# Clo-HDnn: Continual On-Device Learning Accelerator with Hyperdimensional Computing via Progressive Search

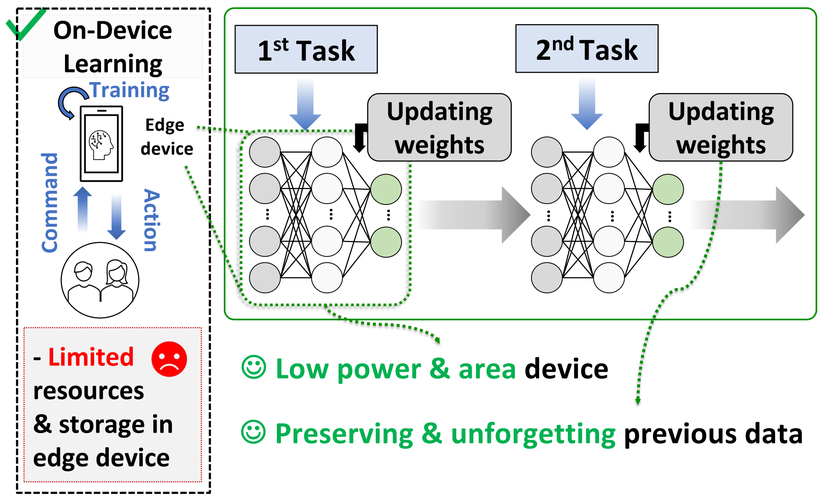

Chang Eun Song[1], Weihong Xu[1], Keming Fan[1], Soumil Jain[1], Gopabandhu Hota[1], Haichao Yang[1], Leo Liu[2], Meng-Fan Chang[2], Carlos H. Diaz[2], Gert Cauwenberghs[1], Tajana Rosing[1], and Mingu Kang[1]

[1] University of California San Diego, La Jolla, CA, USA [2]Taiwan Semiconductor Manufacturing Company (TSMC)


VVIP Lab @UCSD

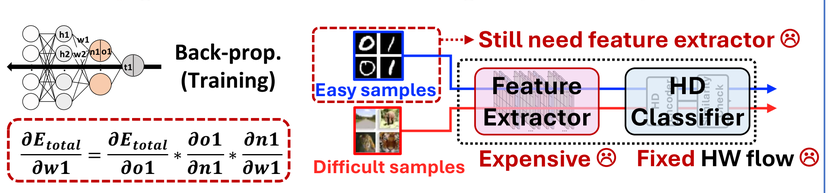
## Continual On-Device Learning for Edge AI

- Continual learning (CL) is essential for dynamic edge environments
- Edge device requires low-power, compact AI accelerators

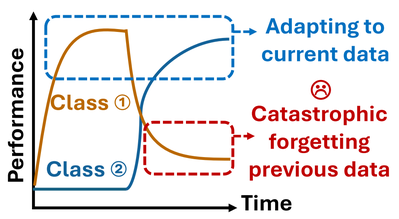



### Challenges for edge device & continual learning

- **Costly** gradient-based training & E2E computation

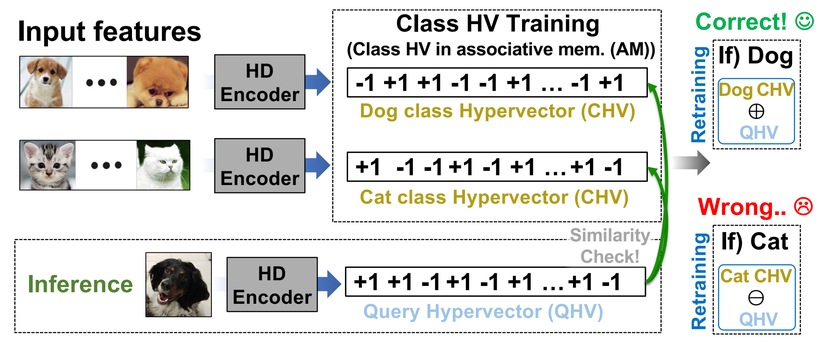



- **Non-supporting** continual learning due to catastrophic forgetting



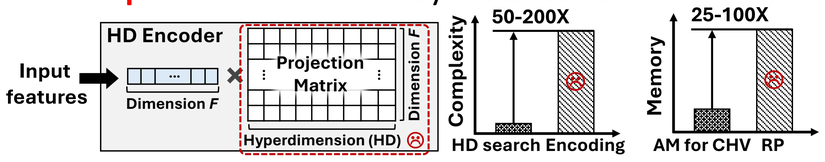

## Hyperdimensional Computing (HDC)

- A brain-inspired, memory-efficient framework
- Using hyperdimensional vectors (HVs) to support robust and incremental learning
- Fast inference & light computation for training

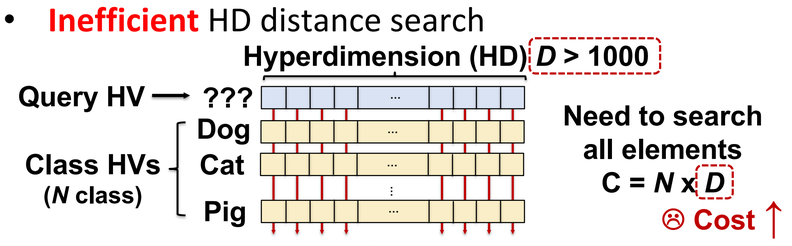



### Challenges of hyperdimensional computing

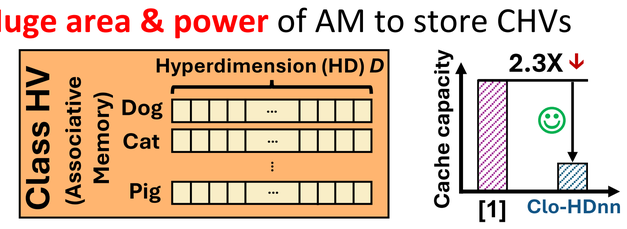
- **Computation overhead** by HD encoder




- **Inefficient** HD distance search

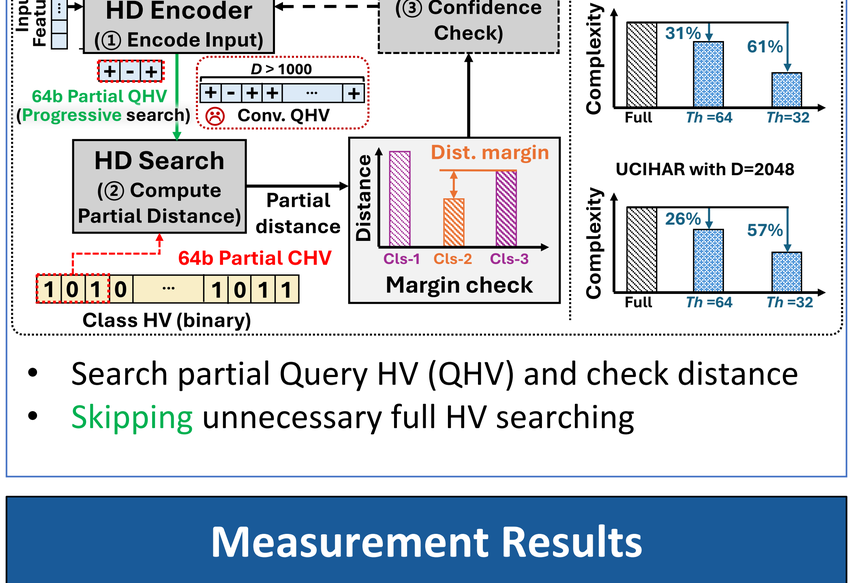


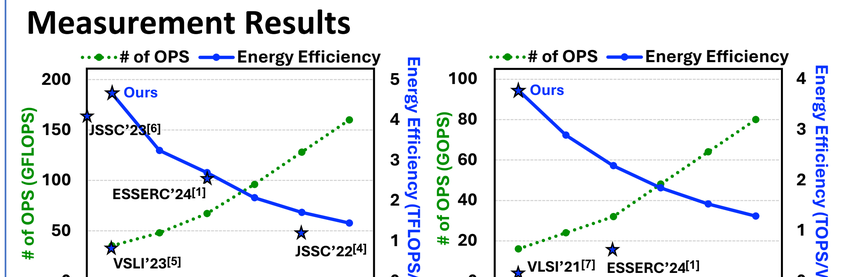

- **Huge area & power** of AM to store CHVs




## Proposed Clo-HDnn

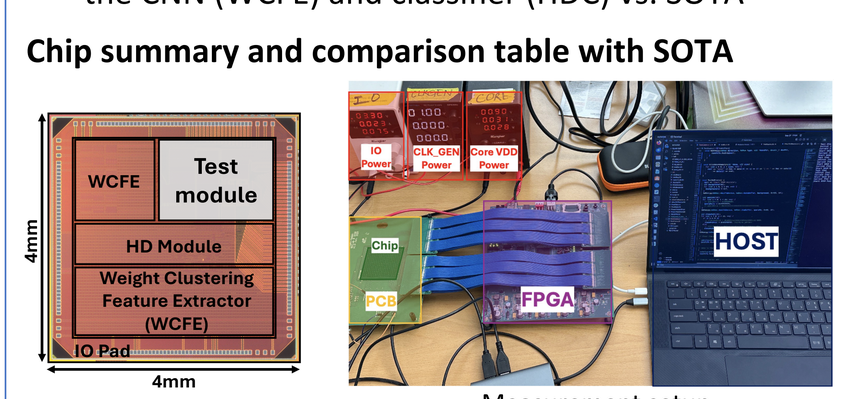
### 1) Dual mode processing




- Supporting reconfigurable hardware for dual mode
- Bypassing expensive weight clustering feature extractor (WCFE) for easy dataset to save power

