized to FP16) on PTB, WikiText2, and C4 for LLaMA-1 family models, comparing LLM binarization methods and our HBLLM.

Experiments

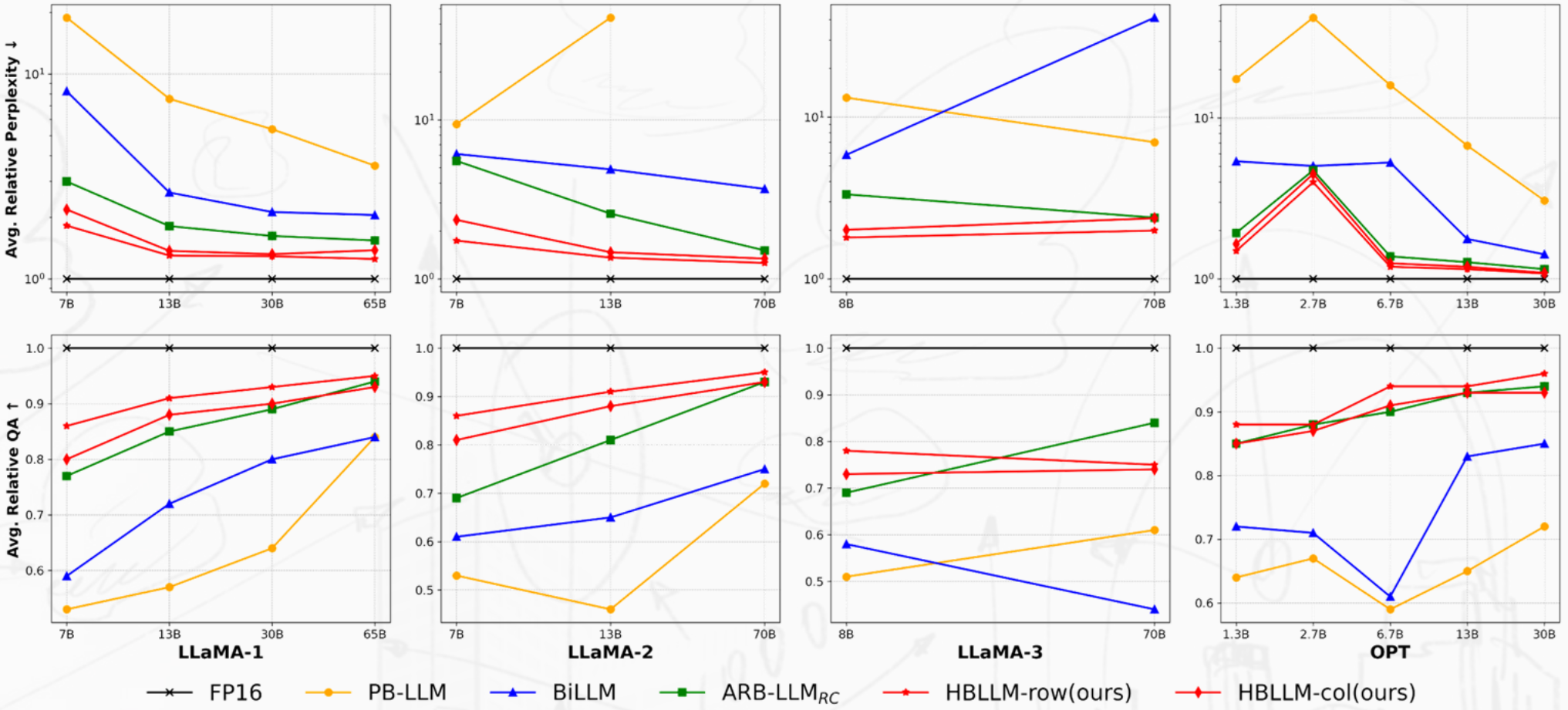
LLaMA1 Size	Method	W-bits	Perplexity↓ C4	Wiki2	PTB	AvgQA↑
7B	FullPrecision	16.00	6.71	5.68	35.80	65.62
	FrameQuant	2.20	10.89	9.96	104.7	56.19
	PB-LLM	1.70	90.19	113.4	830.0	35.71
	BiLLM	1.09	43.74	44.85	369.3	40.01
	ARB-LLM$_X$	1.09	22.80	24.70	240.5	45.65
	ARB-LLM$_{RC}$	1.09	15.13	13.45	155.8	52.23
	HBLLM-row	1.09	**9.49**	**8.82**	**88.86**	**57.48**
	HBLLM-col	1.00	10.38	9.67	117.7	54.03
13B	FullPrecision	16.00	6.24	5.09	25.36	68.09
	FrameQuant	2.20	8.79	7.84	50.69	60.69
	PB-LLM	1.70	38.41	46.02	190.2	40.39
	BiLLM	1.10	13.93	14.99	69.75	50.89
	ARB-LLM$_X$	1.10	N/A	N/A	N/A	N/A
	ARB-LLM$_{RC}$	1.10	10.68	10.19	43.85	59.58
	HBLLM-row	1.09	**7.62**	**6.68**	**34.94**	**62.57**
	HBLLM-col	1.00	7.77	6.98	37.62	61.25
65B	FullPrecision	16.00	5.31	3.53	21.11	72.27
	FrameQuant	2.20	6.69	5.55	27.48	68.58
	PB-LLM	1.70	12.66	12.76	99.67	62.48
	BiLLM	1.10	9.26	8.58	41.93	62.05
	ARB-LLM$_X$	1.10	N/A	N/A	N/A	N/A
	ARB-LLM$_{RC}$	1.10	7.48	6.47	29.14	68.53
	HBLLM-row	1.09	**6.28**	**5.07**	**24.11**	**69.18**
	HBLLM-col	1.00	6.44	5.26	30.38	67.83

