THE CHIPS TO SYSTEMS CONFERENCE

SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

THE CHIPS TO SYSTEMS CONFERENCE
SHAPING THE NEXT GENERATION OF ELECTRONICS
JUNE 23-27, 2024
MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

北京大学集成电路学院
SCHOOL OF INTEGRATED CIRCUITS,PEKING UNIVERSITY

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Oltron: Software-Hardware Co-design for Outlier-Aware Quantization of LLMs with Inter-/Intra-Layer Adaptation

Chenhao Xue[1,4], Chen Zhang[2,✉], Xun Jiang[1] , ZhuTianYa Gao[2], Yibo Lin[1,3,4] , Guangyu Sun[1,3,4,✉]

[1] School of Integrated Circuits, Peking University

[2] Shanghai Jiao Tong University

[3] Institute of EDA, Peking University, Wuxi, China

[4] Beijing Advanced Innovation Center for Integrated Circuits

Quantization for Large Language Models

Original tensor
(16-bit float)

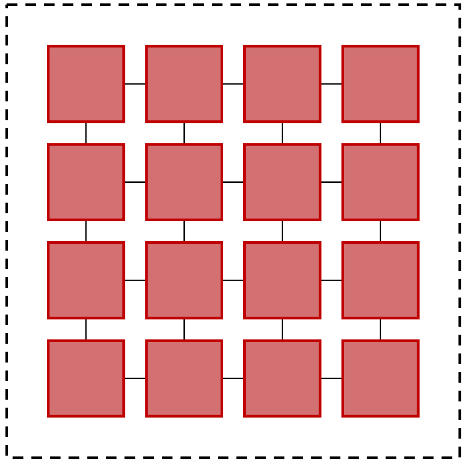
2.09	-0.98	1.48	0.09
0.05	-0.14	-1.08	2.12
-0.91	1.92	0	-1.03
1.87	0	1.53	1.49

Quantization

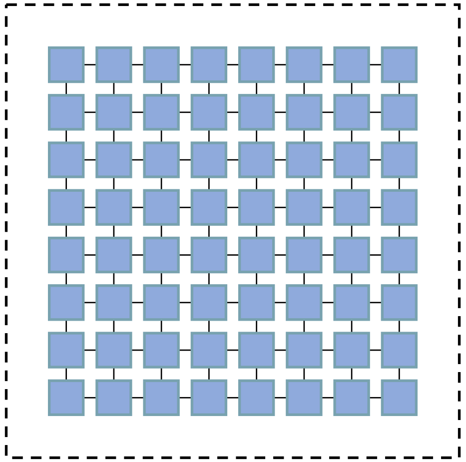
2	-1	1	0
0	0	-1	2
-1	2	0	-1
2	0	2	1

Quantized tensor
(4-bit int)

