ICML
International Conference
On Machine Learning

Paper
arXiv:2512.00956

Code
IST-DASLab/WUSH

WUSH: Near-Optimal Adaptive Transforms for LLM Quantization

Jiale Chen (jiale.chen@ist.ac.at)

Vage Egiazarian

Roberto L. Castro

Torsten Hoefler

Dan Alistarh

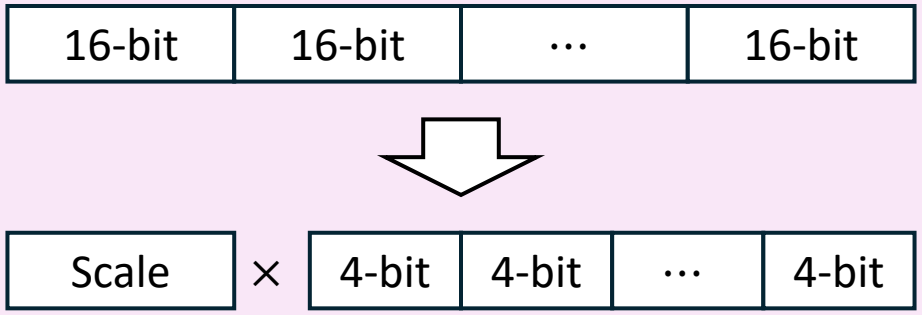
Institute of Science and Technology Austria