LLaMA2 Size	Method	W-bits	Perplexity↓ C4	Wiki2	PTB	AvgQA↑
7B	FullPrecision	16.00	8.66	6.94	37.86	65.54
	FrameQuant	2.20	14.66	13.34	177.1	52.75
	PB-LLM	1.70	63.95	55.40	486.2	36.54
	BiLLM	1.08	33.97	31.38	373.0	42.11
	ARB-LLM$_X$	1.08	26.55	21.74	314.2	45.41
	ARB-LLM$_{RC}$	1.08	17.87	15.85	462.2	46.71
	HBLLM-row	1.07	**11.75**	**10.52**	**89.23**	**57.74**
	HBLLM-col	1.00	12.51	11.33	150.6	54.09
13B	FullPrecision	16.00	6.18	4.88	43.02	69.18
	FrameQuant	2.20	9.40	7.80	109.3	61.35
	PB-LLM	1.70	313.4	289.4	934.4	32.91
	BiLLM	1.08	22.17	19.57	303.4	46.76
	ARB-LLM$_X$	1.08	N/A	N/A	N/A	N/A
	ARB-LLM$_{RC}$	1.08	11.90	10.98	151.8	57.35
	HBLLM-row	1.07	**7.82**	**6.71**	**61.75**	**63.61**
	HBLLM-col	1.00	8.28	7.00	69.74	62.04
70B	FullPrecision	16.00	5.24	3.32	21.49	72.96
	FrameQuant	2.20	N/A	N/A	N/A	N/A
	PB-LLM	1.70	N/A	N/A	N/A	54.26
	BiLLM	1.09	15.57	15.86	71.03	55.81
	ARB-LLM$_X$	1.09	N/A	N/A	N/A	N/A
	ARB-LLM$_{RC}$	1.09	7.26	6.00	28.43	68.77
	HBLLM-row	1.08	**6.18**	**4.82**	**24.69**	**70.01**
	HBLLM-col	1.00	6.63	5.04	26.31	68.61

Note: All methods are calibrated on C4 with 128 samples and a sequence length of 2048. A block size of 128 is used for channel-wise quantization, as commonly done in prior work. N/A: ARB-LLM$_X$ method cannot run on a single 3090 GPU - 24GB. W-bits is the average weight overhead per weight.

- ✓ **Overall: Achieve SOTA performance**
- ✓ **Flexibility: Be beneficial to various LLMs**

Experiments



- ✓ On language modeling tasks, the relative perplexity(vs. FP16) remains within the range of **1.2–2.2**, outperforming the next-best methods by **33%–66%**
- ✓ On 9 zero-shot QA benchmarks, HBLLM retains **73.8%–88.8%** of the original model’ s accuracy

Conclusion

- We propose **HBLLM**, a 1-bit **weight-only** PTQ method built upon the BiLLM pipeline with **Haar transform**.
- HBLLM **outperforms SOTA 1-bit PTQ methods** across multiple LLM families and benchmarks.

Thank you for watching!

NEURAL INFORMATION PROCESSING SYSTEMS

广东省计算科学重点实验室
Guangdong Province Key Laboratory of Computational Science

中山大學
SUN YAT-SEN UNIVERSITY

Paper

Code