Float
PE Array

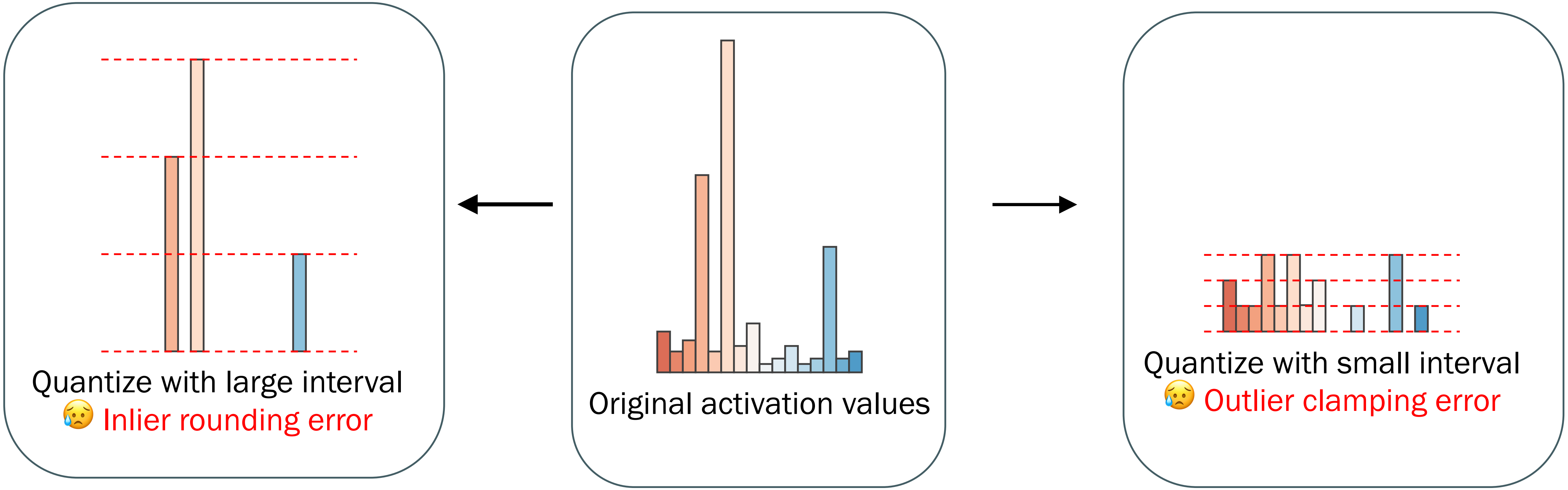


Integer
PE array



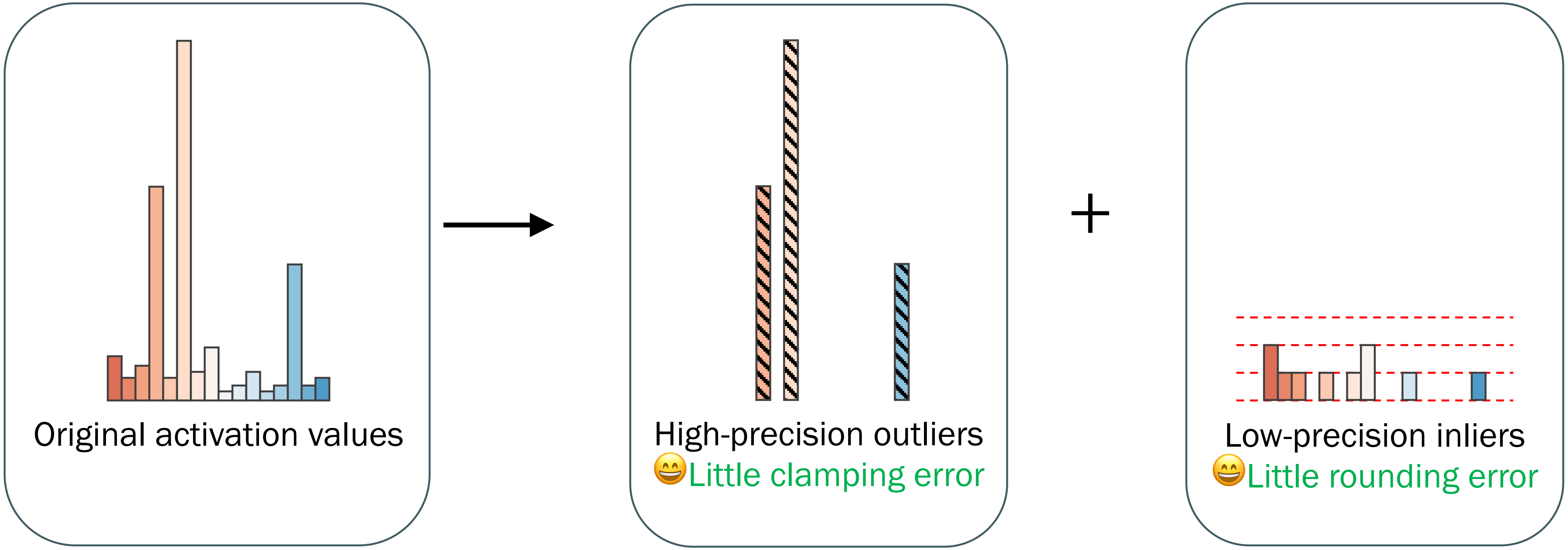
Activation Outlier Issue

- A small fraction of activation values have extremely large magnitudes [1,2]
  - Uniform quantization leads to significant model accuracy losses.



[1] Tim Dettmers et al. Llm. int8 (): 8-bit matrix multiplication for transformers at scale
[2] Zhewei Yao et al. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers

Outlier-Aware Quantization



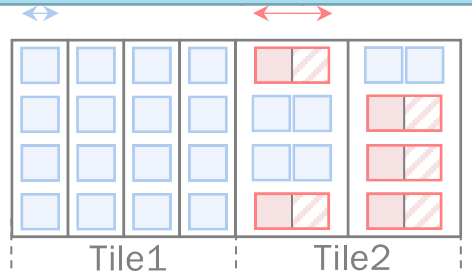
- Challenges in efficient implementation
  - Customized encoding for mixed-precision storage
  - Dedicated hardware for mixed-precision computation

Limitation of Previous Arts

- Split Encoding [1,2]
  - Separately store outlier values with compressed sparse format

- Outlier-Victim Pair Encoding [3]
  - Locally extend outlier bit-width by pruning adjacent victim values

Can we strike a balance between model accuracy and hardware efficiency for outlier-aware quantization of LLMs?