Red Hat

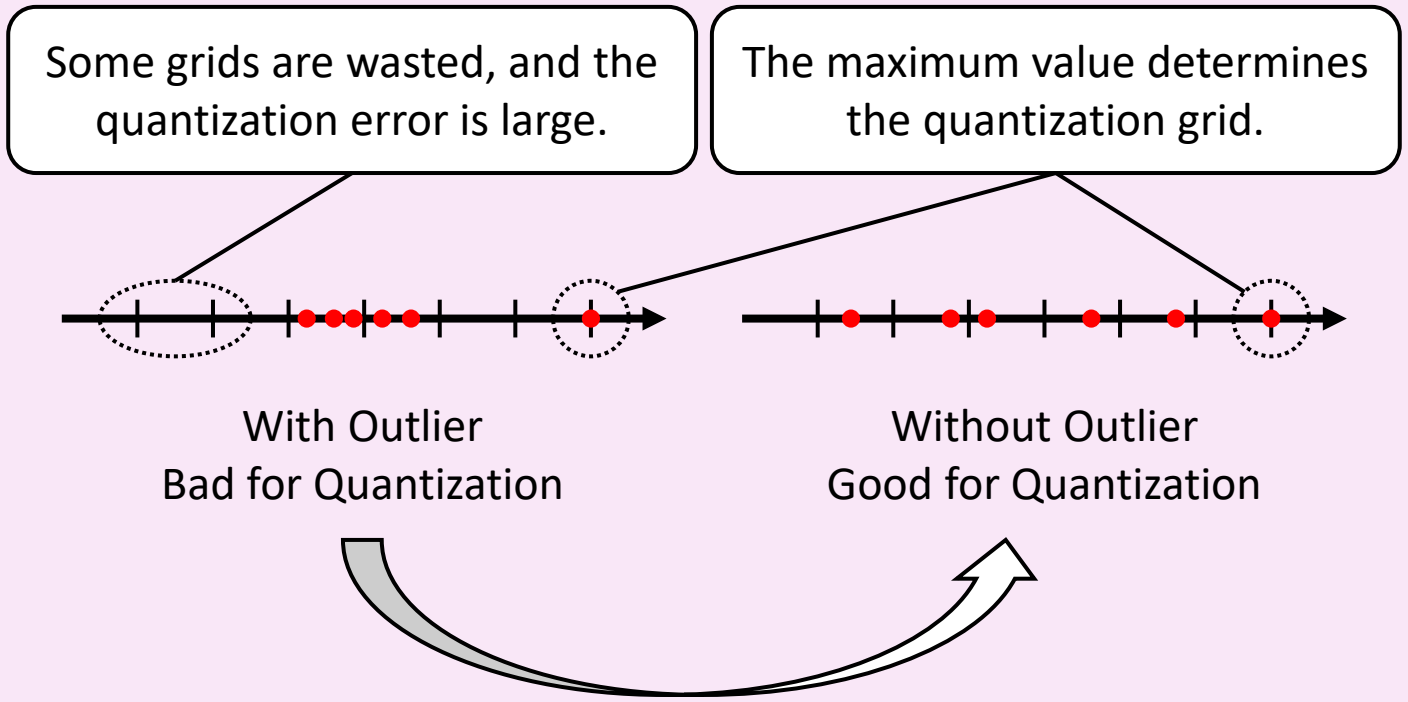
ETH zürich

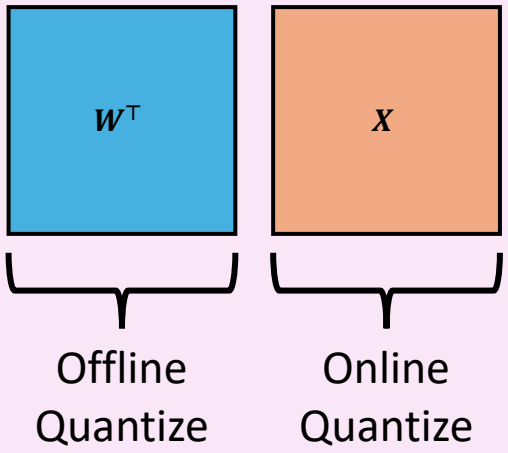
Background

- Quantization compresses full-precision values into low-bit values plus a shared high-precision scale.
- Goal: lower memory and faster inference for LLMs, while preserving accuracy.

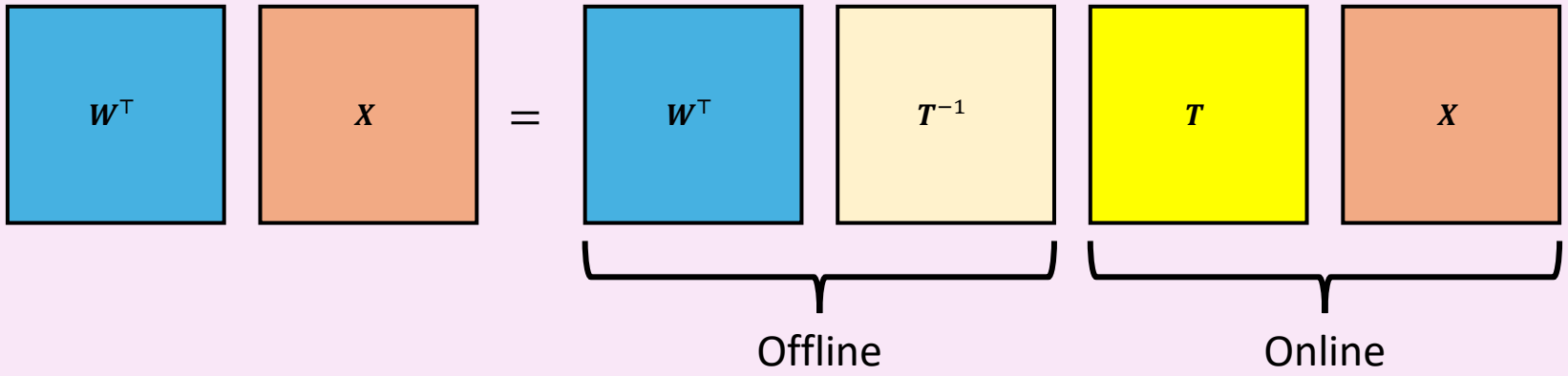


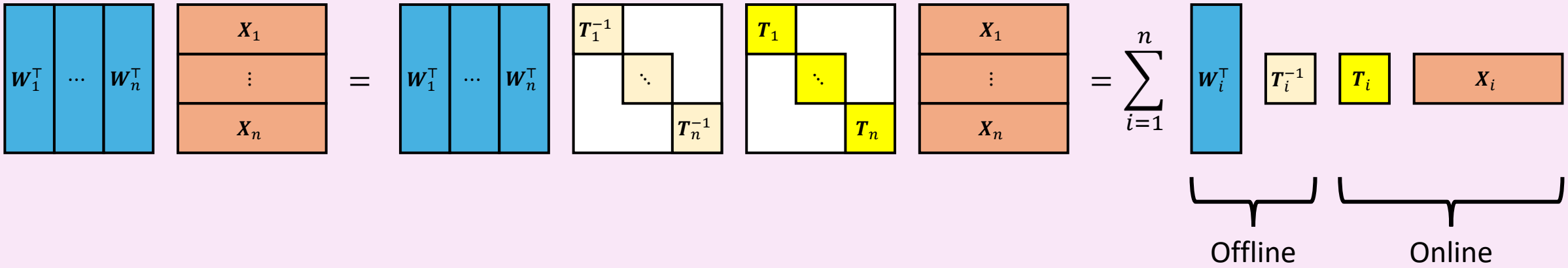
- The presence of outliers is the main challenge for LLM quantization.
- Outliers stretch the scale, while most normal values then receive fewer effective quantization grid points.
- Transforms can redistribute outlier energy and better preserve the accuracy.