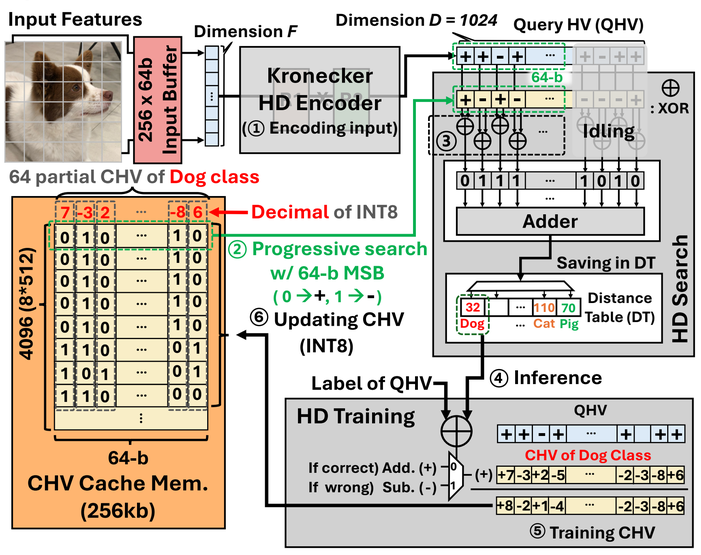
### 2) Progressive HD Distance Search



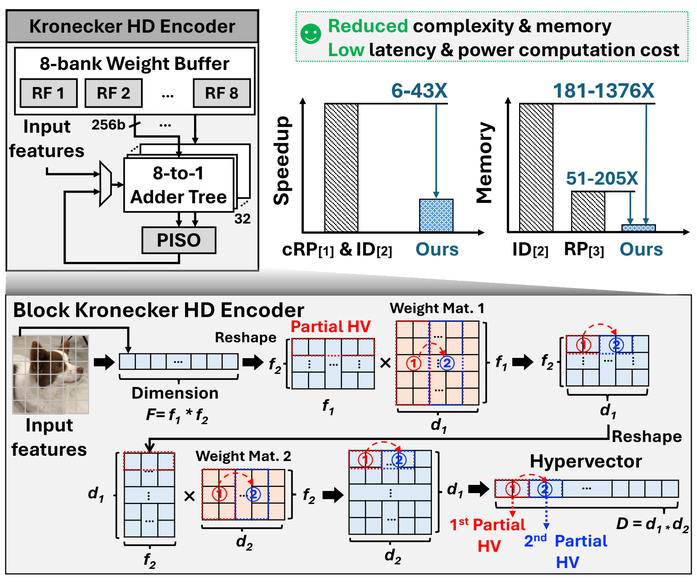

- Search partial Query HV (QHV) and check distance
- Skipping unnecessary full HV searching

## Key Hardware Blocks

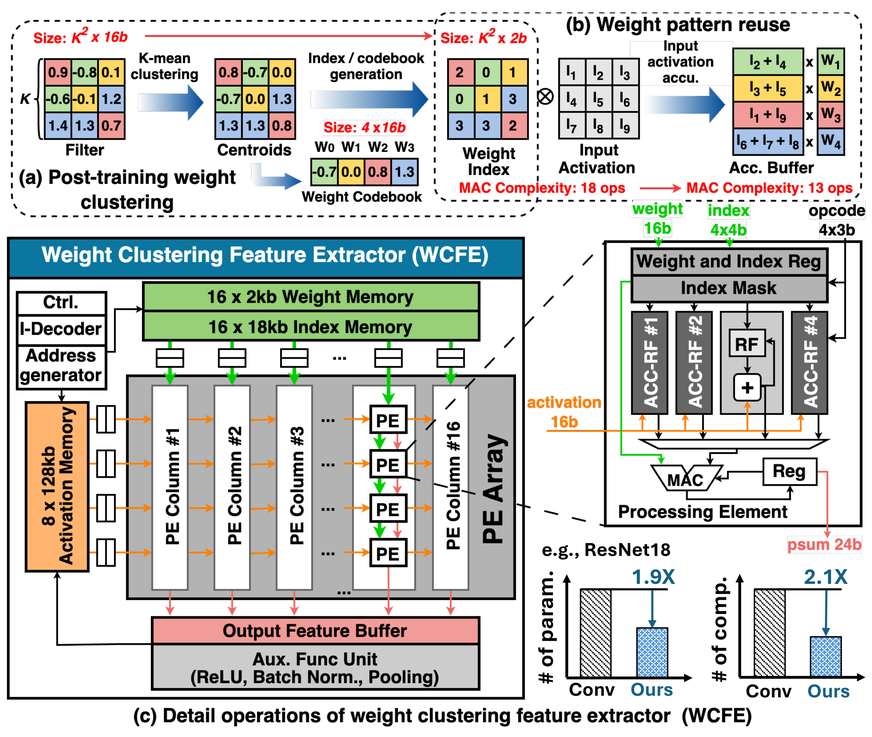
### 1) HD Module




### 2) Kronecker HD Encoder




### 3) Weight Clustering Feature Extractor (WCFE)




(c) Detail operations of weight clustering feature extractor (WCFE)

1) **HD module** supports progressive search, allowing partial Class HV (CHV) storage in a **small cache**
2) **Kronecker HD encoder** uses partial matrix for **efficient** progressive search achieving **43×** speedup and **1376×** memory savings over SOTA
3) **WCFE** merges inputs sharing the same averaged weights, multiplying the weights once, **reducing parameters** by **1.9×** and computation by **2.1×** compared to SOTA.

## Measurement Results

### Measurement Results




- **1.73-7.77×** and **4.85×** higher energy efficiency for the CNN (WCFE) and classifier (HDC) vs. SOTA

### Chip summary and comparison table with SOTA




| | Our work | ESSERC'24 [1] | VLSI'23 [5] | JSSC'23 [6] | JSSC'22 [4] | VLSI'21 [7] |
|---|---|---|---|---|---|---|
| Technology | 40nm | 40nm | 28nm | 28nm | 40nm | 40nm |
| Learning Mode | CL HDC | FSL HDC | LET | Sparse BP | Low-rank BP | OSL |
| Design | Digital | Digital | Digital + CIM | Digital | Digital + CIM | ReRAM CIM |
| Encoder Type | Kronecker | cRP-based | - | - | - | - |
| Precision | BF16/INT1-8 | BF16/INT16 | BF16 | FP8/16 | INT8 | FP32 |
| On-chip Mem. (kB) | SRAM: 200 | SRAM: 424 | SRAM: 329 | SRAM: 1280 | ReRAM: 204 SRAM:512 | ReRAM: 8 |
| Area (mm²) | 14.4 | 11.3 | 5.8 | 16.4 | 29.2 | 0.2 |
| Frequency (MHz) | 50-250 | 100-250 | 20-450 | 75-340 | 200 | 200 |
| Supply voltage (V) | 0.7-1.2 | 0.9-1.2 | 0.56-1.05 | 0.6-1.1 | 1.1 | - |
| Scaled EE (TFLOPS/W) (CNN) | 4.66 @ResNet18 | 2.69 @VGG16 | 0.6-0.87 | 4.1@ ResNet20 | 1.1* @ResNet18 (2.2 TOPS/W) | - |
| Scaled EE (TOPS/W) (Classifier) | 3.78 (HDC) | 0.78 | - | - | - | 0.12 |

All the energy efficiency (EE) is scaled to 40nm technology, * Scaled INT8 (TOPS/W) to BF16 (TFLOPS/W)


### References

[1] H. Yang et al., ESSERC, 2024, pp. 33-36. [2] M. Imani et al., ICRC 2017 (pp. 1-8). IEEE. [3] A. Hernandez-Cano et al., DAC, 2021. p. 7-12 [4] K. Prabhu, et al., JSSC, vol. 57, no. 4, pp. 1013-1026, 2022. [5] J. -H. Kim, et al., Symp. on VLSI, 2023, pp. 1-2. [6] S. K. Venkataramanaiah et al., JSSC, vol. 58, no. 7, 2023. [7] H. Li et al., Symp. on VLSI, 2021, pp. 1-2.



