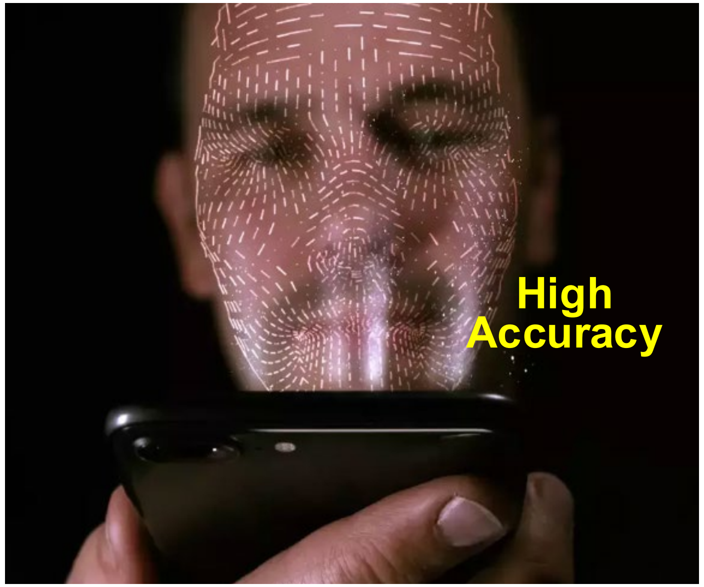
DSPU: A 281.6mW Real-Time Deep Learning-Based Dense RGB-D Data Acquisition with Sensor Fusion and 3D Perception System-on-Chip

Dongseok Im, Gwangtae Park, Zhiyong Li, Junha Ryu, Sanghoon Kang, Donghyeon Han, Jinsu Lee, Wonhoon Park, Hankyul Kown, and Hoi-Jun Yoo

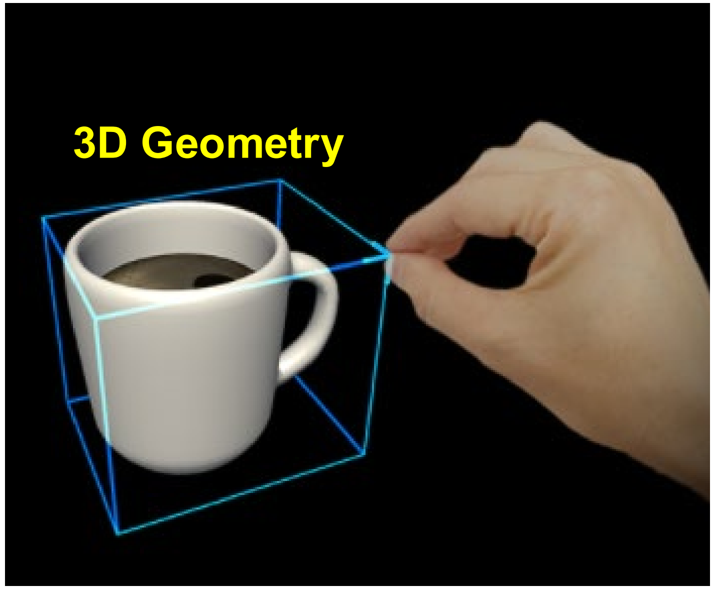
Semiconductor System Lab.
School of EE, KAIST

3D Data in Mobile Platforms

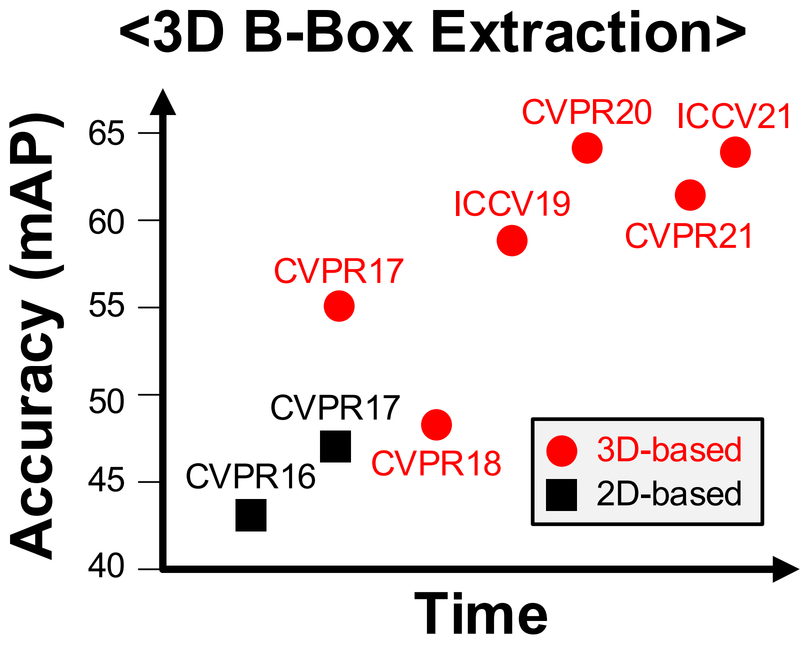
- **RGB-D data → More Accurate and Versatile Applications**
  – CNN recognizes only 2D pictures, but real world consists of 3D objects
  – RGB-D (3D) data enables the exact 3D object recognitions



Face Recognition

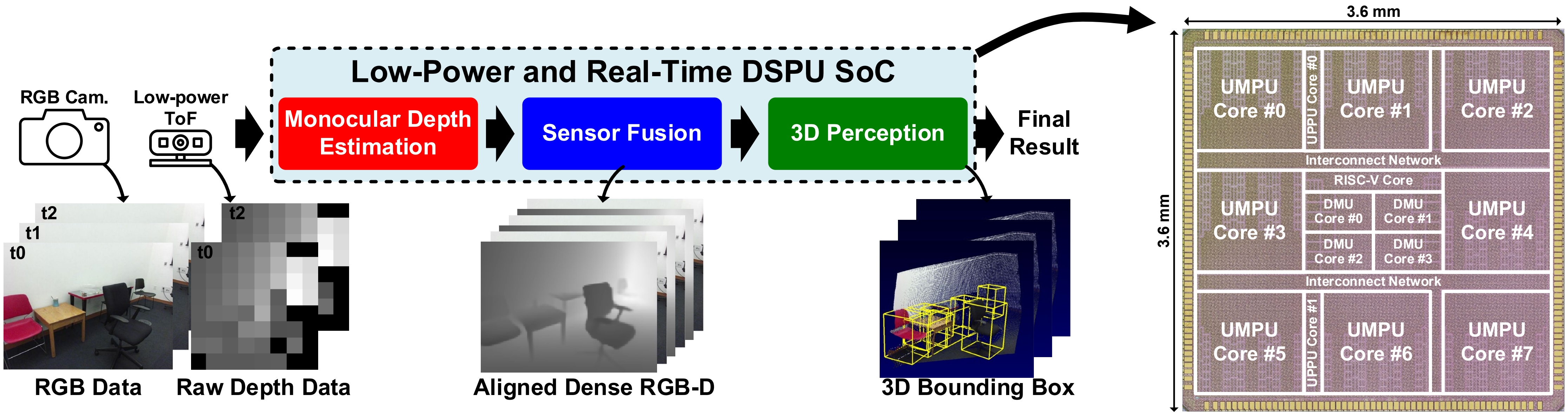


AR/VR



DSPU: End-to-end 3D Perception SoC

- **A 281 mW and 31.9 fps 3D Object Recognition Processor**

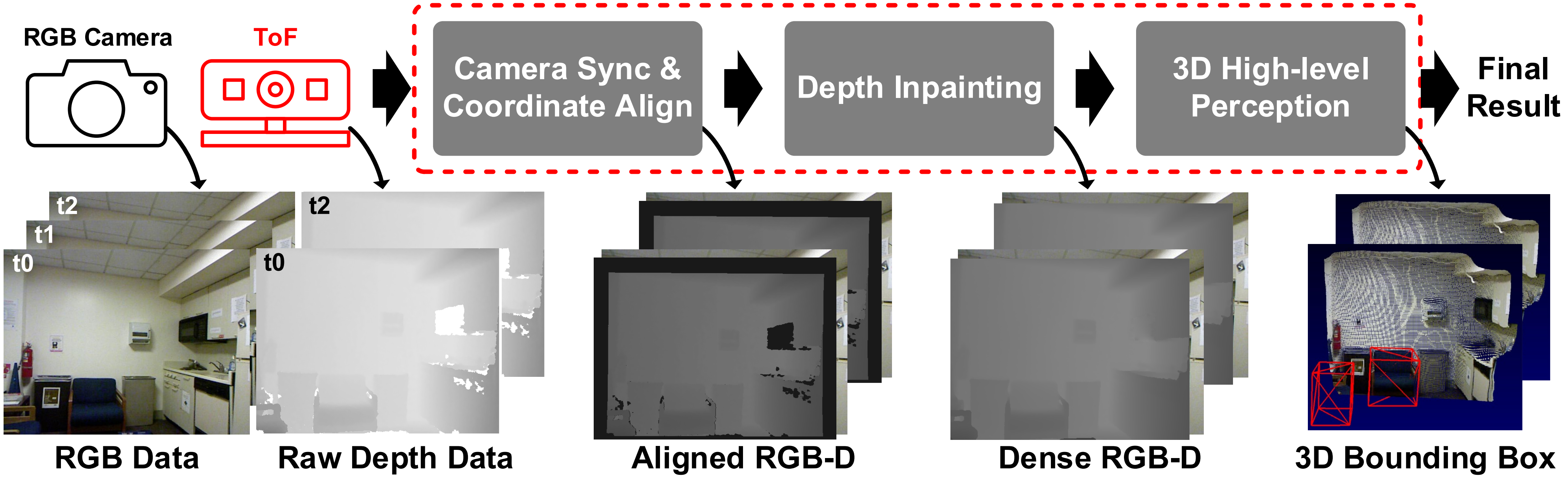


  - **<u>For Low-Power RGB-D Data Acquisition</u>**
    ➔ CNN-based MDE & Sensor Fusion SW/HW Architecture
  - **<u>For Real-time 3D Perception (e.g. 3D Bounding Box)</u>**
    ➔ Window-based Search & Point Feature Reuse SW/HW Architecture