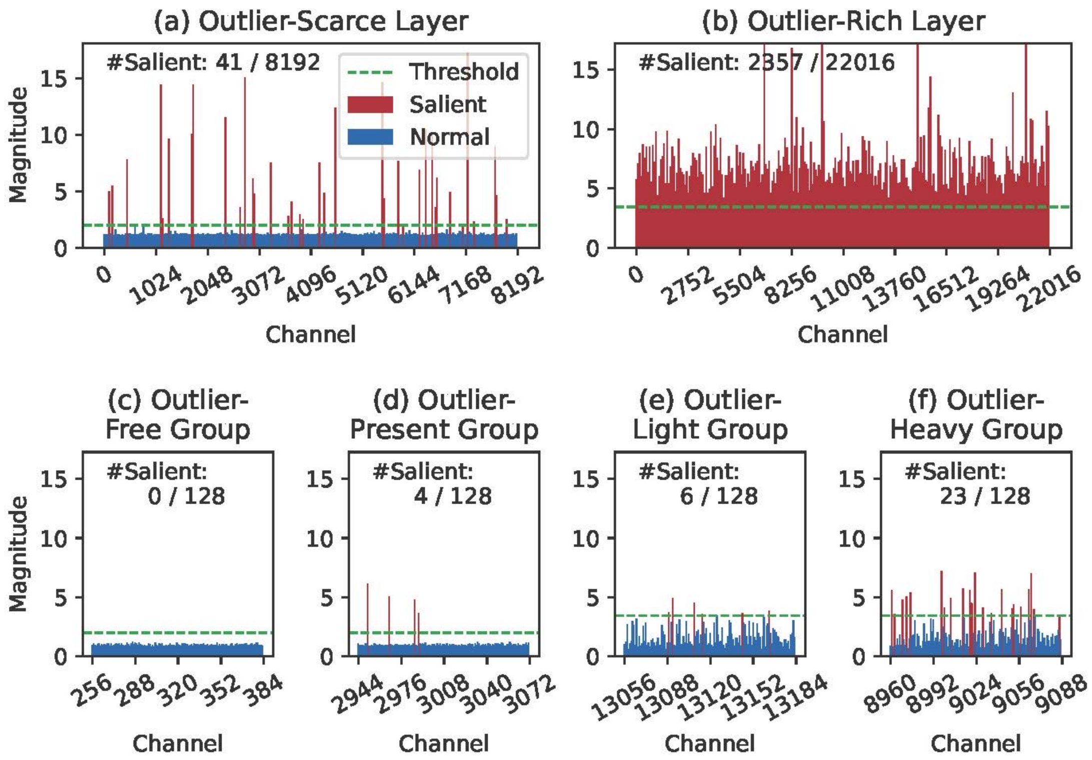


[1] Eunhyeok Park et al. 2018. Energy-efficient neural network accelerator based on outlier-aware low-precision computation
[2] Ali Hadi Zadeh et al. 2020. Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference
[3] Cong Guo et al. 203. OliVe: Accelerating Large Language Models via Hardware friendly Outlier-Victim Pair Quantization

normal outlier indexing

Activation Outlier Characteristics

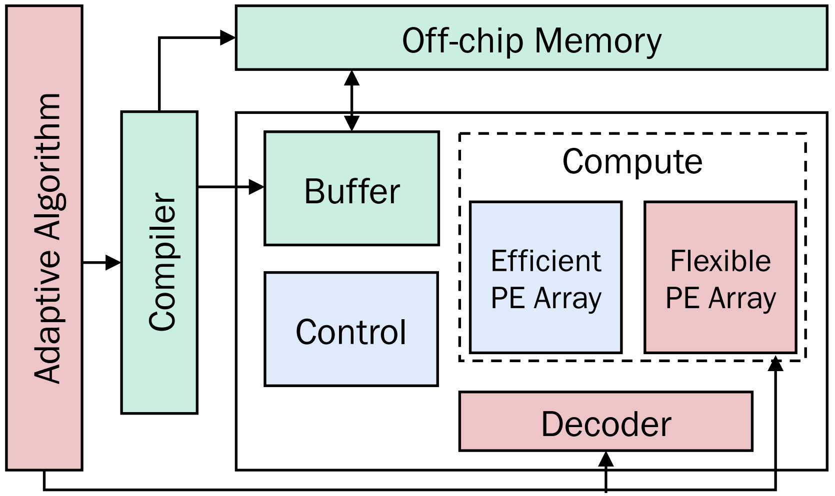
- Observation 1: salient channels
  - Activation outliers tend to only cluster in certain channels
- Observation 2: inter-layer heterogeneity
  - Different layers exhibit significantly different ratios of salient channels
- Observation 3: intra-layer heterogeneity
  - Salient channels are randomly distributed across different channel groups