### Acknowledgements

This work was supported by TSMC and in part by PRISM and CoCoSys, centers in JUMP 2.0, an SRC program sponsored by DARPA. #455140


Semiconductor Research Corporation

PRISM

**For more information, please visit https://www.ucsdvvip.com/, http://seelab.ucsd.edu/ or contact cesong@ucsd.edu**

Scan me for connection!

# Clo-HDnn: Continual On-Device Learning Accelerator with Hyperdimensional Computing via Progressive Search


Chang Eun Song[1], Weihong Xu[1], Keming Fan[1], Soumil Jain[1], Gopabandhu Hota[1], Haichao Yang[1], Leo Liu[2], Meng-Fan Chang[2], Carlos H. Diaz[2], Gert Cauwenberghs[1], Tajana Rosing[1], and Mingu Kang[1]

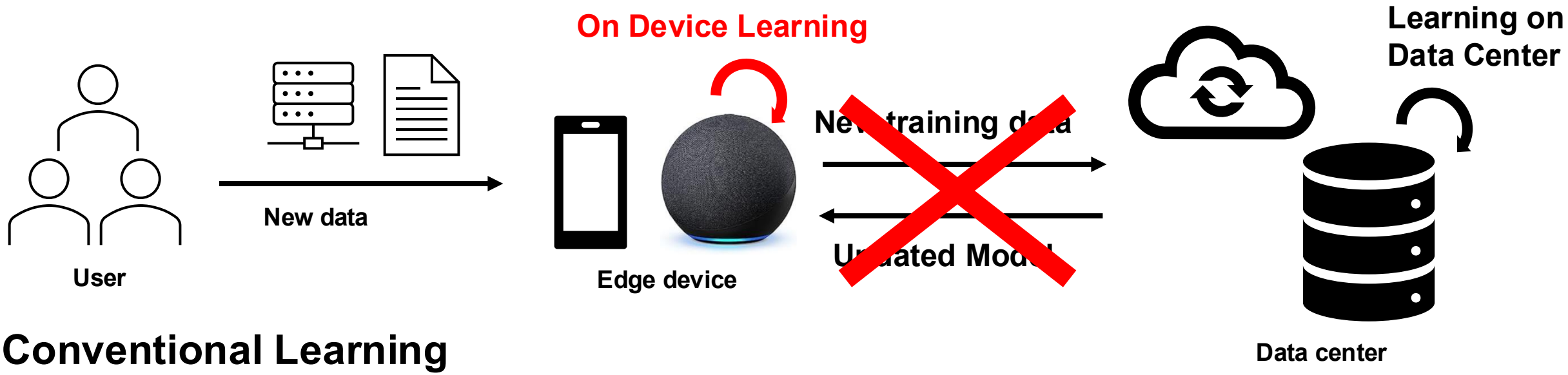
[1] University of California San Diego, La Jolla, CA, USA
[2] Taiwan Semiconductor Manufacturing Company (TSMC)

*VVIP Lab @UCSD*

# Abstract

Clo-HDnn is an **on-device learning (ODL) accelerator** designed for emerging continual learning (CL) tasks. Clo-HDnn integrates **hyperdimensional computing (HDC)** along with low-cost Kronecker HD Encoder and weight clustering feature extraction (WCFE) to optimize accuracy and efficiency. Clo-HDnn adopts gradient-free CL to efficiently update and store the learned knowledge in the form of class hypervectors. Its dual-mode operation enables bypassing costly feature ex- traction for simpler datasets, while progressive search reduces complexity by up to **61%** by encoding and comparing only partial query hypervectors. Achieving **4.66 TFLOPS/W (FE) and 3.78 TOPS/W (classifier)**, Clo-HDnn delivers **7.77× and 4.85× higher energy efficiency compared to SOTA ODL accelerators**.

# On Device Learning (ODL)



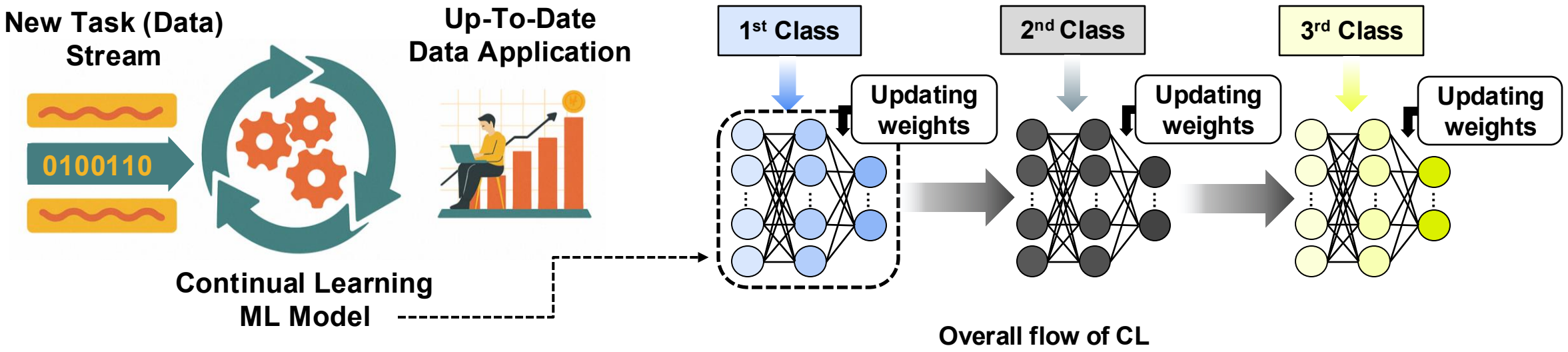

## Conventional Learning

- AI models need to be continually trained based on new data
- Security problem, expensive data transmission, and long latency ☹

## On Device Learning

- No data transmission, high privacy, and low latency ☺
  → High adaptability to dynamically changing environment with a low cost

# Continual Learning (CL)

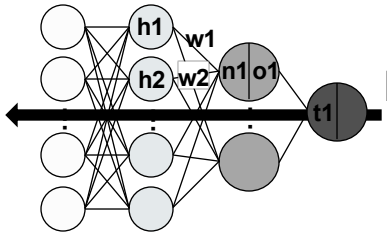



**Continual learning in artificial intelligence**

- Ability to learn and adapt continually to the changing environment
- Model can learn and accumulate knowledge for the entire duration of its "life"

# Challenges of ODL Accelerator & CL

**Challenge 1)**
Expensive gradient-based training

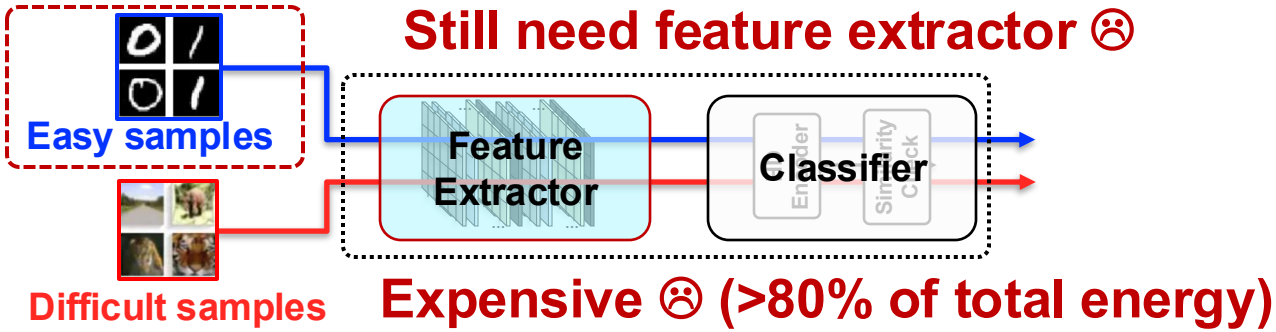



Back-prop. (Training)

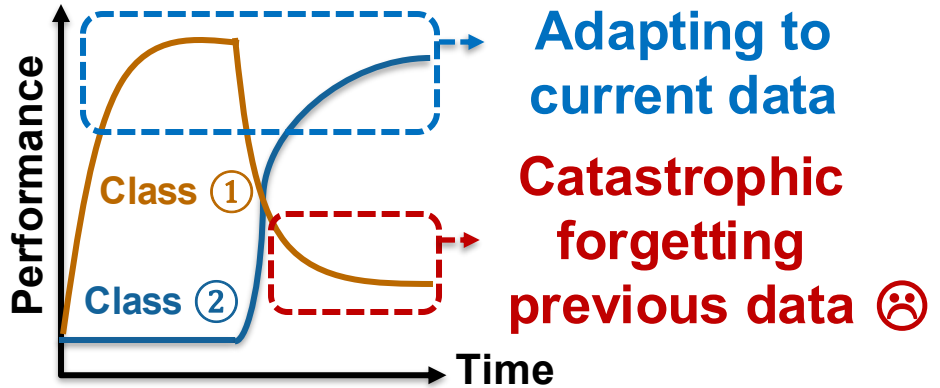
$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial o1} * \frac{\partial o1}{\partial n1} * \frac{\partial n1}{\partial w1}$$