1) MDE: Monocular Depth Estimation

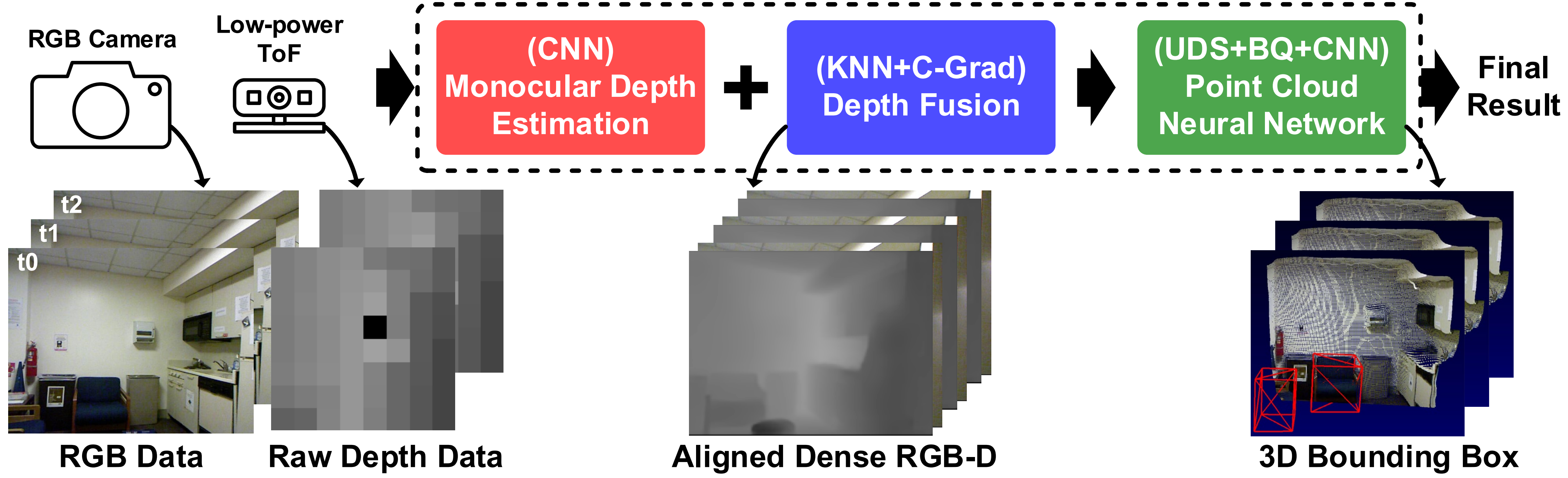
Challenges of 3D Perception

- **Power and Latency Challenges in Mobile Platforms**
  - High sensor power (>3 W)
  - High latency in CPU+GPU Platform (~10 fps)



Proposed End-to-end 3D Perception

1. CNN-based MDE for Low-Power Dense RGB-D Acquisition
2. Sensor Fusion for Accurate RGB-D Data
3. Window Search-based PNN for Low-Latency 3D Perception

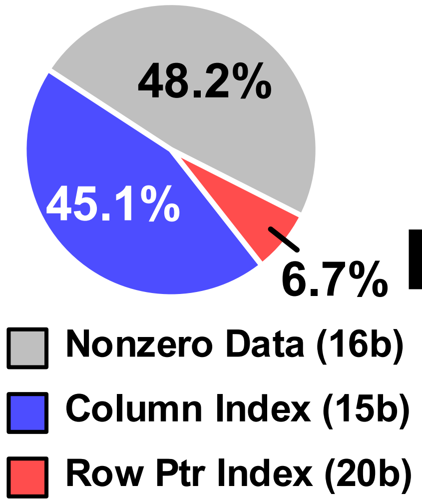


1) LP ToF: Low Power ToF, 2) CNN: Convolutional Neural Network, 3) KNN: K-nearest neighbor search, 4) C-Grad: Conjugate-gradient, 5) UDS: Uniform Distance Point Sampling, 6) BQ: Ball Query

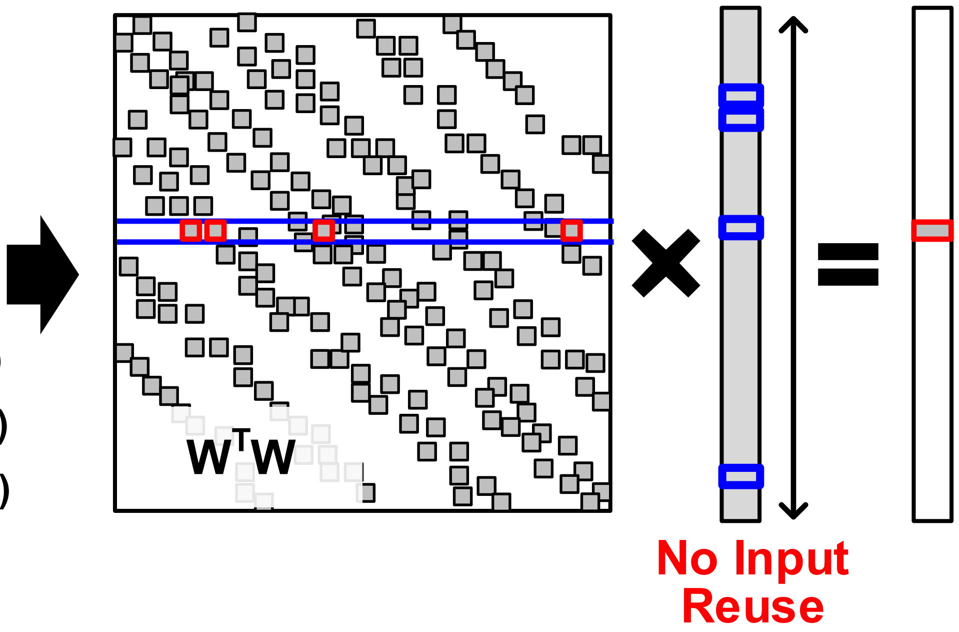
Challenges of Sensor Fusion

- **Irregular Sparse Matrix generated by KNN**
  - CSR produces 'Data + Index', but still large data size (1.86 MB)
  - SpMM & SpMV result in many data transactions due to low data reuse