Collected from LLaMA-65B with 128 calibration sequences
Threshold = 2 × median magnitude
(c,d)/(e,f) are zoom-in views of (a)/(b)

Oltron's Quantization Framework

- Key insight
  - Encode salient channels at high precision
  - Maintain representation regularity unaffected by non-uniform salient channel distribution
- Intra-layer heterogeneity adaptation
  - Tile-wise Outlier-Balanced Encoding
  - Dataflow Optimization
- Inter-layer heterogeneity adaptation
  - Adaptive quantization algorithm
  - Reconfigurable architecture



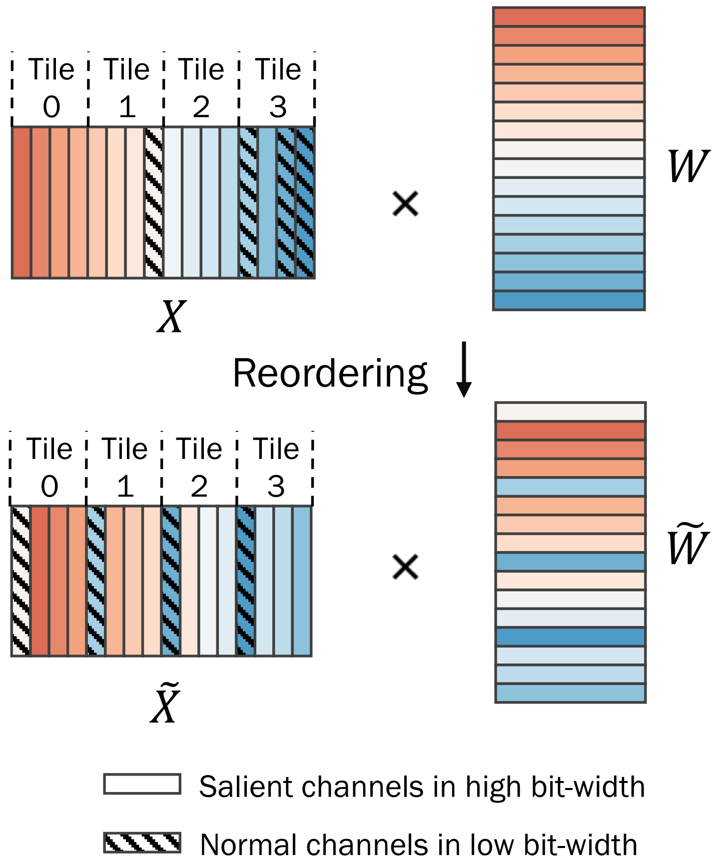
😄 Aligned memory access

😄 Flexible outlier precision & ratio

😄 Efficient hardware implementation

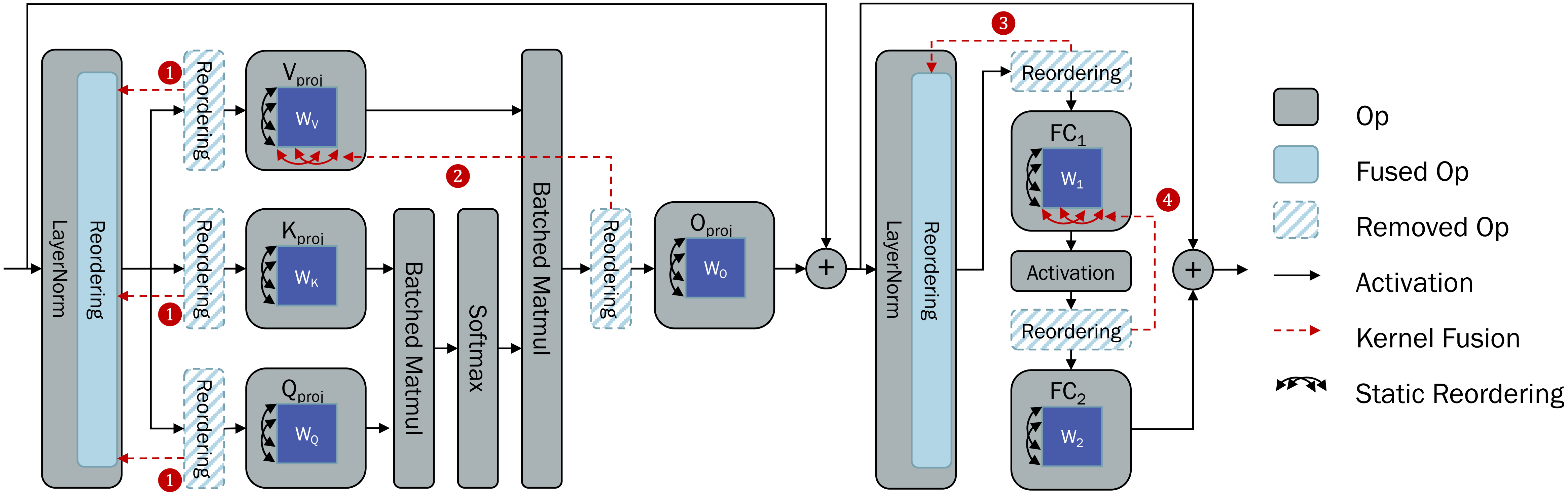
Tile-wise Outlier-Balanced Encoding

- TOBE for activation matrix X
  - Statically determine activation salient channels with profiled distribution