☹Cost ↑

**Challenge 2)**
Expensive feature extractor
& redundant end-to-end computation



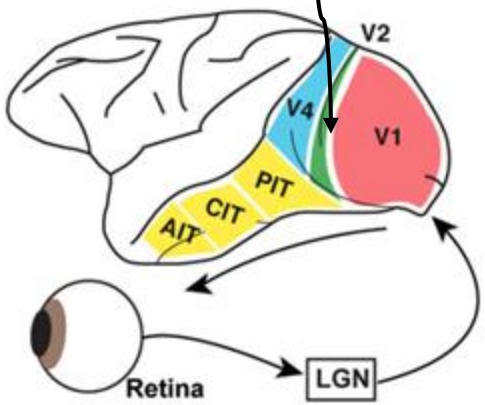

**Challenge 3)**
Non-supporting continual learning
w/ conventional ODL [VLSI'23, JSSC'23]



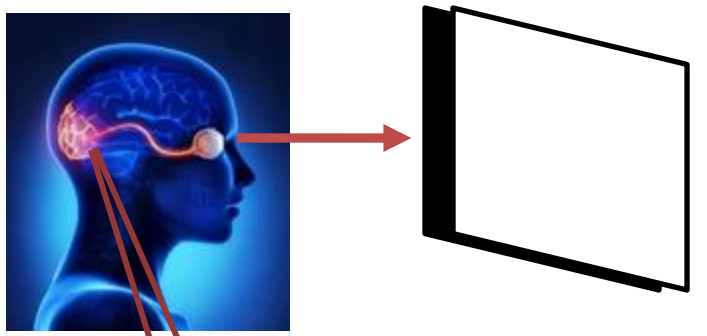
# Hyperdimensional (HD) Computing

"Dense sensory input is mapped to ***high-dimensional (HD) sparse representation*** on which brain operate"
[Babadi and Sompolinsky 2014]

**High-dimensional** Sparse Representation (190M-D)




Dense Input (1M-D)



+1 -1 -1 +1 +1 -1 +1 ... +1

+1 +1 -1 -1 +1 +1 -1 ... -1

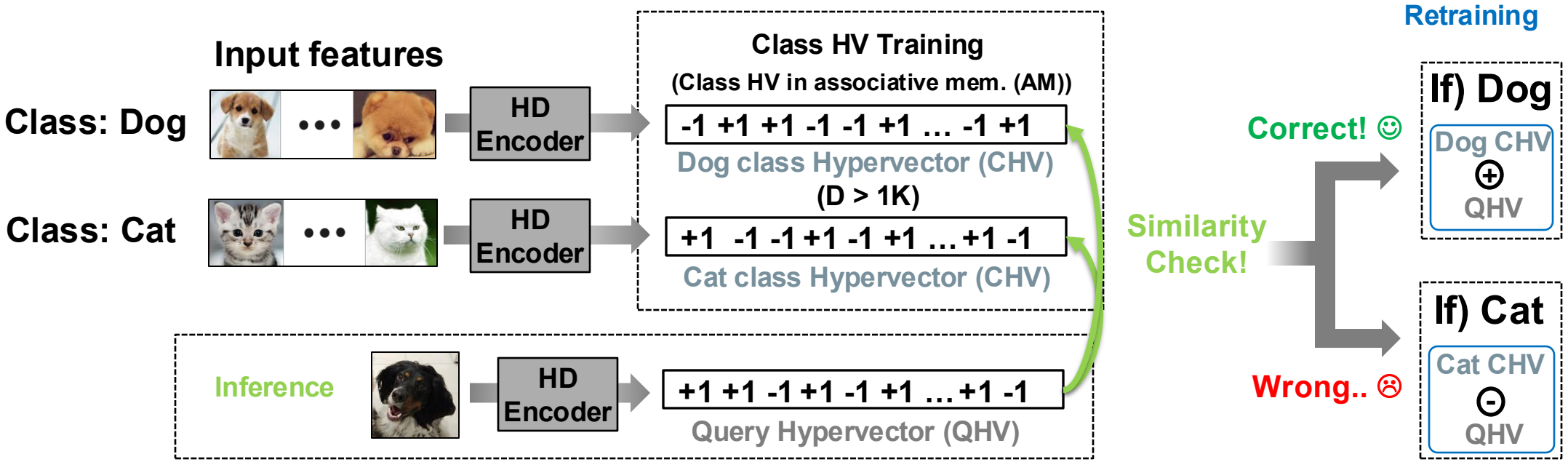
D (Dimension)

**Very lengthy vector**



Babadi B, Sompolinsky H. Sparseness and expansion in sensory representations. Neuron. 2014 Sep 3;83(5):121326

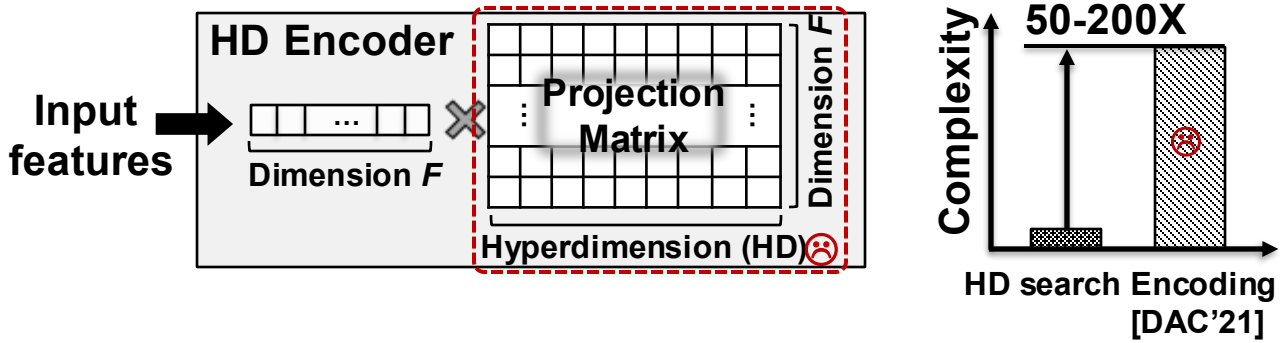
# HD Computing Example



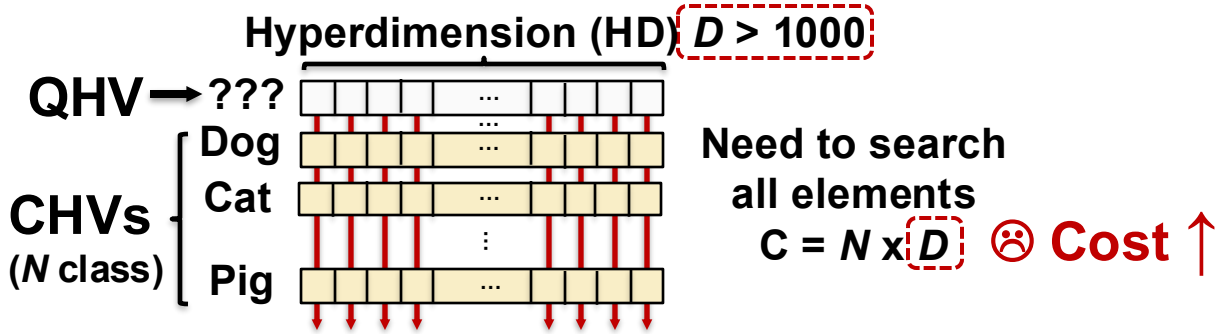

- **Lightweight:** Less computation & Low energy consumptions ☺
- **Robustness:** Resilient against noise ☺
- **Memory Preservation:** Not affecting previous CHVs ☺
- **Highly Parallelizable:** Simplified HW design ☺

# Challenges of HD Computing Accelerator

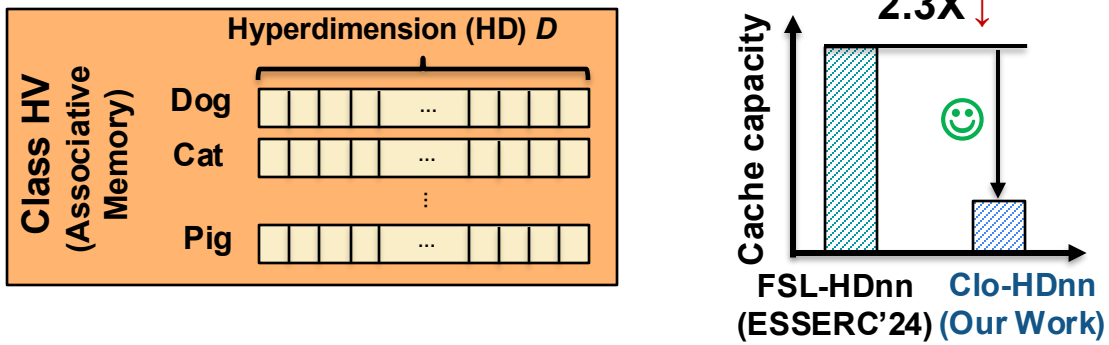
**Challenge 1)**
Computation overhead
by HD encoder