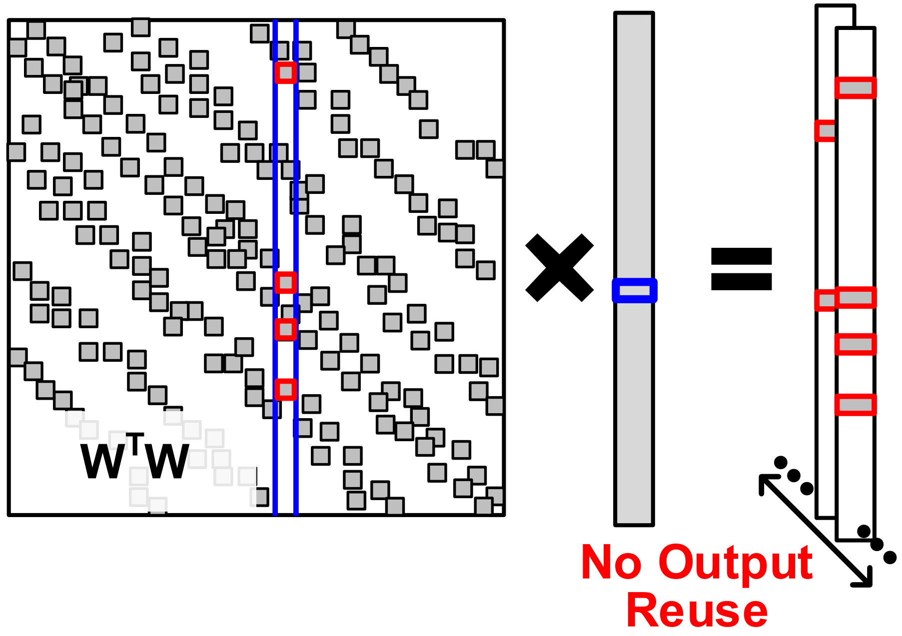
Encoded Data Size Breakdown



Method 1: Inner-Product



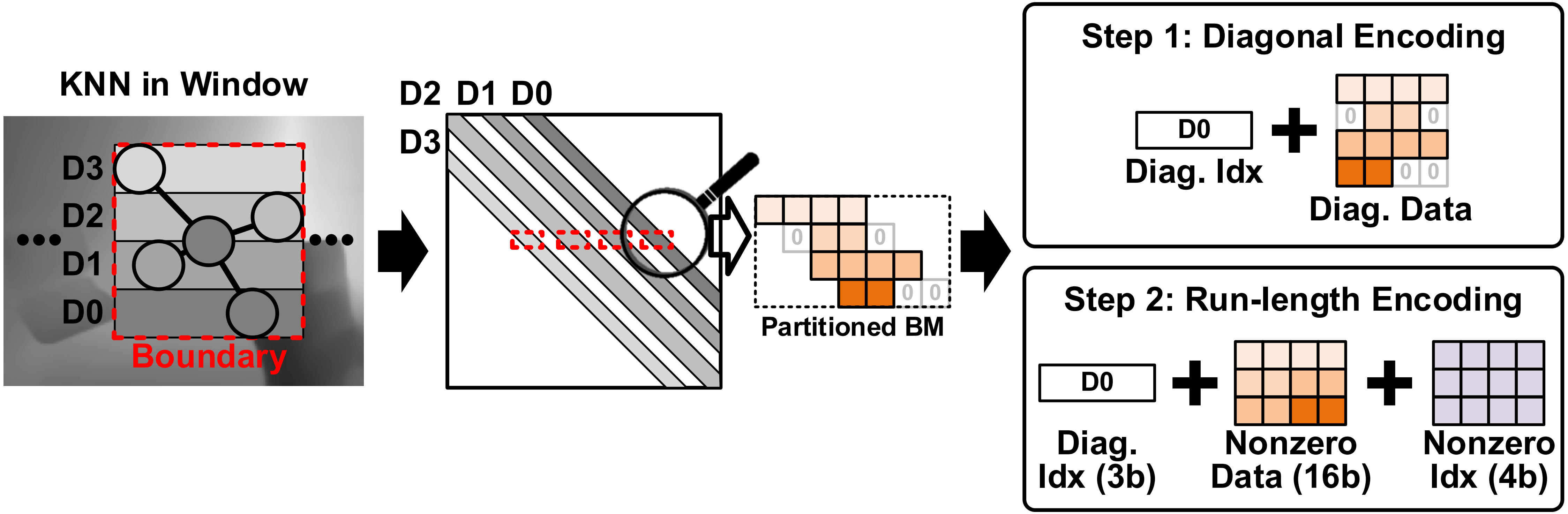
Method 2: Outer-Product



1) CSR: Compressed Sparse Row

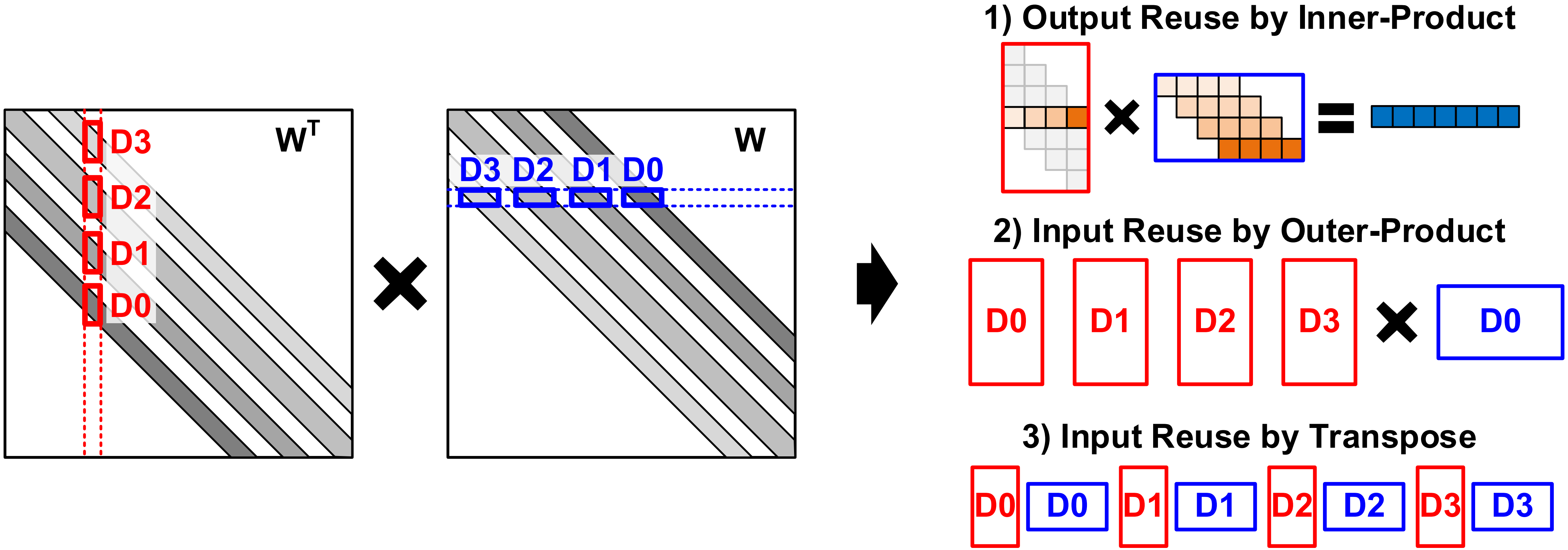
Band Matrix Encoding

- **Diagonal BM generated by Window Search-based KNN**
  – Hierarchical BM encoding produces 'Diagonal Index + Data + Small Index'
  ➔ Increase the data compression ratio



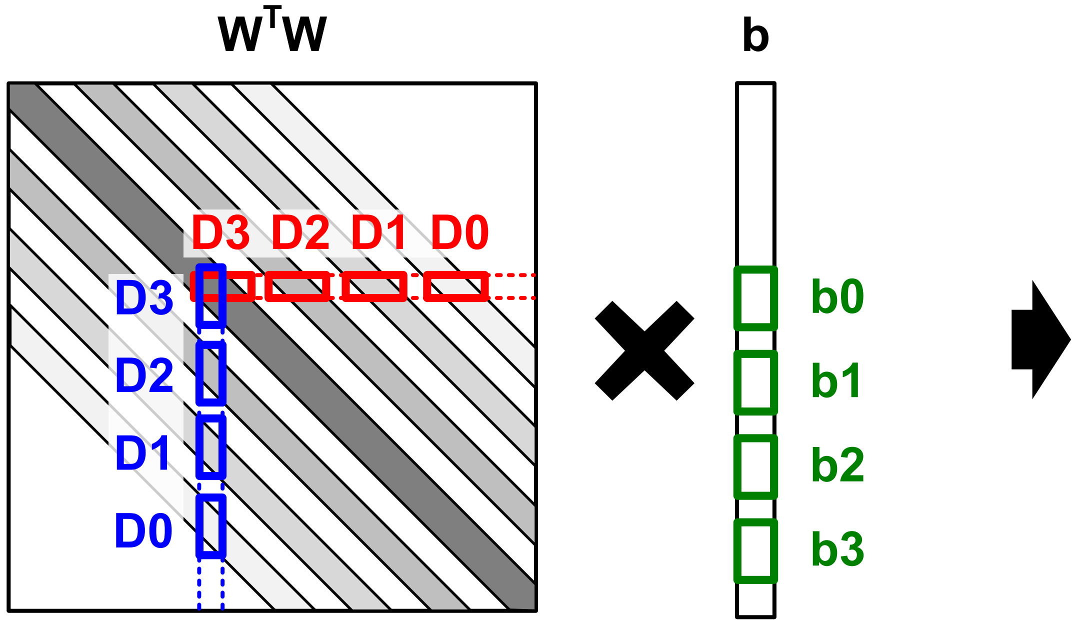
Band Matrix Decoding for SpMM

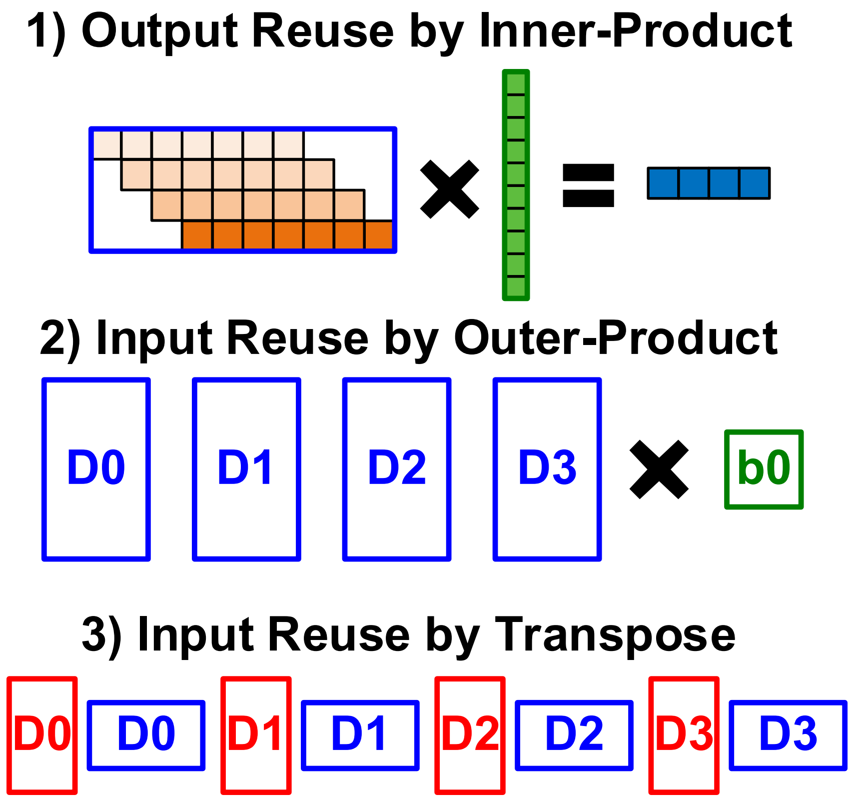
- **Simultaneous W^T & W Computation**
  - Increase both input data and output data reuse
  - ➔ Reduce the number of data transaction



Band Matrix Decoding for SpMV

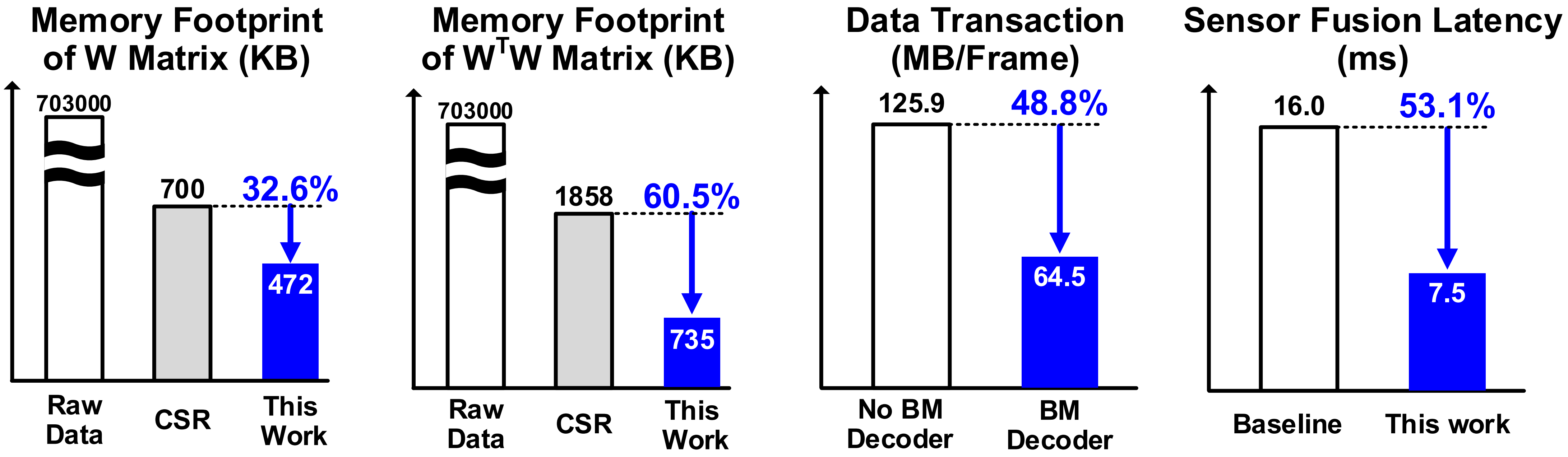
- **Simultaneous Lower & Upper Triangle of W^TW Computation**
  - – Increase both input data and output data reuse
  - ➔ Reduce the number of data transaction