  - Evenly distribute salient channels into tiled sub-blocks
    - Regular off-chip memory access
  - Always place salient channels at forefront of a tile
    - Regular on-chip memory access
- TOBE-based matrix multiplication
  - Statically reorder rows of weight matrix W w.r.t. column permutation of X



Dataflow Optimization

- Mitigate explicit reordering overhead to prepare TOBE data layout
- Strategy 1: commutative operators
  - Reorder non-reductive dimensions of previous operators (2, 4)
- Strategy 2: kernel fusion
  - Adjust the write address of previous operator's result (1, 3)



Adaptive Quantization Algorithm

Algorithm 1 Adaptive Quantization Algorithm

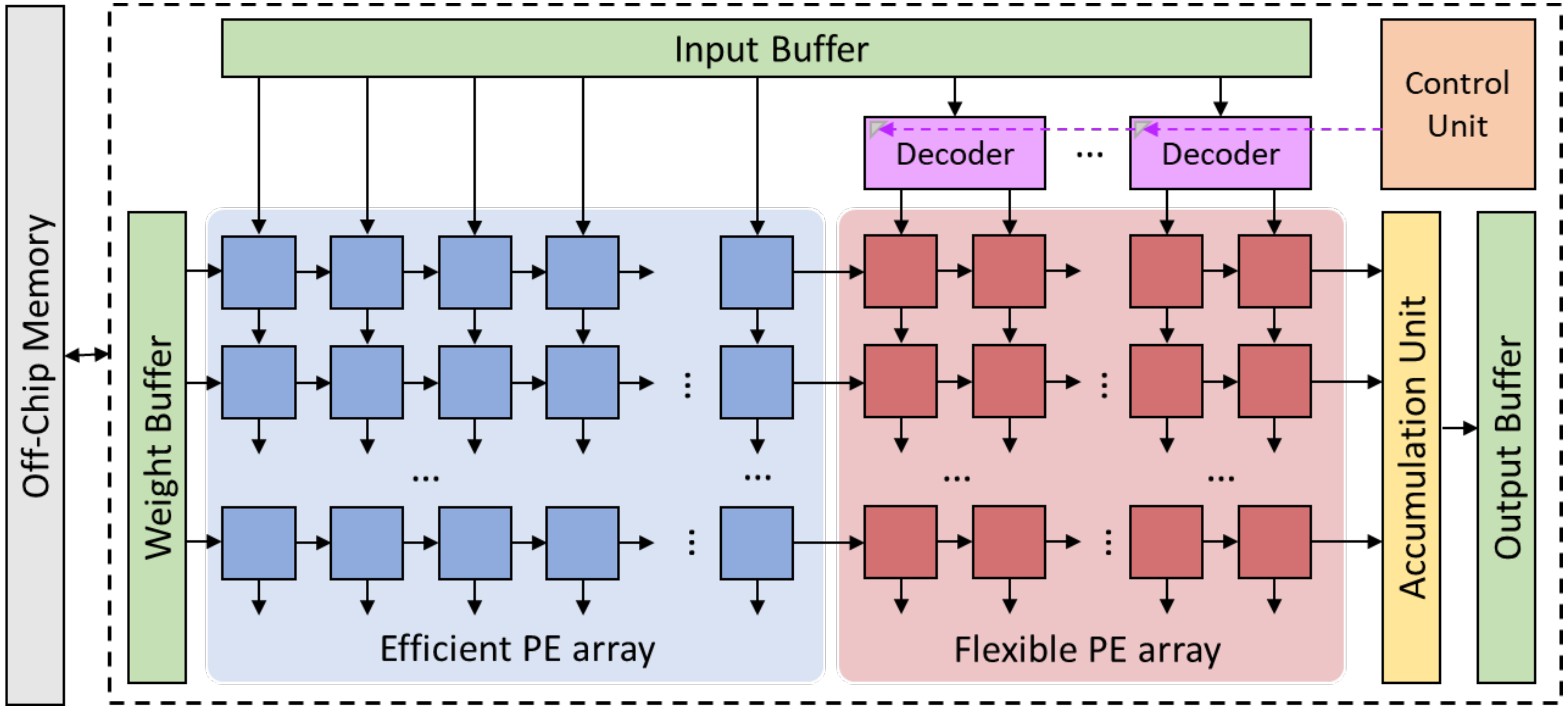
Require:
- $\mathcal{M} = \{M \in \mathbb{R}^{\#\text{batch} \times d_l} | 1 \leq l \leq L\}$: activation statistics;
- $B^* \in \mathbb{R}$: target storage budget;