**Challenge 2)**
Inefficient full HD encoding & distance search



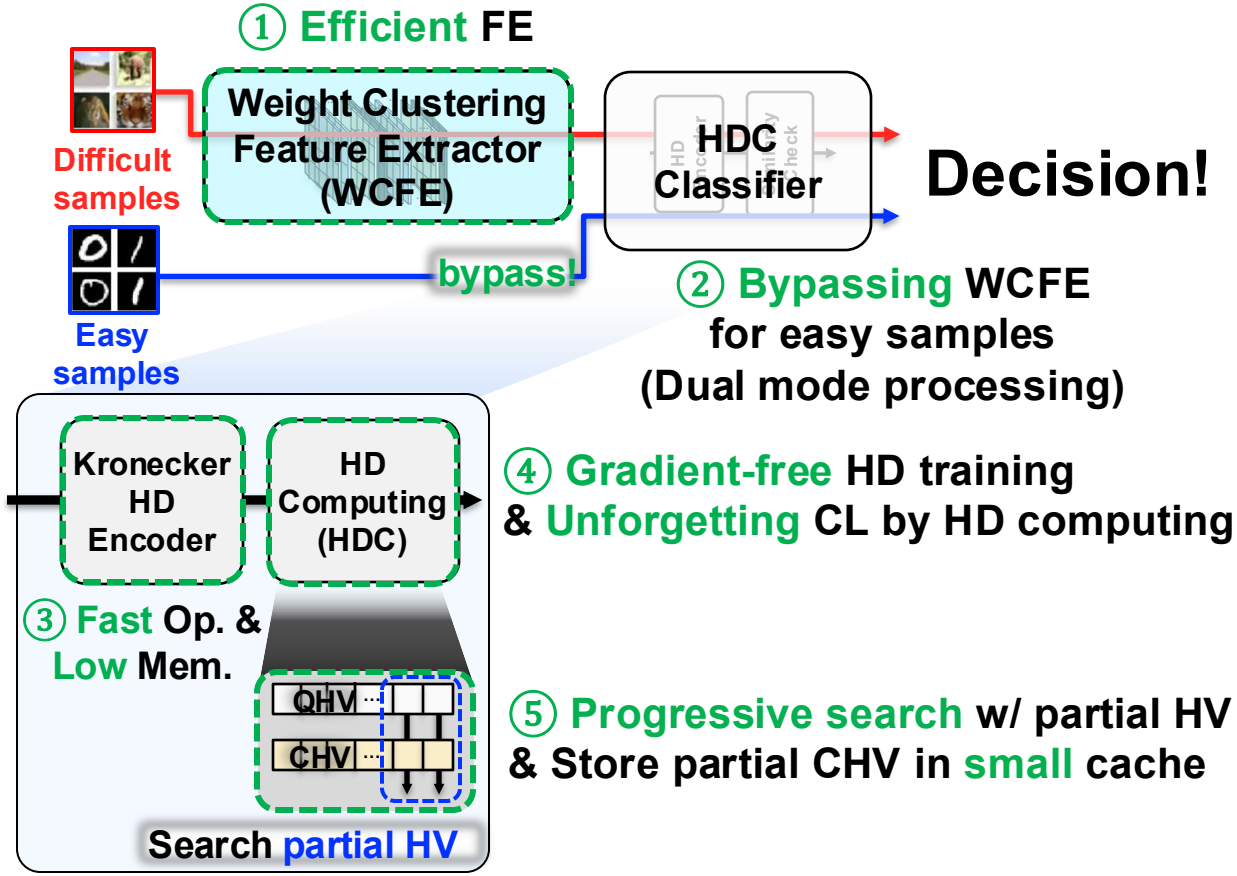

**Challenge 3)**
Huge area & power of associative memory (AM) to store CHVs



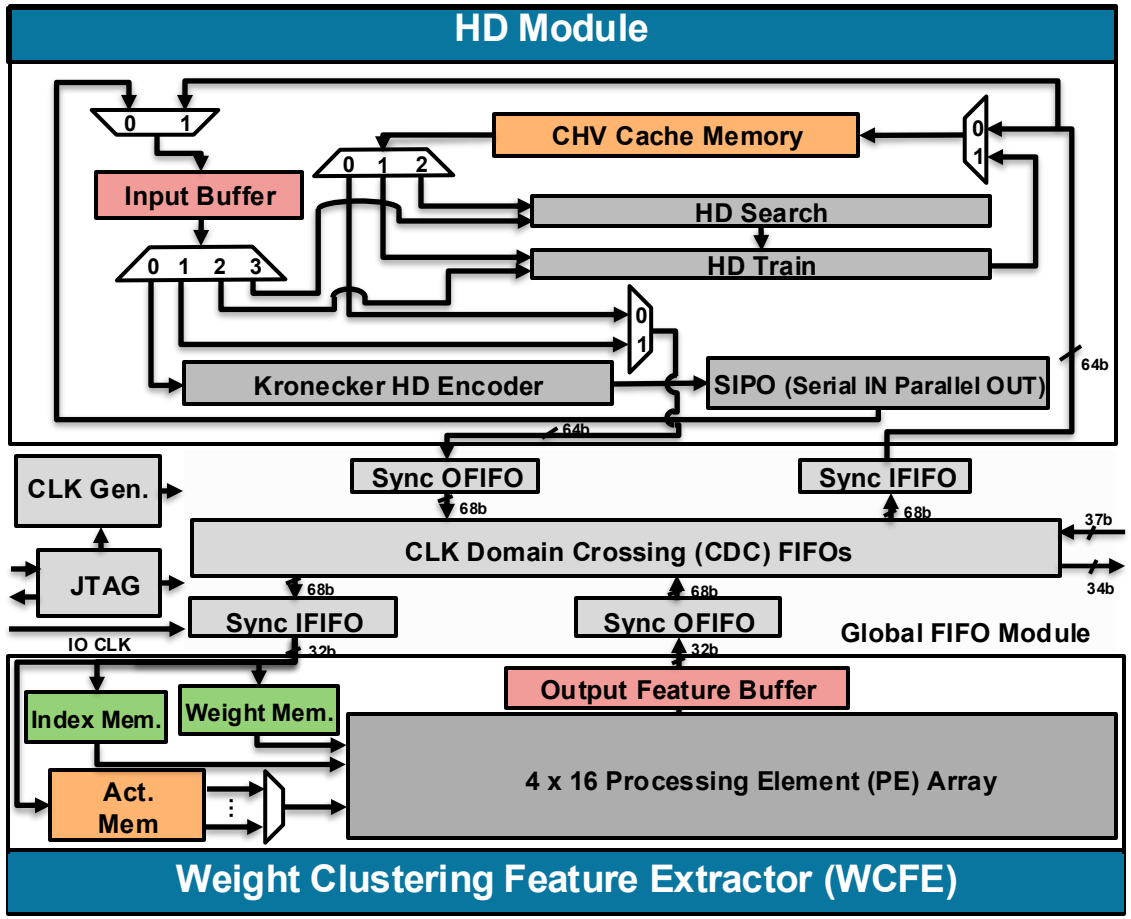
# Related Works

| | **Clo-HDnn (Our Work)** | FSL-HDnn [ESSERC'24] | SP-PIM [VLSI'23] | S. K. Venkataramanaiah ., [JSSC'23] | CHIMERA [JSSC'22] |
|---|---|---|---|---|---|
| **Learning Engine** | CNN-HDC | CNN-HDC | LET | Sparse BP | CNN-Back Prop. |
| **Workload** | CNN+CL | CNN+FSL | CNN | CNN | CNN |
| **CL** | ✓ | △ | ✗ | ✗ | ✗ |
| **CNN** | ✓ | ✓ | ✓ | ✓ | ✓ |
| **HD** | ✓ | ✓ | ✗ | ✗ | ✗ |
| **HD Encoder Type** | Kronecker | cRP-based | - | - | - |
| **On-Chip Mem (kB)** | **SRAM: 200 ☺** | SRAM: 424 | SRAM: 329 | SRAM: 1280 | RRAM: 204<br>SRAM: 512 |
| **Bit precision** | BF16/INT1-8 | BF16/INT16 | BF16 | FP8/16 | INT8 |