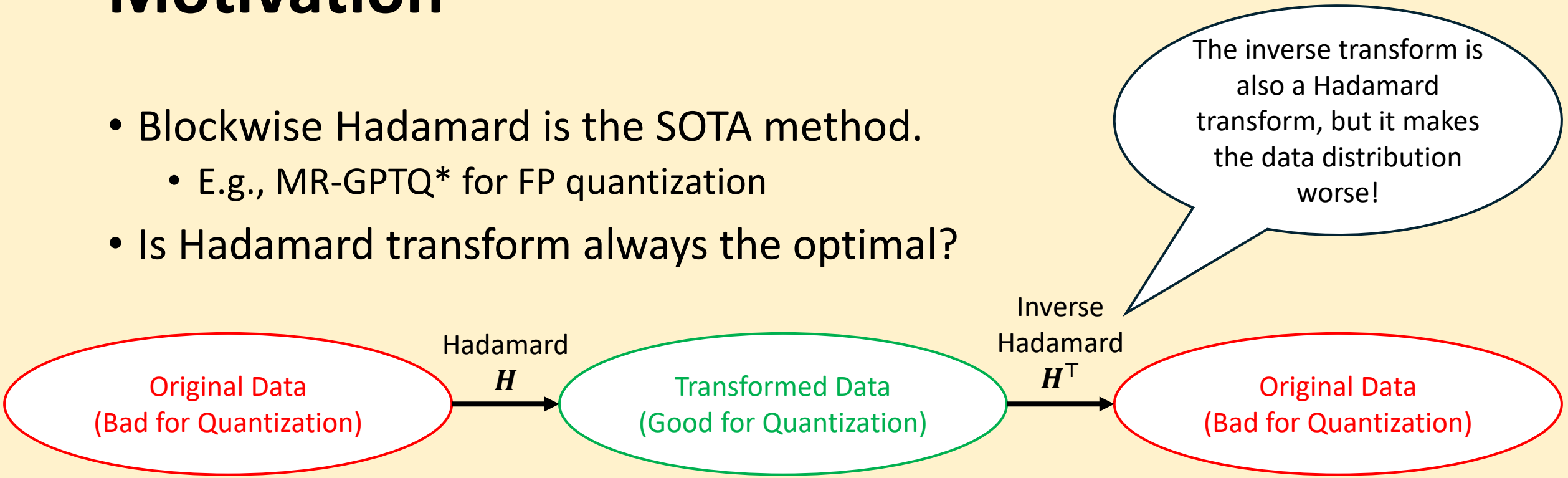
- For a linear layer in LLM: transform-quantize weights offline; transform-quantize activations online.
- Before quantization, the matrix product is unchanged.
- Block transforms are more computationally efficient.





Motivation

- Blockwise Hadamard is the SOTA method.
  - E.g., MR-GPTQ* for FP quantization
- Is Hadamard transform always the optimal?