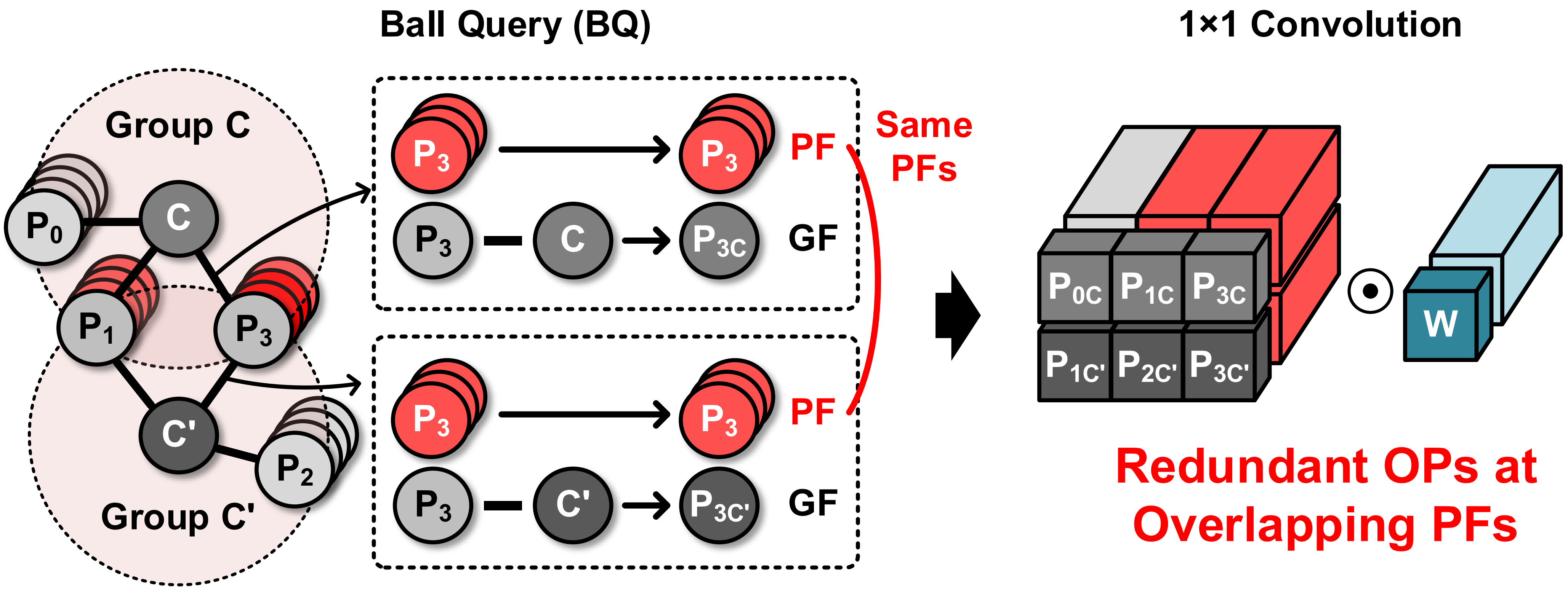
Performance of BM Codec

- **Reduction of Memory Footprint and Data Transactions**
  - BM encoding-decoding increases the speed of sensor fusion



Redundant Operations in PNN

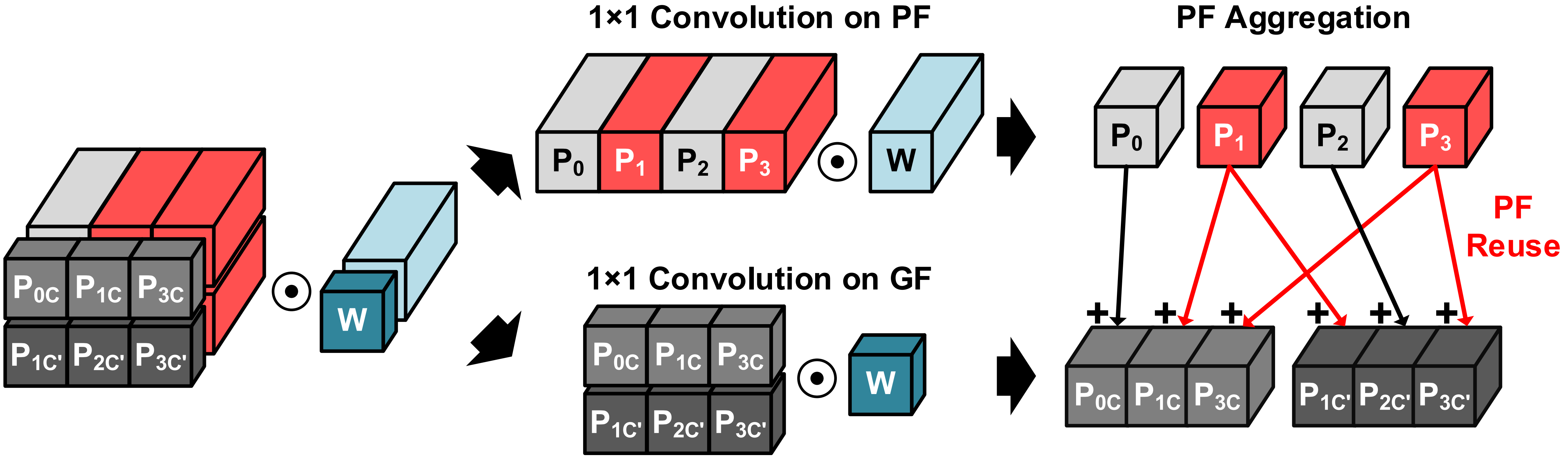
- **Redundant Convolution OPs at Overlapping Neighbors**
  – Average 50% of neighbors are overlapped after BQ
  ➔ Their point features cause the redundant convolution OPs



1) PF: Point Feature, 2) GF: Group Feature

Point Feature Reuse

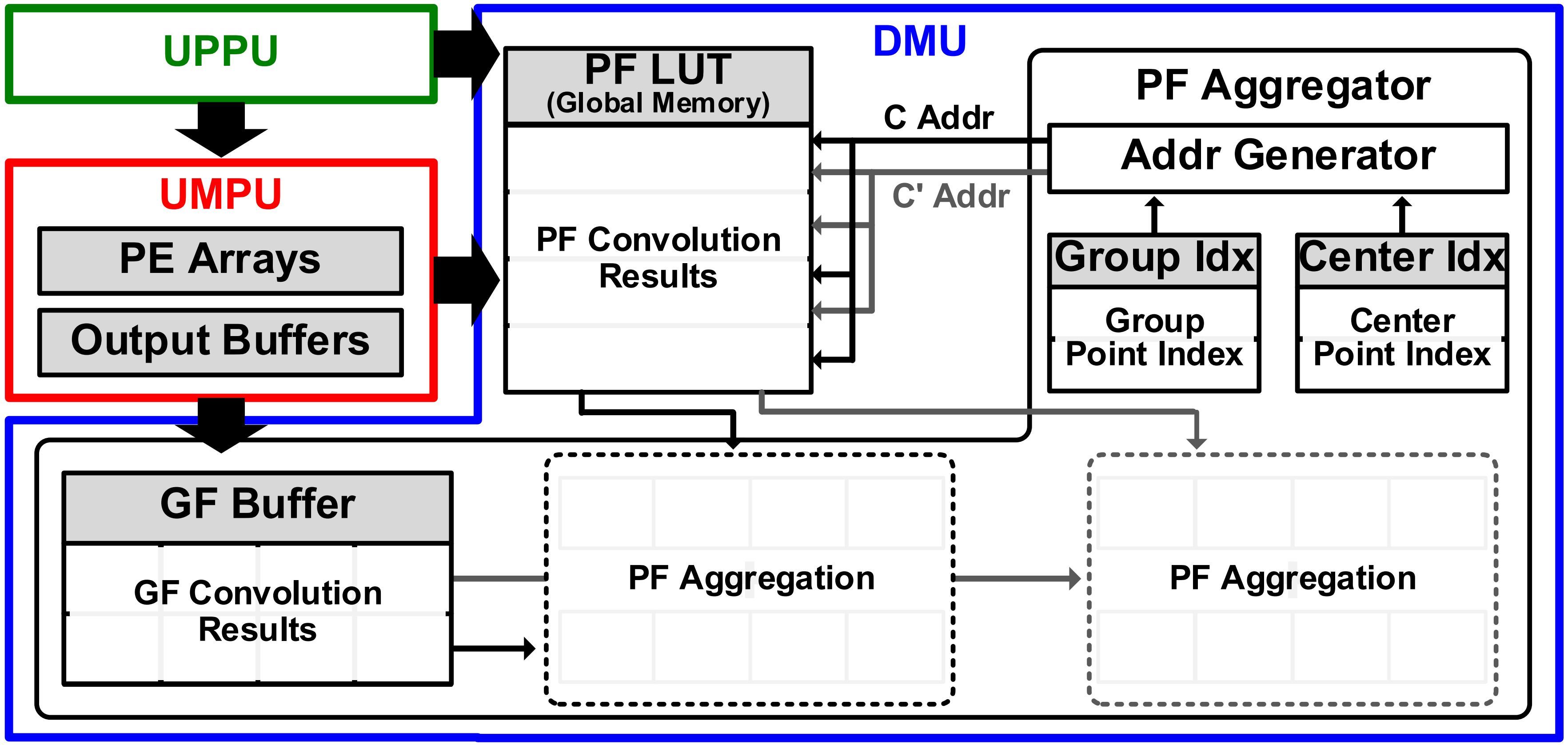
- **Computational Reuse at Overlapping Point Features**
  - Execute the convolution on PFs and GFs separately
  - Reuse the PF convolution results by aggregating corresponding GF results