# Contributions of Clo-HDnn

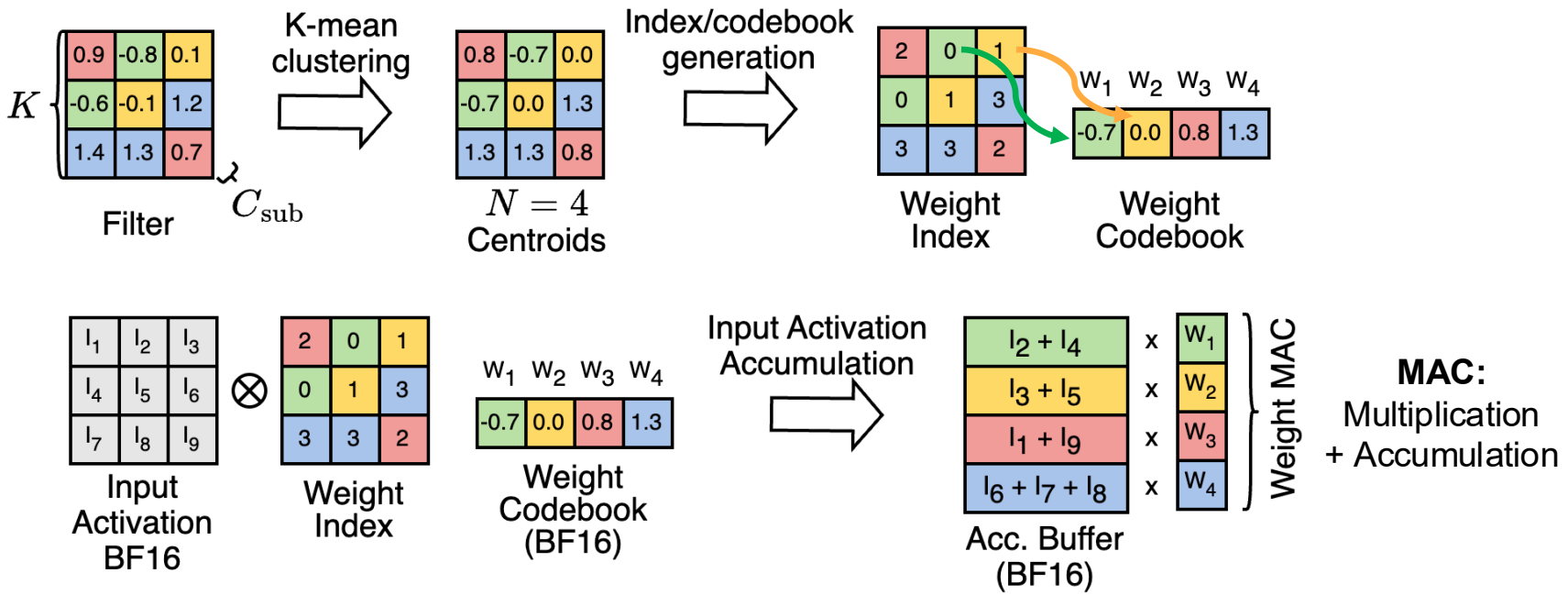



## Clo-HDnn Data Flow




## Clo-HDnn Architecture

# Weight Clustering Feature Extractor




**Step 1:** Replace with average(centroid) weights by weight clustering

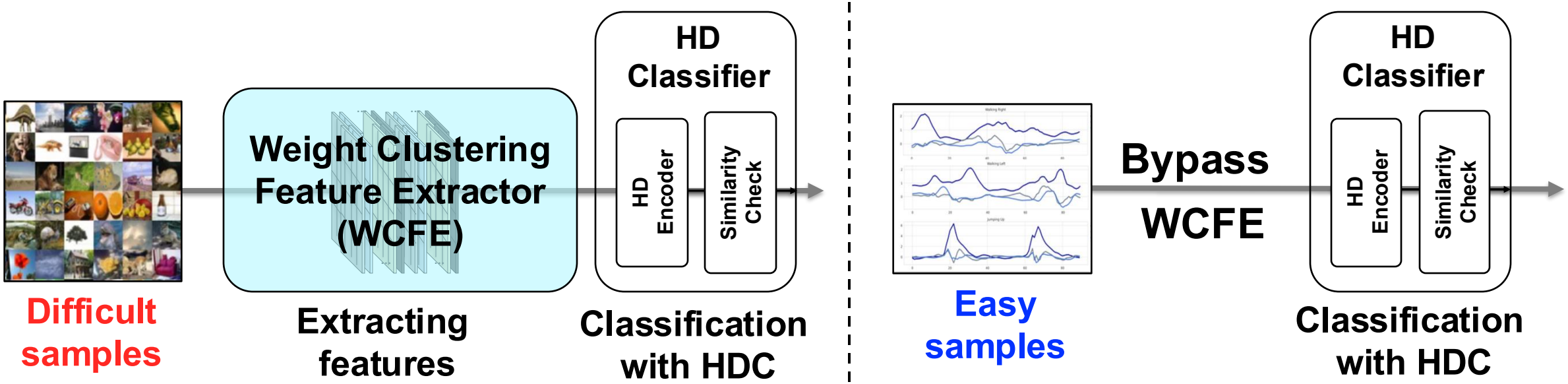
**Step 2:** Generate weight index codebook

**Step 3:** Merge inputs which sharing the same weights

**Step 4:** Multiply accumulated input with average(centroid) weights

☺ **Reduce MAC Complexity!**

# Dual Mode Processing



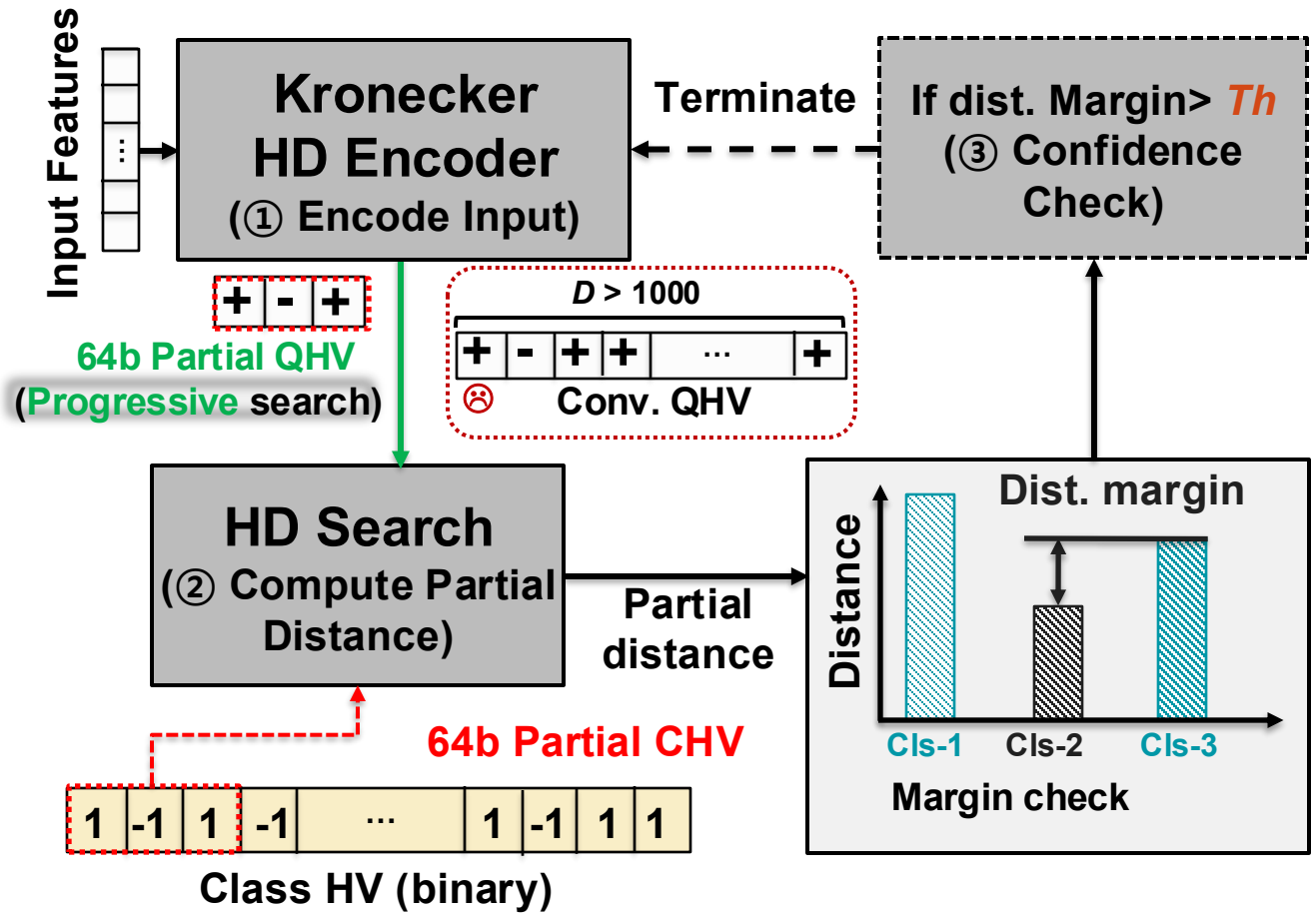

## (1) Normal Mode

Dataset with complex 2D or 3D image data with background noise, and viewpoint variation (e.g., CIFAR-100)