Ensure:
- $\mathcal{S} = \{S_l | S_l \subseteq \{1, 2, \cdots d_l\}, 1 \leq l \leq L\}$: salient channel sets;
- $\vec{t} \in \mathcal{T}^L$: layer-wise salient channel data types;

$B \leftarrow 0$
for layer $l \in \{1, 2, \cdots, L\}$ **do**
    $\tau_l \leftarrow 3 \times \text{Standard-Deviation}(M_l)$
    $t_l \leftarrow \text{FP8}$
while target budget B^* not reached **do**
    $\langle e, B \rangle \leftarrow \text{Estimate-MSE-And-Budget}(\mathcal{M}, \vec{\tau}, \vec{t})$
    $i \leftarrow -\infty$
    $T \leftarrow \text{Considered-Modification}(\vec{\tau}, \vec{t})$
    for $\langle \vec{\tau}', \vec{t}' \rangle \in T$ **do**
        $\langle e', B' \rangle \leftarrow \text{Estimate-MSE-And-Budget}(\mathcal{M}, \vec{\tau}, \vec{t})$
        $i' \leftarrow \text{Estimate-Improvement}(e - e', B - B')$
        if $i' > i$ **then**
            $\langle i, \vec{\tau}, \vec{t} \rangle \leftarrow \langle i', \vec{\tau}', \vec{t}' \rangle$
$\mathcal{S} \leftarrow \text{Select-Salient-Channels}(\mathcal{M}, \vec{\tau})$
Return $\mathcal{S}, \vec{t}$

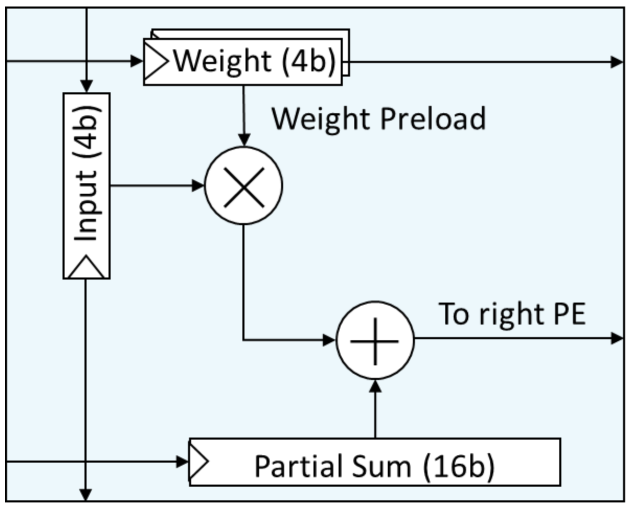
Architecture Design

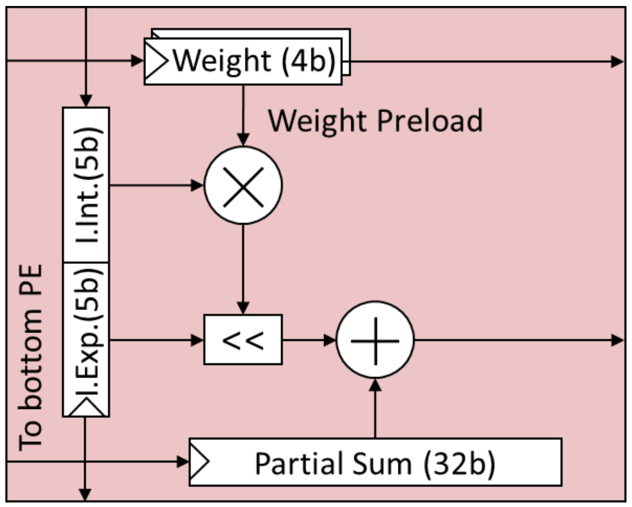
- Reconfigurable to support various TOBE settings
- Efficient by leveraging TOBE regularity