1) PF: Point Feature, 2) GF: Group Feature

Pipelined Architecture

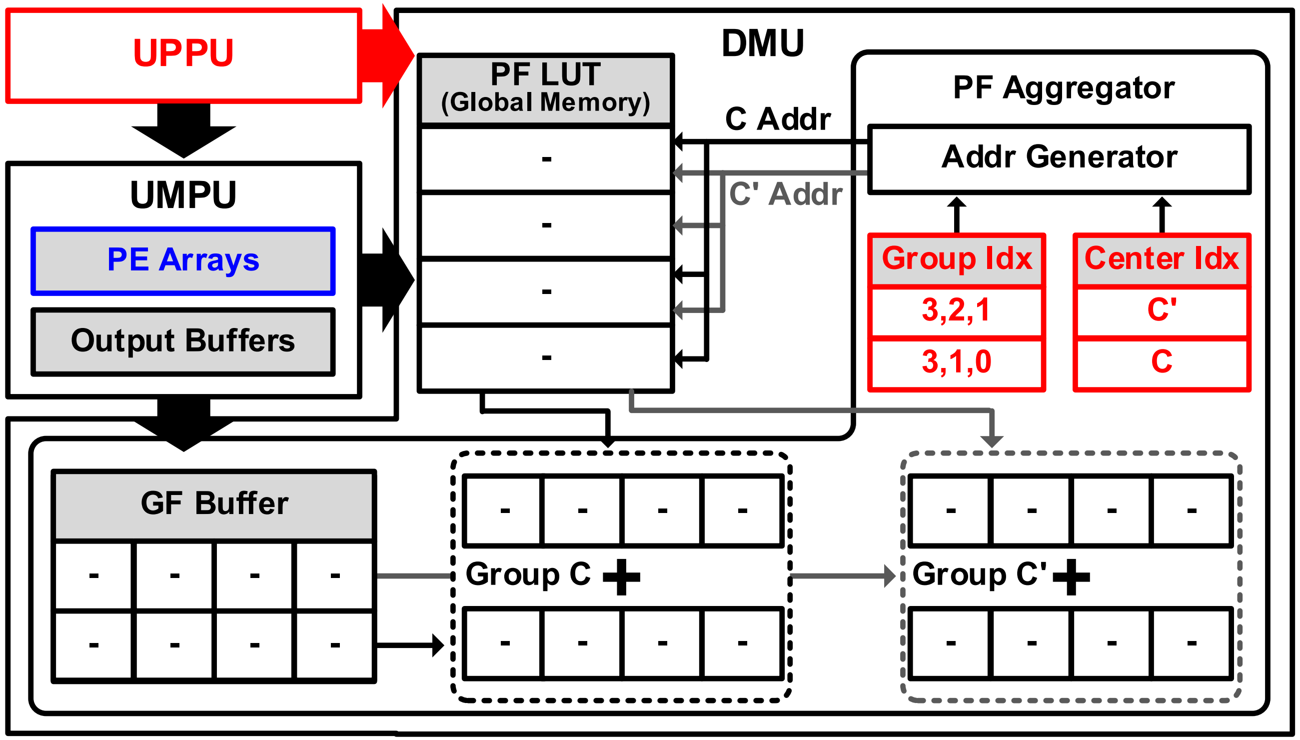
- **Point Feature Reuse with the UPPU, UMPU, and DMU**
  - Pipelined architecture hides the processing time of each HW unit

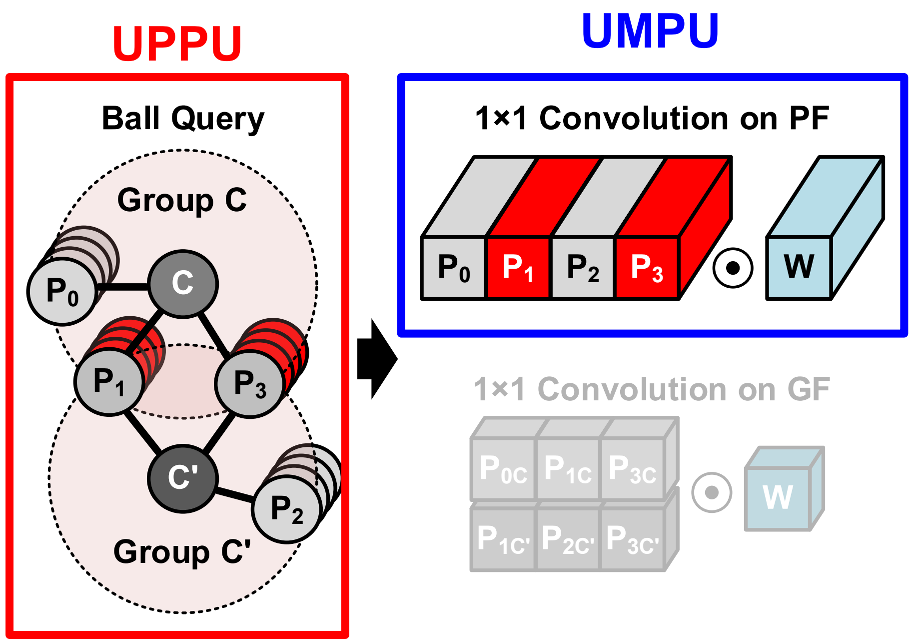


1) UPPU: Unified Point Processing Unit, 2) UMPU: Unified Matrix Processing Unit, 3) DMU: Data Management Unit

Pipelined Architecture

- **Simultaneous Convolution and Ball Query Operations**
  - UPPU performs the BQ on 3D point data
  - UMPU computes the convolution on PFs of all 3D point data

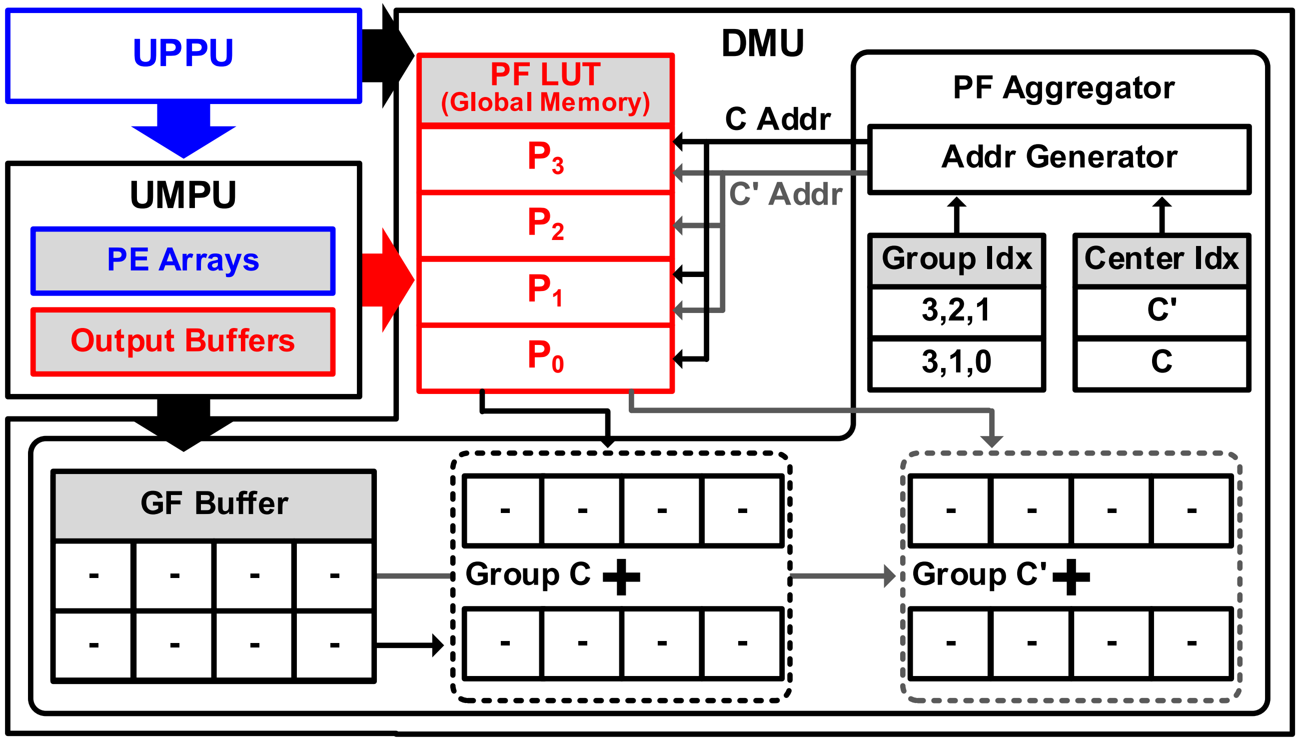


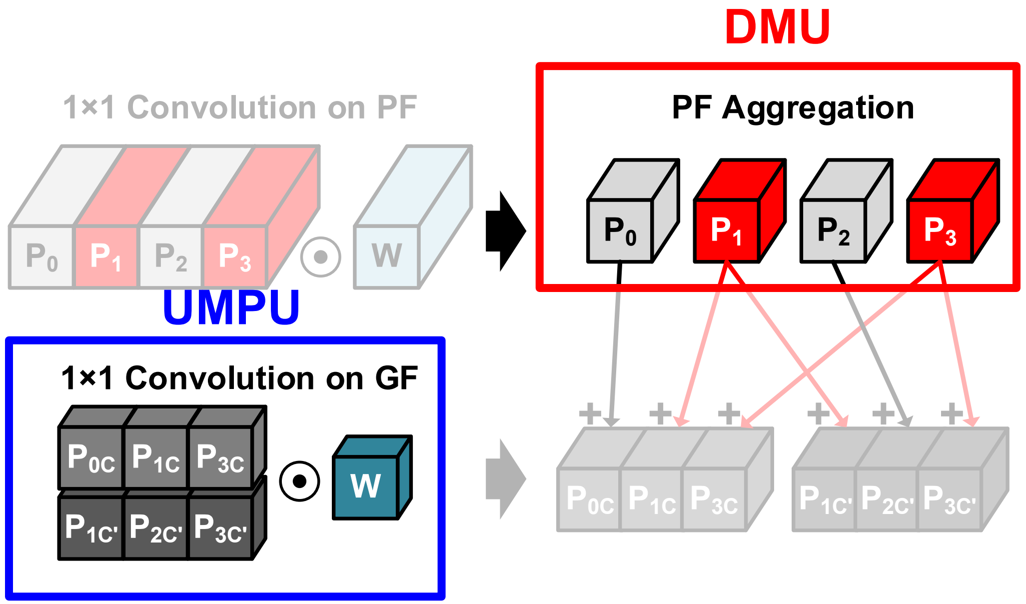


1) UPPU: Unified Point Processing Unit, 2) UMPU: Unified Matrix Processing Unit, 3) DMU: Data Management Unit