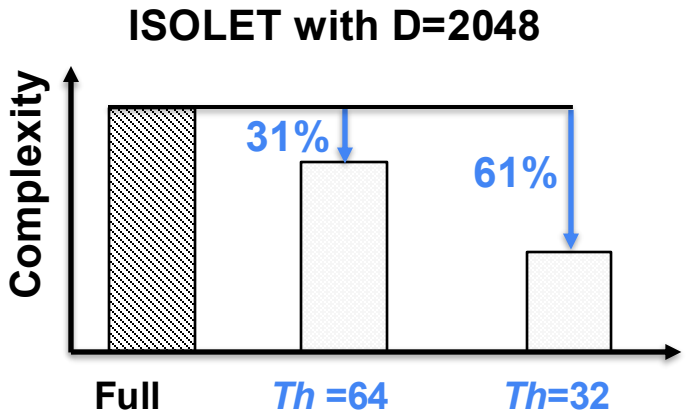
## (2) Bypassing Mode

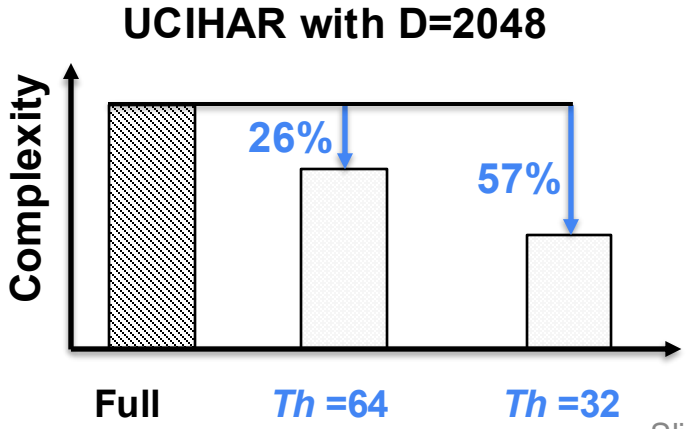
Simple 1D signals or clean time-series data collected in controlled environments (e.g, ISOLET, UCIHAR, MNIST)