Hybrid PE Design

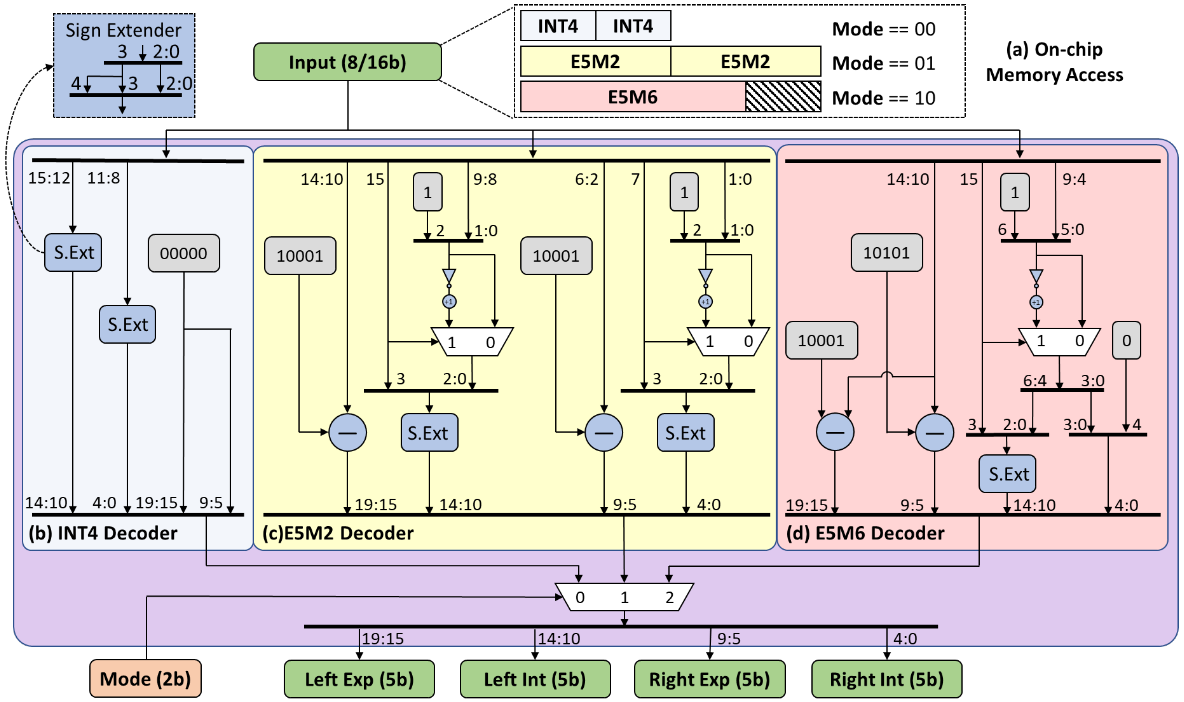
- Flexible PE modification
  - Support both uint4 and sint4 input
  - Add 5-bit input exponent and shifter
  - Augment partial sum bits to 32
- Functionality
  - int4×int4 with 1 efficient/flexible PE
  - int4×fp8 (E5M2) with 1 flexible PE
  - int4×fp12 (E5M6) with 2 flexible PEs





Decoder Design

- Functionality
  - Convert different types of input data into unified exponent-integer pairs
  - The decoding mode can be reconfigured during runtime by the controller



Experiment Setup

- Quantization Setup
  - Models: LLaMA 7-65B, OPT 6.7-66B
  - Perplexity evaluation: WikiText2, C4
  - Calibration data: 128 sequences randomly sampled from WikiText2
  - Weight quantization: GPTQ (4-bit)
- Baselines
  - Quantization: OmniQuant, OliVe
  - Accelerator: OLAccel, OliVe
- Architecture Implementation
  - Performance simulation: DnnWeaver
  - PE /decoder power & area: Synopsys DC and TSMC-28nm PDK
  - Memory power & area: CACTI
  - Process scaling: DeepScaleTool

Architecture	PE Number	Core Area	Buffer
Oltron	4096	0.322 mm^2	512 KB 4.2 mm^2
OliVe	2048	0.318 mm^2	
OLAccel	1152	0.320 mm^2	