Pipelined Architecture

- **Simultaneous Convolution and PF LUT Update**
  - UMPU computes the convolution on GFs
  - PF LUT is updated by new PF convolution results

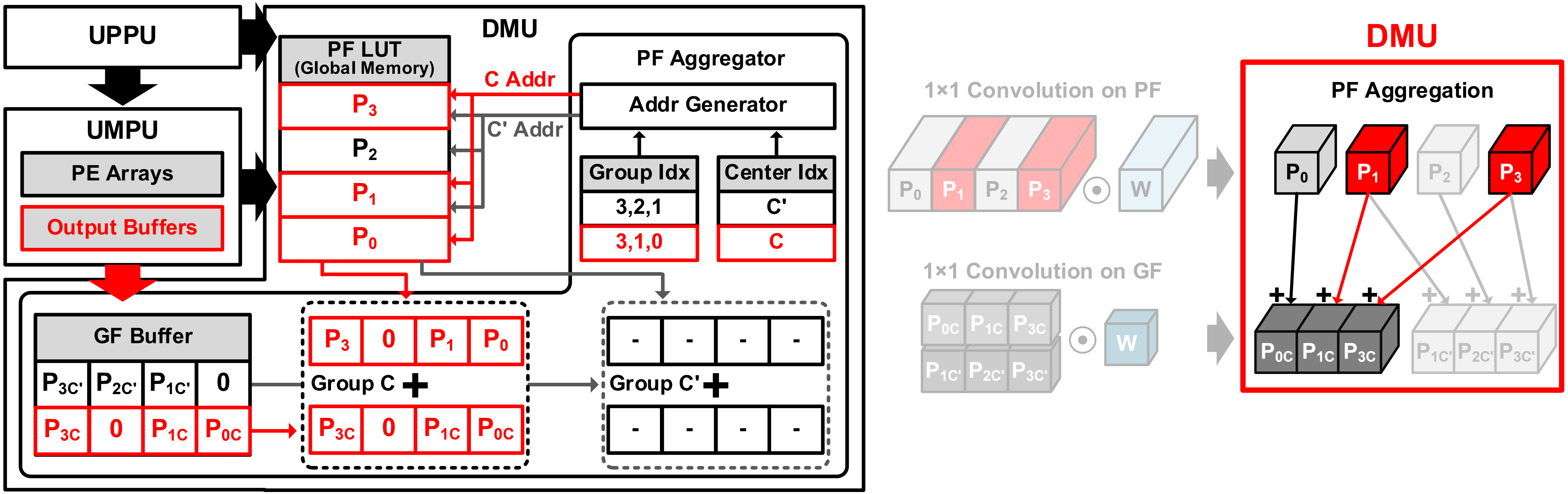




1) UPPU: Unified Point Processing Unit, 2) UMPU: Unified Matrix Processing Unit, 3) DMU: Data Management Unit

Pipelined Architecture

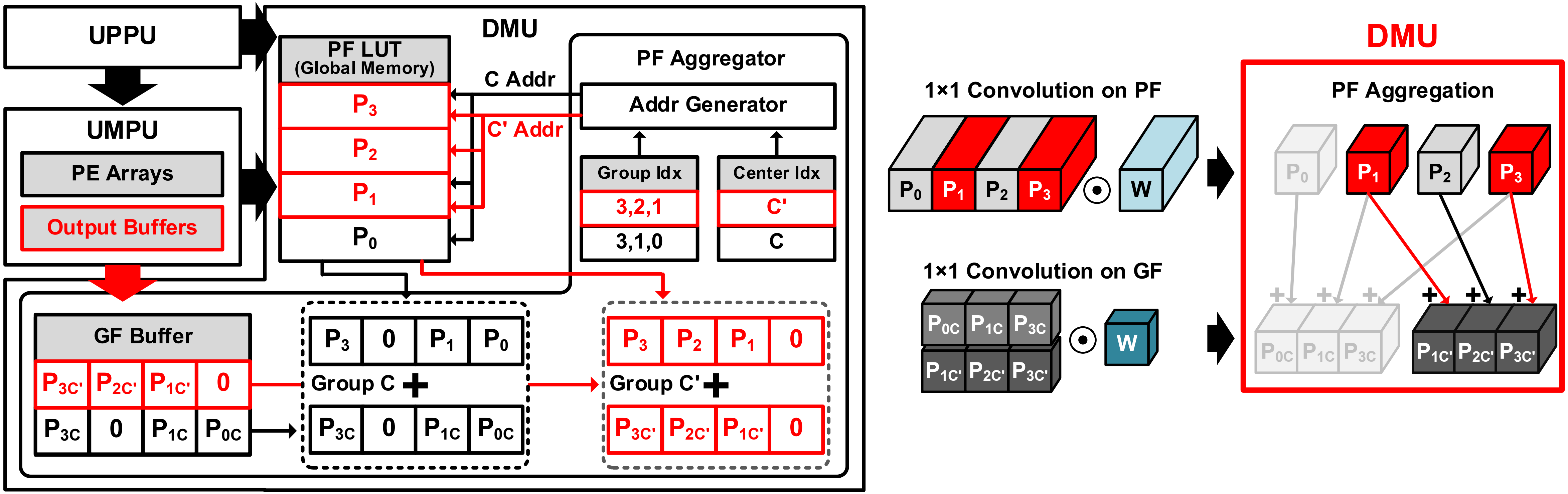
- **PF Aggregation on the Group C**
  - P_0, P_1, and P_3 are loaded from PF LUT by the address generator, and summed up with P_{0C}, P_{1C}, and P_{3C}



1) UPPU: Unified Point Processing Unit, 2) UMPU: Unified Matrix Processing Unit, 3) DMU: Data Management Unit

Pipelined Architecture

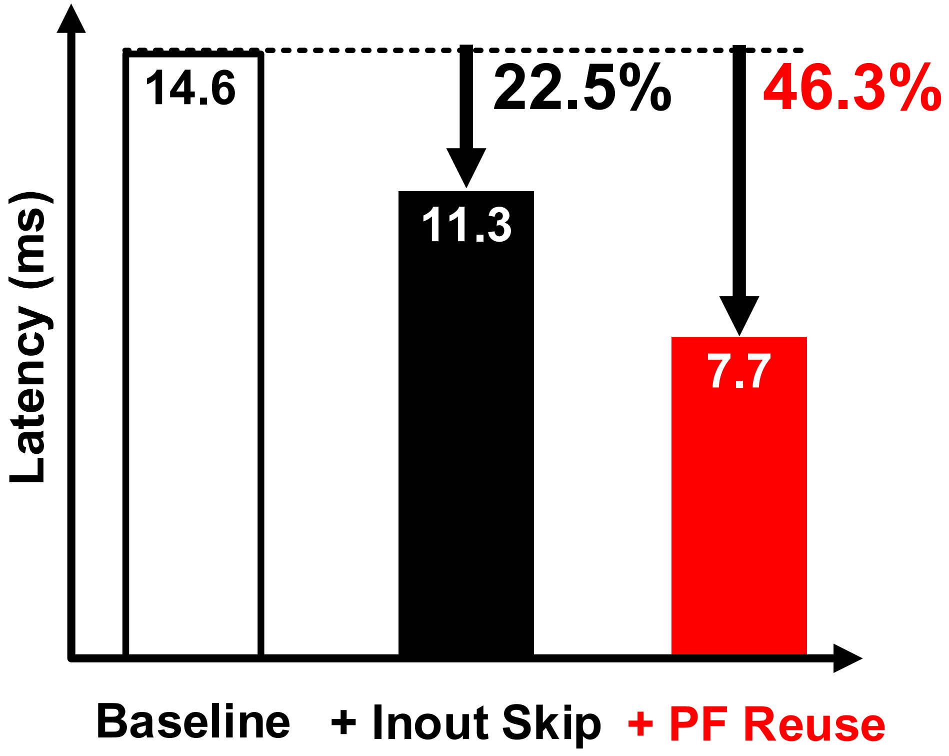
- **PF Aggregation on the Group C**
  - – P_1, P_2, and P_3 are loaded from PF LUT by the address generator, and summed up with $P_{1C'}$, $P_{2C'}$, and $P_{3C'}$