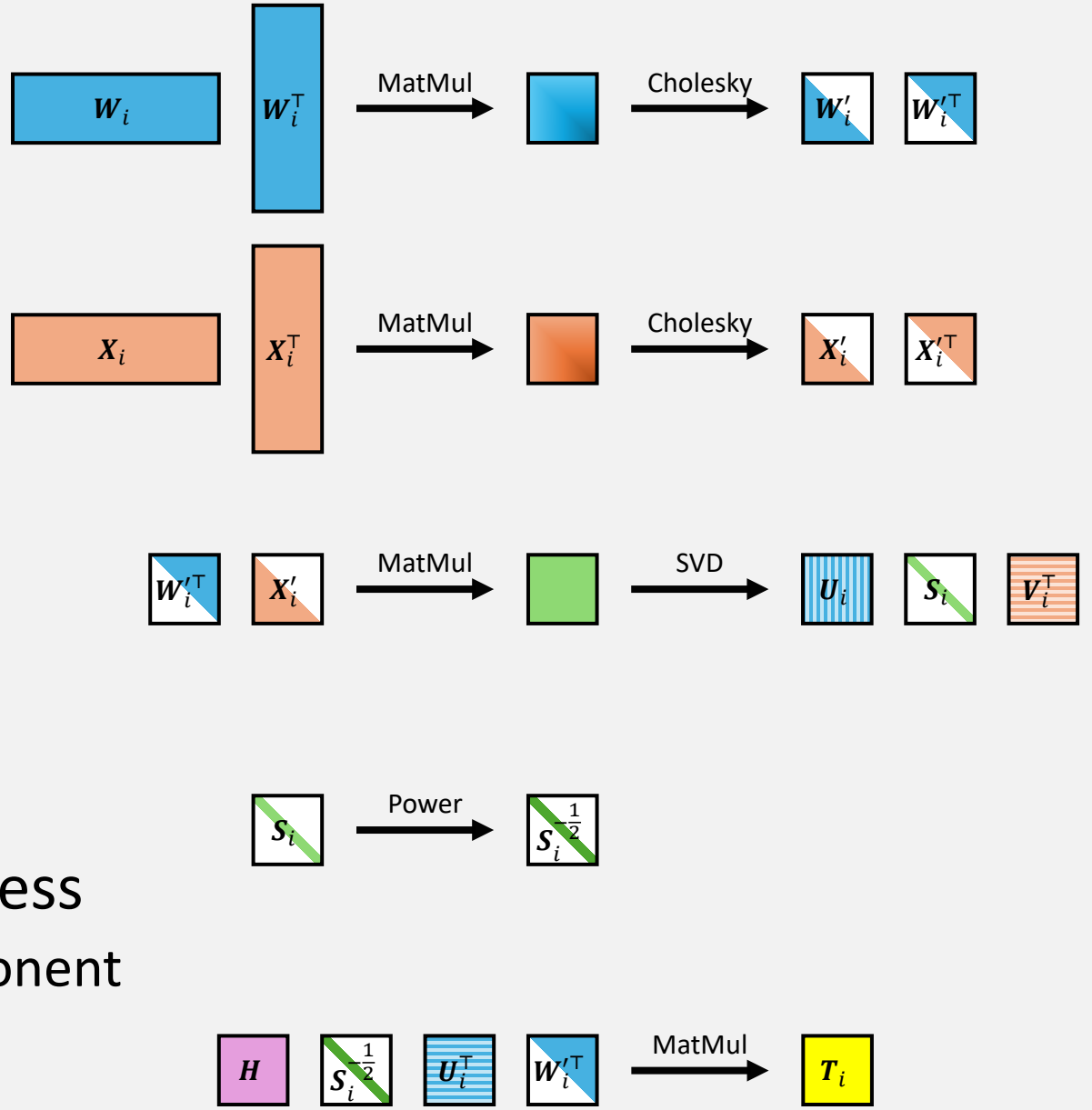


- No! The optimal transform must be data-dependent!

* Egiazarian et al. Bridging the gap between promise and performance for microscaling FP4 quantization. ICLR 2026.

WUSH Formulation

- WUSH
  - Weight-Unitary-Singular-Hadamard
- Data-aware
- Closed-form (Cholesky, SVD, etc.)
- Proved (under mild assumptions)
  - Optimal for FP
  - Asymptotically optimal for INT
- Explains Hadamard's empirical success
  - It is the only data-independent component



Layerwise Loss

Table 1. Layerwise RTN quantization loss in the unit of 10^{-3}.

		Q	K	V	O	G	U	D
MXFP4	I	11.1	12.0	10.7	4.35	7.10	6.56	5.47
	R	7.61	7.73	9.14	3.84	5.56	5.68	4.07
	H	7.24	7.20	8.60	3.79	5.45	5.61	3.90
	WUS	6.27	7.22	4.05	3.57	5.76	4.75	4.46
	WUSH	**3.34**	**3.34**	**3.30**	**2.76**	**4.49**	**4.39**	**3.39**
NVFP4	I	4.23	4.35	4.37	2.34	3.49	3.41	2.41
	R	4.98	5.01	5.86	2.54	3.63	3.68	2.66
	H	5.60	5.62	6.70	2.58	3.71	3.79	2.78
	WUS	**2.26**	**2.36**	2.30	2.00	**3.04**	**3.01**	**2.23**
	WUSH	2.40	2.44	**2.28**	**1.92**	3.09	3.02	2.33
INT4	I	170.	123.	13.2	4.55	55.9	9.83	19.3
	R	5.79	5.84	6.96	2.94	4.22	4.29	3.10
	H	5.57	5.55	6.80	2.86	4.09	4.25	3.03
	WUS	213.	142.	10.7	4.54	50.2	7.42	13.1
	WUSH	**2.39**	**2.43**	**2.54**	**2.10**	**3.43**	**3.43**	**2.55**