Accuracy Result

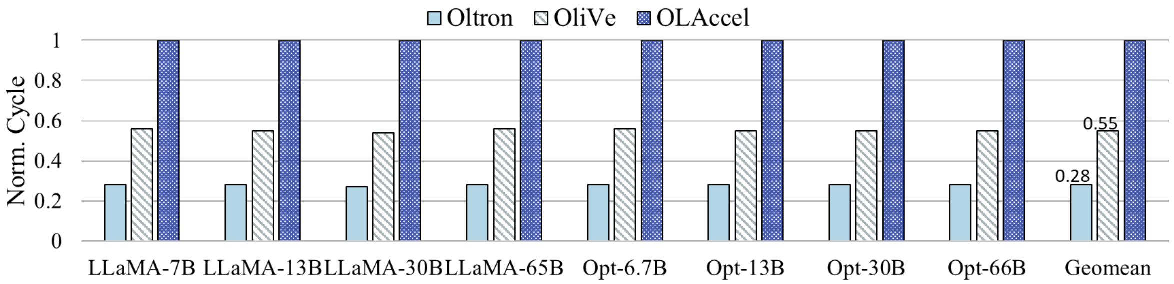
- Perplexity results (↓)
  - Oltron outperforms existing methods OliVe and Omniquant on most models
- Ablation study with adaptive quantization algorithm
  - Consistently better than uniform TOBE configuration (Oltron*)

Model/PPL↓		LLaMA-7B		LLaMA-13B		LLaMA-30B		LLaMA-65B		OPT-6.7B		OPT-13B		OPT-30B		OPT-66B	
Method	A bits	WIKI	C4	WIKI	C4	WIKI	C4	WIKI	C4	WIKI	C4	WIKI	C4	WIKI	C4	WIKI	C4
FP16	16	5.68	7.08	5.09	6.61	4.10	5.98	3.53	5.62	10.86	11.74	10.12	11.19	9.56	10.69	9.34	10.28
Olive	4	144.78	117.49	42.24	43.13	36.55	33.78	1.4e7	1.8e7	107.15	61.24	416.57	994.64	334.7	572.02	4058.83	2926.87
Omniquant	4	11.26	14.51	10.87	13.78	10.33	12.49	9.17	11.28	12.24	13.56	11.65	13.46	10.60	11.89	10.29	11.35
Oltron*	4.01	126.81	92.50	185.50	170.75	164.97	357.00	30.61	45.73	16.63	16.26	13.41	14.66	11.20	12.68	151.48	190.71
Oltron	4.01	36.47	44.62	144.08	100.18	439.25	131.15	15.85	20.85	12.69	13.58	**11.49**	**12.53**	10.72	**11.87**	11.61	11.71
Oltron*	4.1	14.47	16.80	**9.48**	**12.42**	**7.51**	**9.42**	**6.69**	**9.41**	**11.99**	**13.04**	**11.61**	**12.42**	10.64	**11.67**	10.50	**11.09**
Oltron	4.1	11.67	15.21	**8.20**	**10.84**	**6.68**	**8.65**	**5.82**	**8.19**	**12.00**	**13.02**	**11.35**	**12.27**	**10.51**	**11.63**	10.49	**11.05**

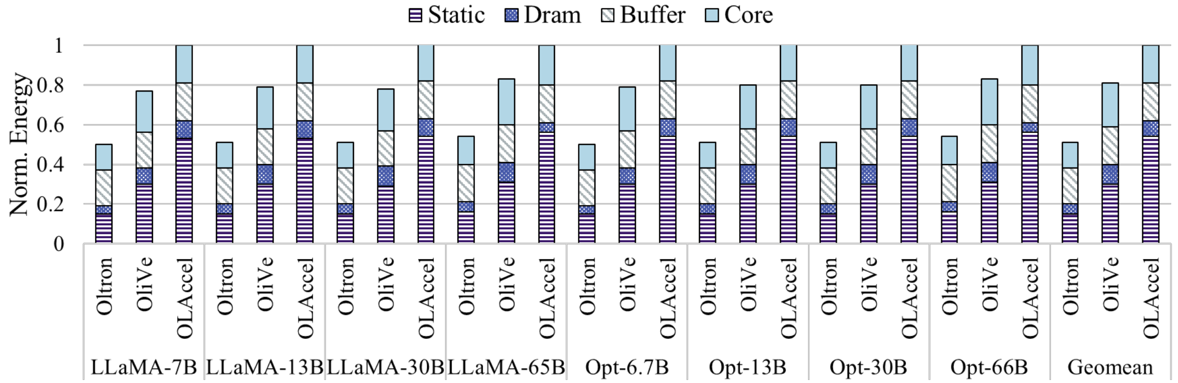
* Use the same salient channel configuration across all layers.

Accelerator PPA Result

- Performance result
  - 1.9× speedup over OliVe
  - 3.6× speedup over OLAccel



- Energy result
  - 1.6× energy reduction over OliVe
  - 1.9× energy reduction over OLAccel



Thank you for listening!

Contact:

xch927027@pku.edu.cn

chenzhang.sjtu@sjtu.edu.cn

gsun@pku.edu.cn