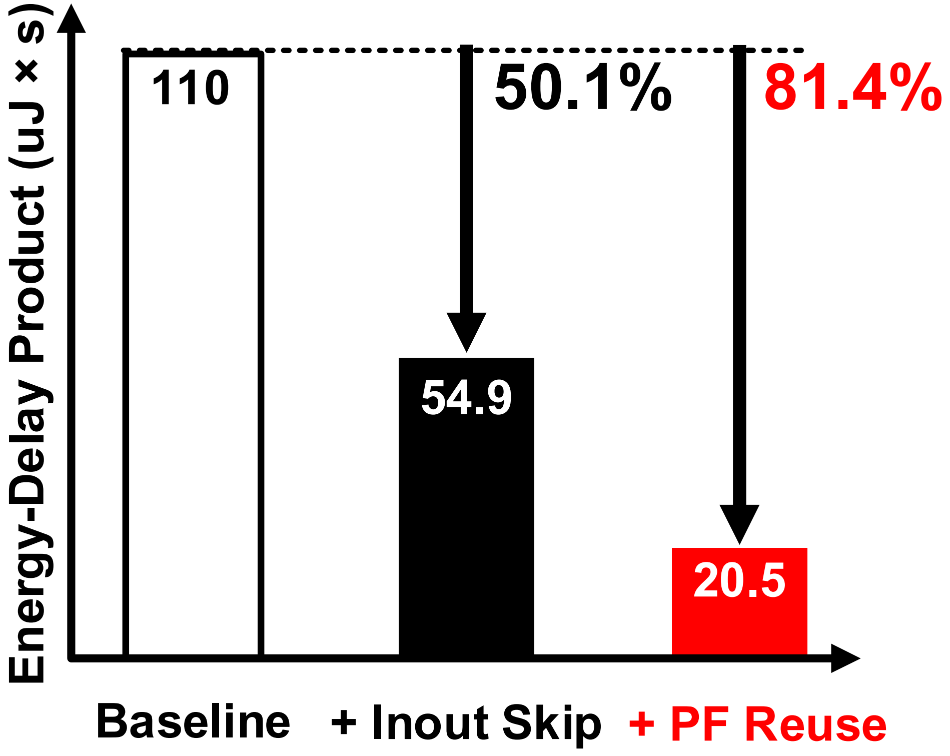


1) UPPU: Unified Point Processing Unit, 2) UMPU: Unified Matrix Processing Unit, 3) DMU: Data Management Unit

PNN Performance

- **Performance Improvement with Pipelined Architecture @ VoteNet**





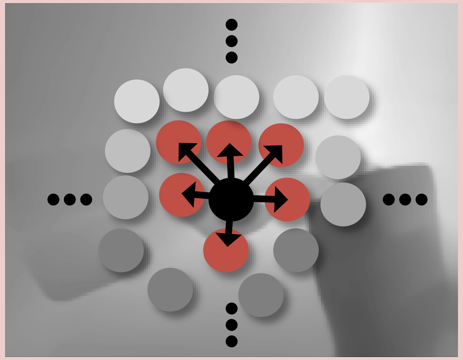
Challenges of Point Processing

- **Different Operations between Point Processing Algorithms**
  – Dedicated HW units are required

➔ The area overhead of HW units increases

Neighbor Search in Depth Fusion

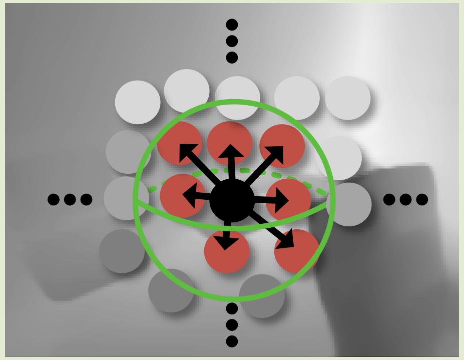
KNN Unit



Group Points,
s.t. *topK(d, K)*

Point Grouping in PNN

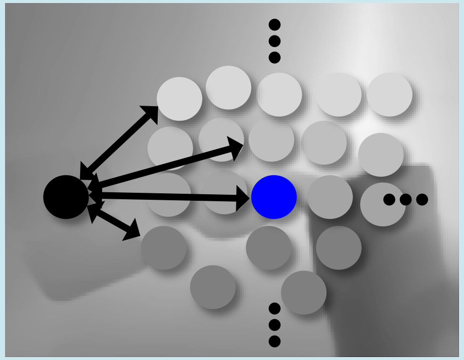
BQ Unit



Group Points,
s.t. *d* < *R*

Point Sampling in PNN

UDS Unit

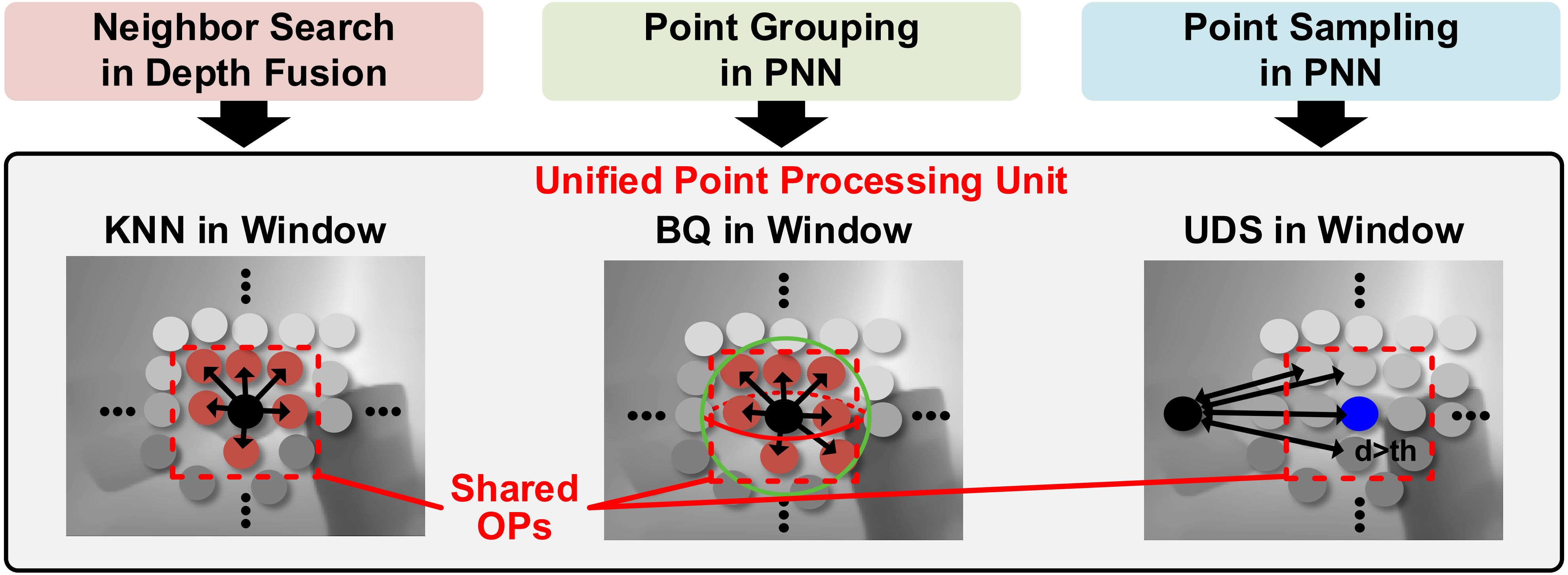


Sample Points,
s.t. *d* > *theshold*

1) PNN: Point Cloud-based Neural Network 2) KNN: K-nearest neighbor search, 3) BQ: Ball Query, 4) UDS: Uniform Distance Point Sampling