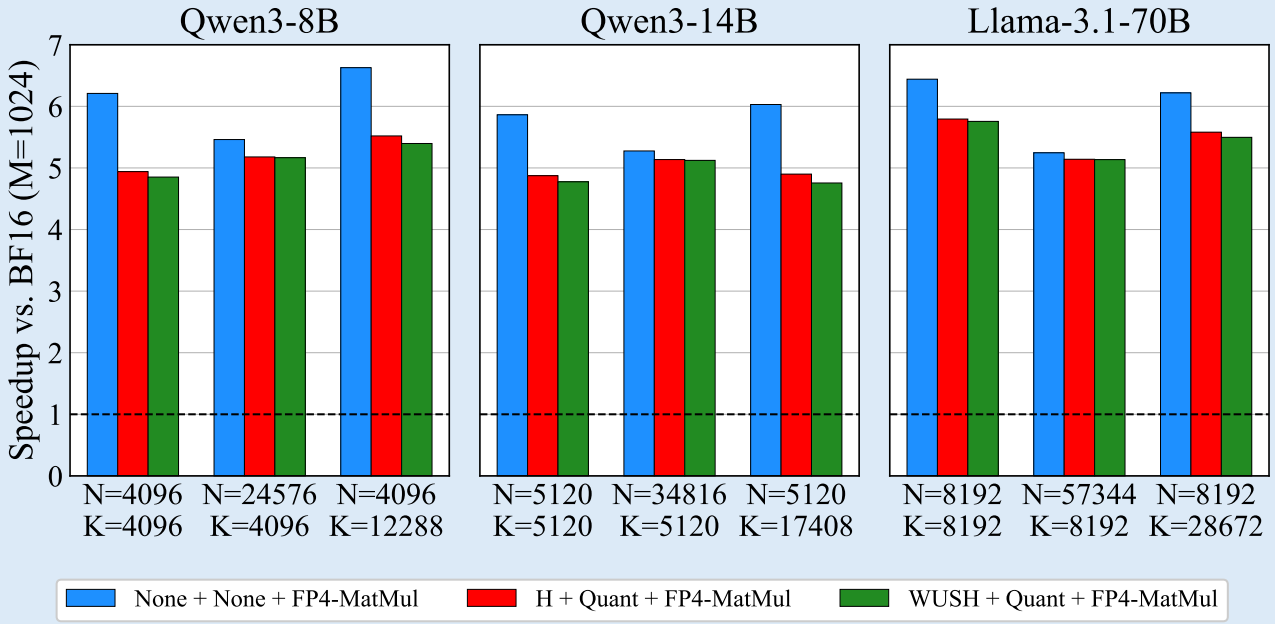
End-to-End Benchmarks

Table 2. Llama-3.1-8B-Instruct W4A4 accuracy results on the LM Evaluation Harness under different methods.

Format	Method	MMLU-CoT	GSM8K	HellaSwag	WinoGrande	Average	Recovery
BF16	-	72.76	85.06	80.01	77.90	78.93	100.0
NVFP4	RTN-I	68.26	78.39	78.15	74.11	74.73	94.67
	RTN-H	67.41	78.01	77.31	73.48	74.05	93.82
	SmoothQuant	68.90	79.50	79.50	74.70	75.70	95.90
	QuaRot	66.50	77.40	77.25	75.14	74.10	93.80
	SpinQuant	66.50	76.10	76.96	75.32	73.70	93.40
	GPTQ-I	68.85	81.25	78.26	74.51	75.72	95.92
	GPTQ-H (MR-GPTQ)	69.12	80.80	78.17	75.24	75.84	96.08
	RTN-WUSH	68.83	78.57	78.22	75.47	75.28	95.37
	GPTQ-WUSH	69.69	80.11	78.52	76.09	**76.10**	**96.40**
MXFP4	RTN-I	62.21	67.85	73.99	73.24	69.32	87.83
	RTN-H	62.38	72.48	75.29	71.67	70.45	89.26
	SmoothQuant	63.93	68.54	75.10	73.56	70.30	89.06
	QuaRot	49.86	56.94	73.50	71.43	62.90	79.70
	SpinQuant	61.80	68.16	74.87	72.93	69.40	88.00
	GPTQ-I	63.49	68.46	76.01	74.51	70.62	89.47
	GPTQ-H (MR-GPTQ)	67.19	75.70	76.91	74.80	73.65	93.31
	RTN-WUSH	66.85	75.16	77.28	73.56	73.21	92.75
	GPTQ-WUSH	67.79	77.41	77.44	74.78	**74.35**	**94.20**

GPU Inference Kernels

- Fused transform-quantization kernel and FP4 matrix multiplication kernel.
- WUSH (different transform per block) is nearly as fast as Hadamard (same transform per block).



Thanks for Listening

- Please refer to our paper and source code for more details.

Paper
arXiv:2512.00956

Code
IST-DASLab/WUSH