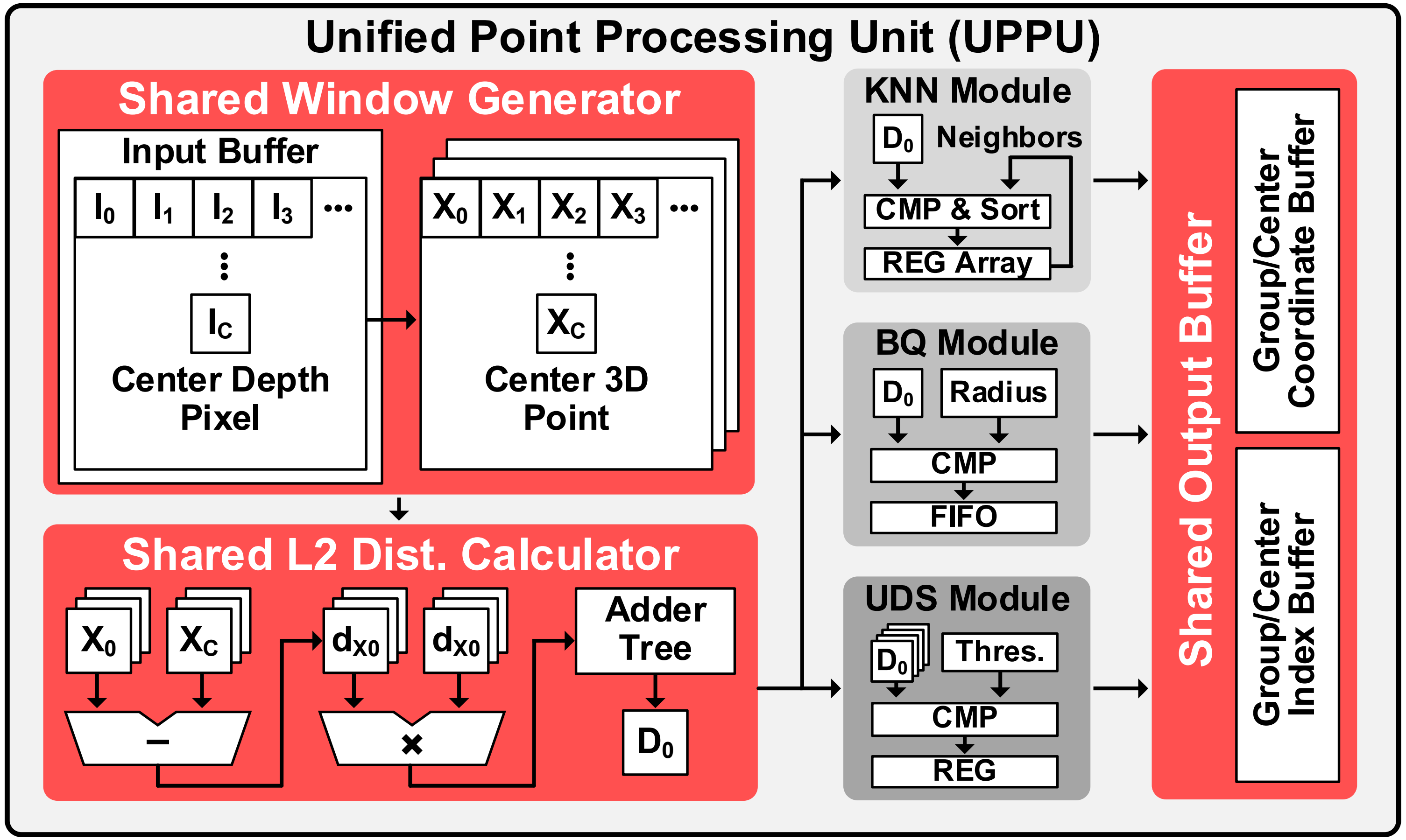
Window Search-based Point Processing

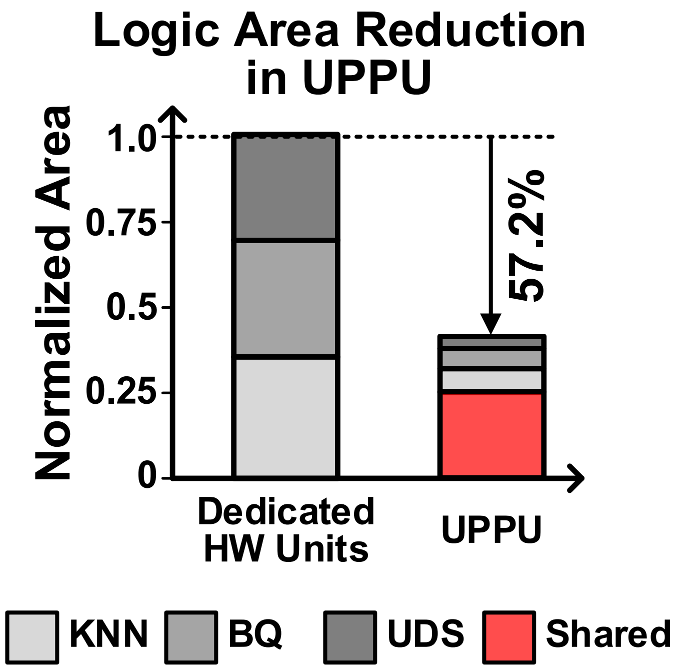
- **Point Processing within the Predefined Window**
  - Number of operations can be reduced largely
  - The different point processing algorithms can share “operations”, e.g., window generation, L2 distance computation, load/store block data



Unified Point Processing Unit

- **Area Saving by Sharing Common Logic and Buffer**
  - Hardware units for the window-based search and output buffers are shared

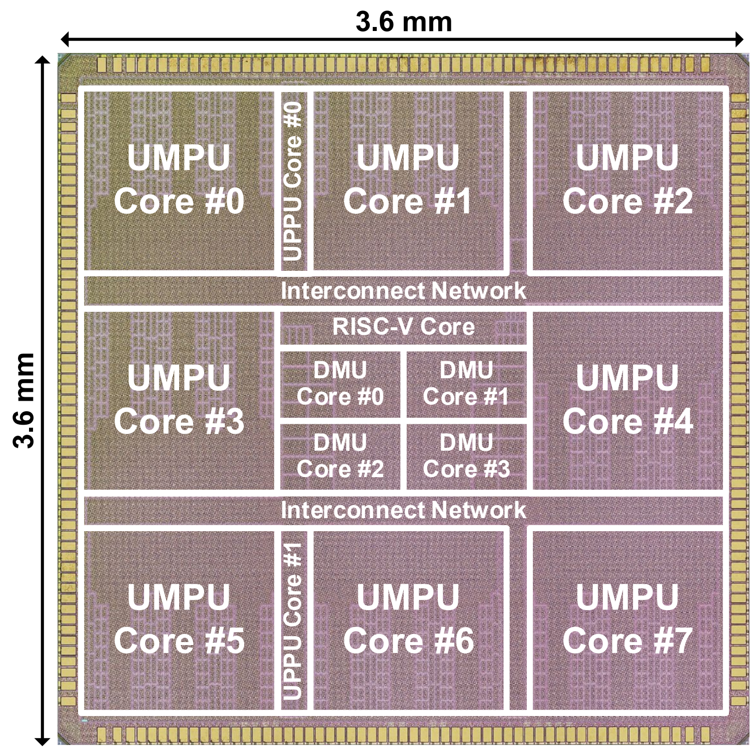




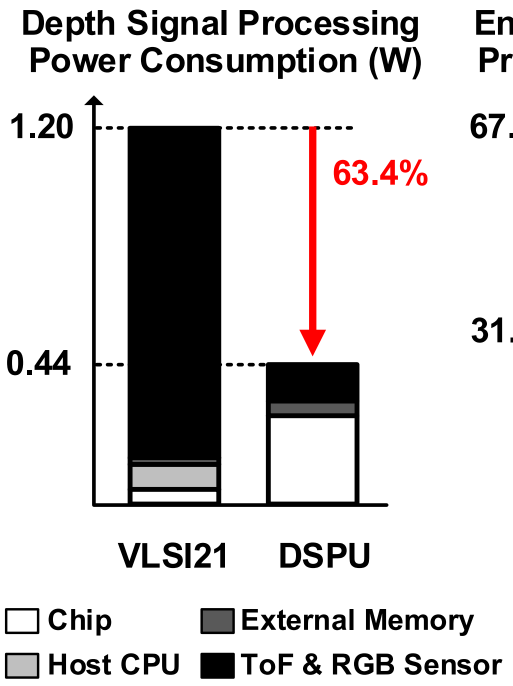
1) PNN: Point Cloud-based Neural Network 2) KNN: K-nearest neighbor search, 3) BQ: Ball Query, 4) UDS: Uniform Distance Point Sampling

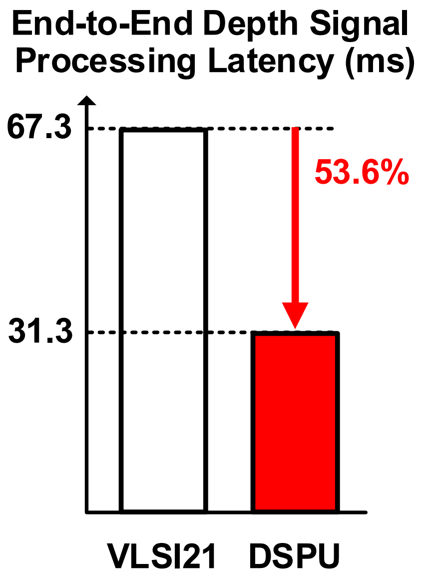
Chip Photography and Summary

- **64.4% Lower Power Consumption than Previous System**
- **53.6% Lower Latency than Previous System**



	Specifications
Technology	Samsung 28 nm
Die Area	12.96 mm^2
SRAM	806 KB
ISA	RISC-V
Supply Voltage	0.72-1.1 V
Max. Frequency	250 MHz
UMPU Performance	
Peak Throughput [TOPS]	4.5 @ Depth CNN (8b) 1.8 @ Depth CNN (12b) 0.1 @ C-Grad (16b) 11.6 @ Point CNN (8b)
Power [mW]	544.7 @ Depth CNN (8b) 640.9 @ Depth CNN (12b) 545.3 @ C-Grad (16b) 609.1 @ Point CNN (8b)
UPPU Performance	
Throughput [TOPS]	1.1 @ Point Grouping 0.3 @ Point Sampling
Power [mW]	25.1 @ Point Grouping 23.0 @ Point Sampling

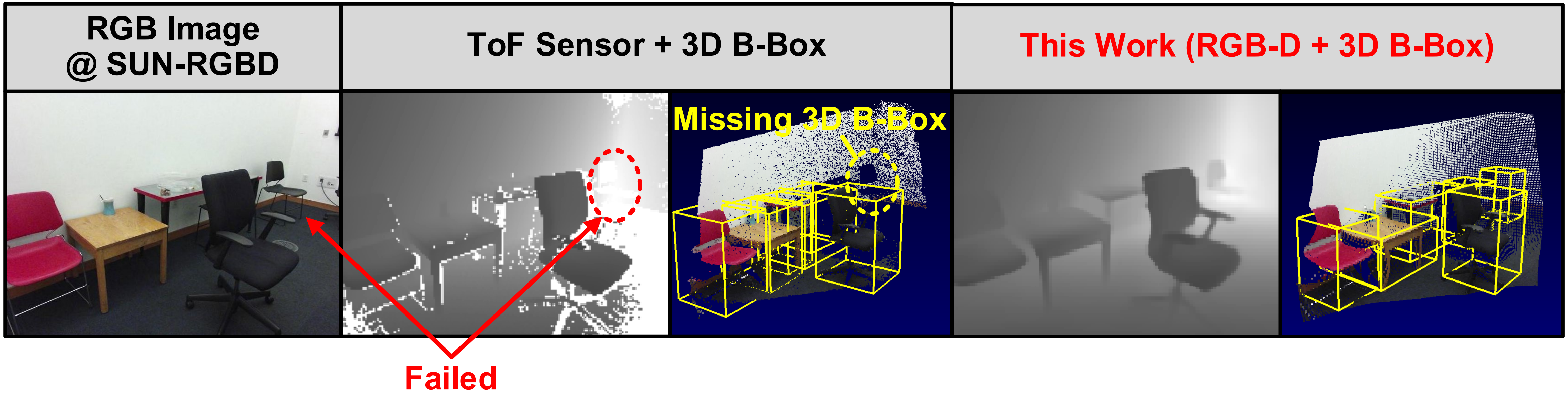




1) VLSI21 System: S.Kim's ASIC (VLSI21) + Host CPU + External Memory + RGB-D Sensor

Measurement Results

- **Visual Results of 3D B-Box Extraction**
  - ToF Sensor cannot capture a chair in the back
    → Fail to extract the 3D bounding-box (B-Box)
  - This work detects all of objects



Conclusion

- **DSPU: Low-power and Real-Time 3D Object Recognition SoC**

- **For Low-power and Real-Time 3D Object Recognition**
  - ***BM Encoder and Decoder for Low Latency***
  - ***PF Reuse with Pipelined Architecture for Low Latency and Energy***
  - ***Shared Unified Point Processing Unit for High Reconfigurability***

A 281.6 mW and 31.9 fps Dense RGB-D Acquisition and PNN 3-D Recognition Processor for Mobile 3-D Vision

Thank You!

- **Questions? Feel Free to Contact Me!**
  - E-mail: dsim@kaist.ac.kr
  - LinkedIn: https://www.linkedin.com/in/dongseok-im-b05007216/