# Progressive HD Distance Search

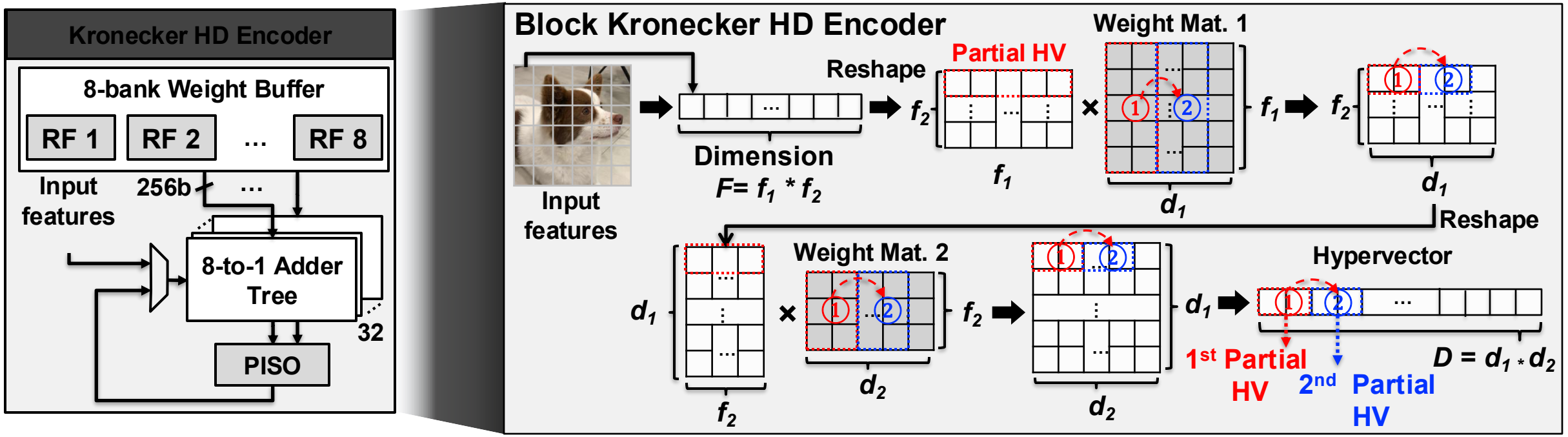



**Both case accuracy drop was ~1% (~2.5%) when *Th=64 (Th =32)***



# Kronecker HD Encoder

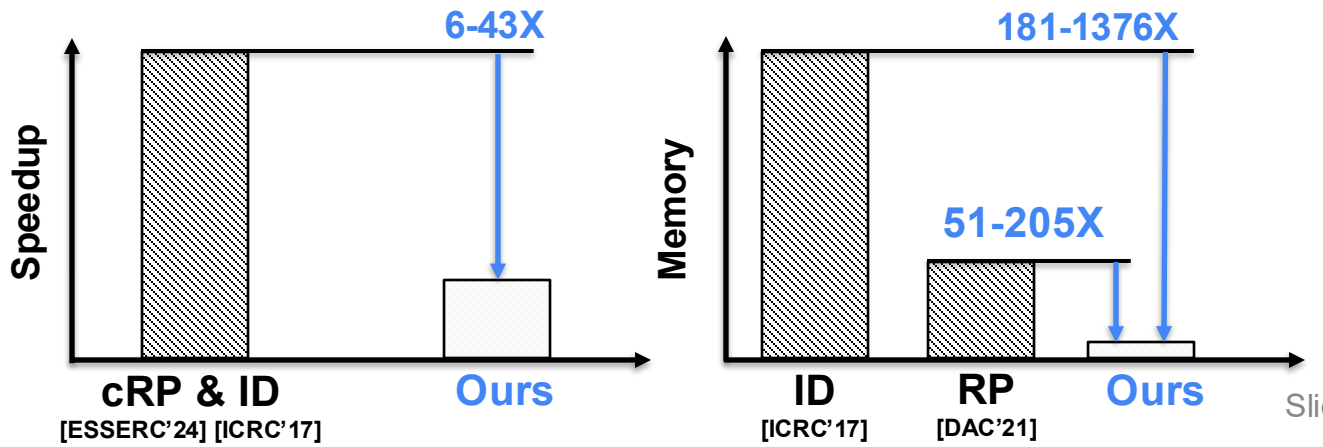



Conv. RP complexity = $f_1 * f_2 * d_1 * d_2$

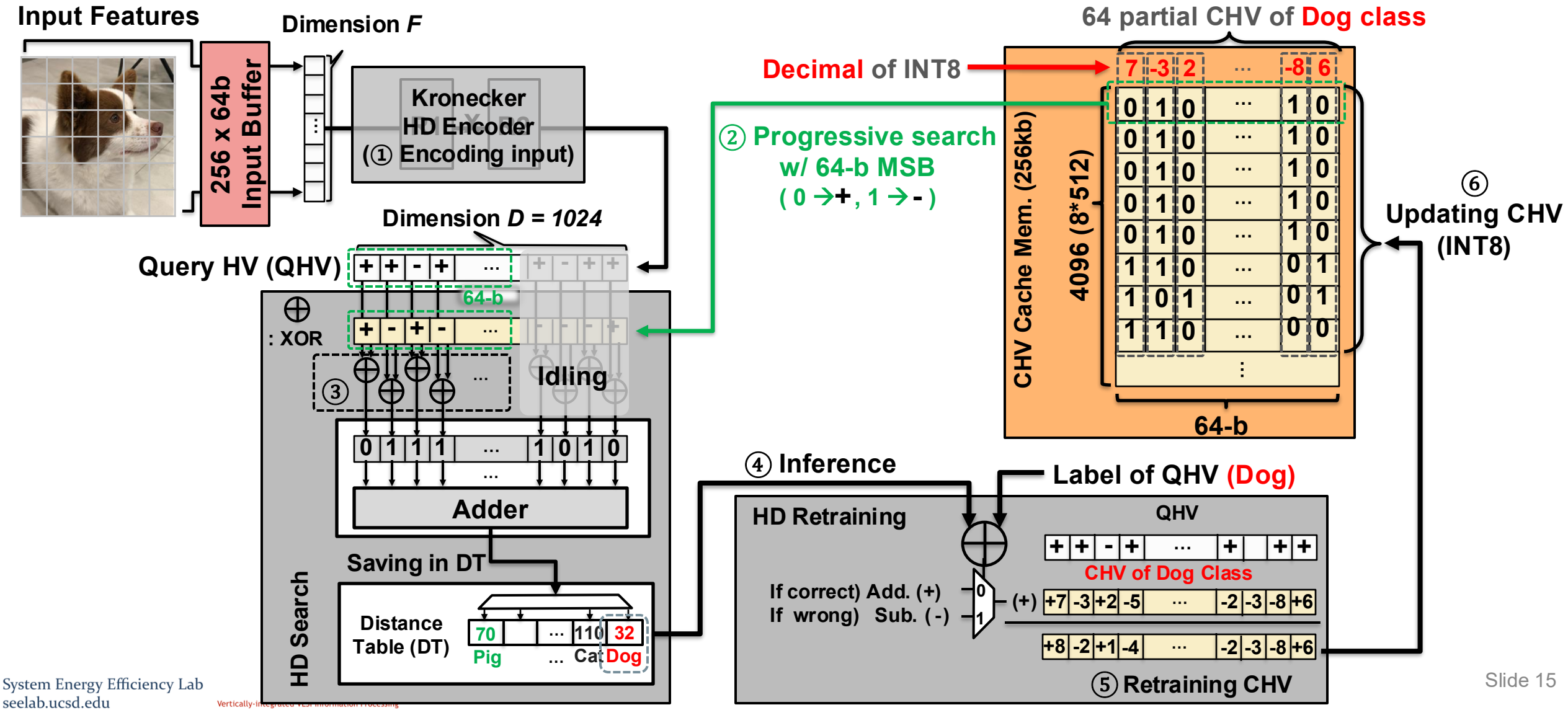
▼

Kronecker HD complexity = $f_2 * d_1 * (f_1 + d_2)$

☺ **Reduced** complexity & memory
**Low** latency & power computation cost

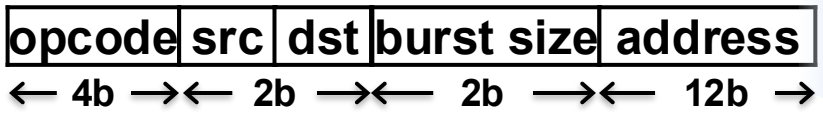


# End-to-end data flow of HD Computing



# Customized Instruction Set Architecture (ISA)

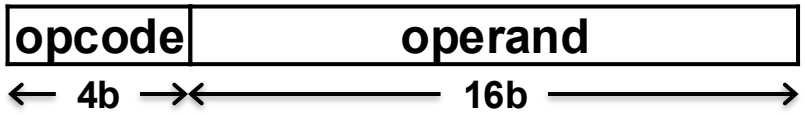
**Memory Instruction Format**




| Memory Instructions | |
|---|---|
| Name | Description |
| STORE_BUF | Data IO for on-chip buffer |
| READ_BUF | |

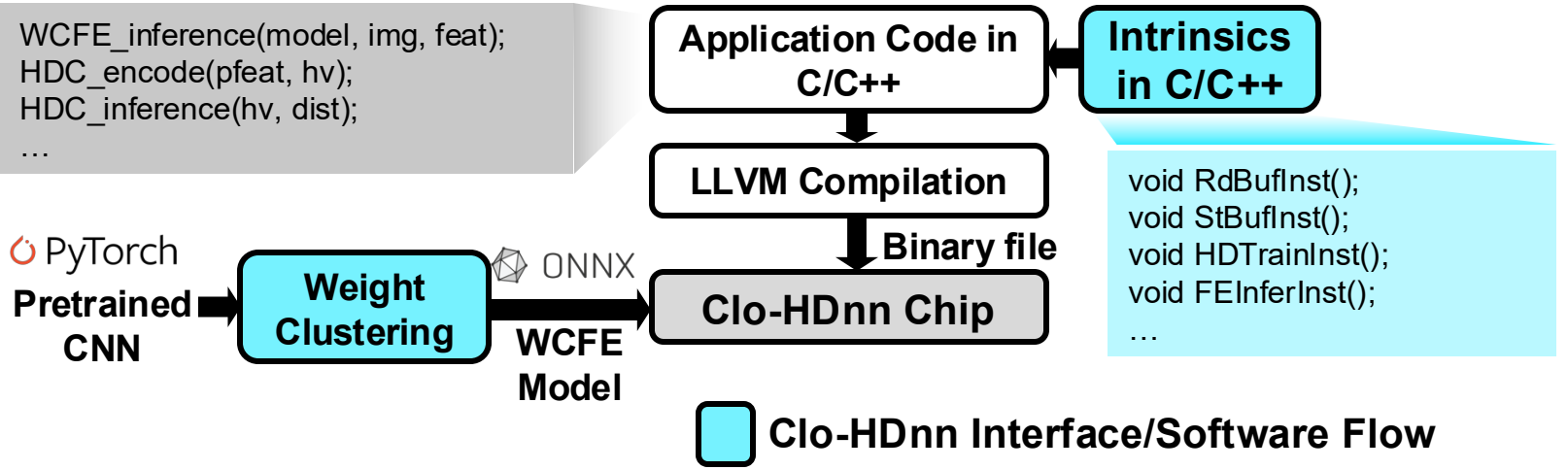
| Arithmetic Instructions | |
|---|---|
| Name | Description |
| HD_ENC_PRELOAD | HDC init. and encoding |
| HD_ENC_SEG | |
| HD_TRAIN | HDC training and inference |
| HD_INFER | |
| FE_LOAD | WCFE config, data loading and inference |
| FE_CONFIG | |
| FE_INFER | |

**Arithmetic Instruction Format**



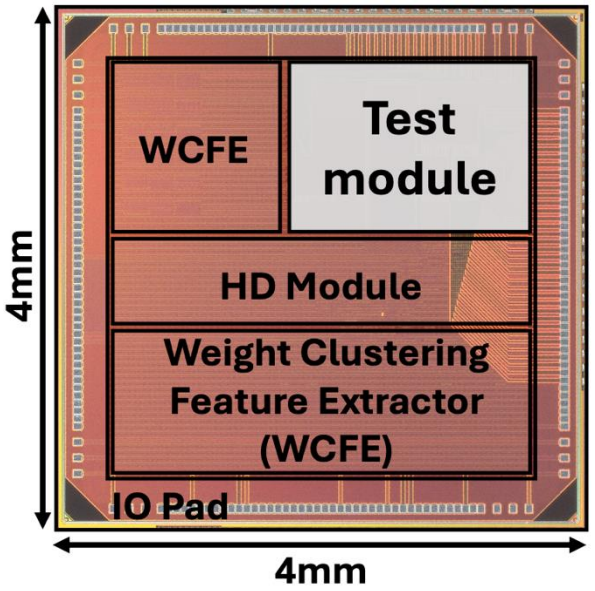

Fast & easy deployment
Flexible programming

```
WCFE_inference(model, img, feat);
HDC_encode(pfeat, hv);
HDC_inference(hv, dist);
…
```



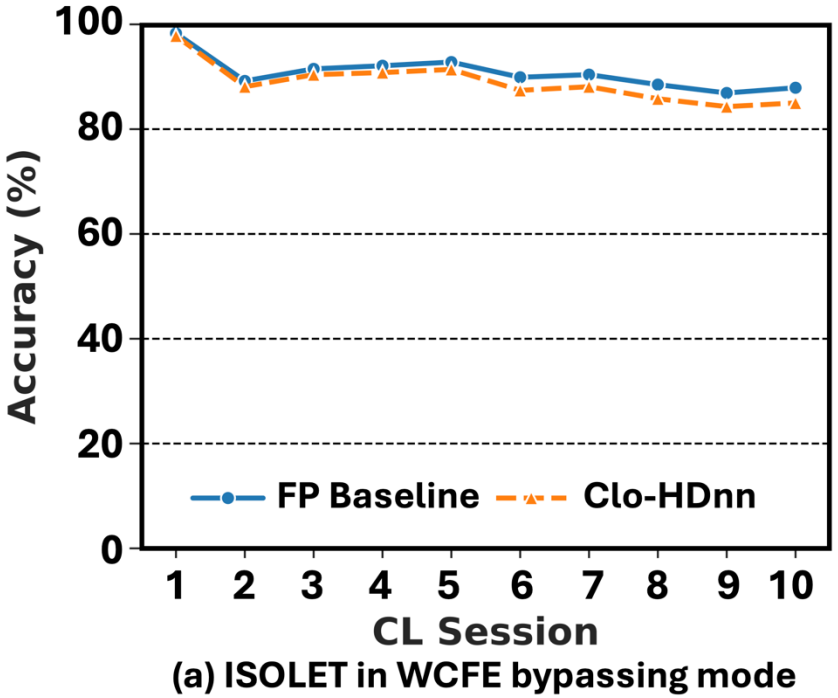
# Experimental Setup

- Software Simulation
  - Nvidia GTX1080Ti with 11GB VRAM
- Continual Learning & HDC Parameters
  - # of learning session: 10
  - Sampling strategy: Class-incremental + Random within class
  - HDC Dimension: 2048
- Software Baseline
  - FP Baseline: [X. Yu et al., IPSN'24]
    → Continual Learning with HDC
- Dataset
  - ISOLET w/ 26 classes for spoken letter recognition
  - UCIHAR w/ 6 classes for human activity recognition
  - CIFAR100 w/ 20 classes for image classification
- Silicon Testbench
  - Xilinx ZC702 FPGA for data transmission
  - Vivado 2018.3 & Vitis

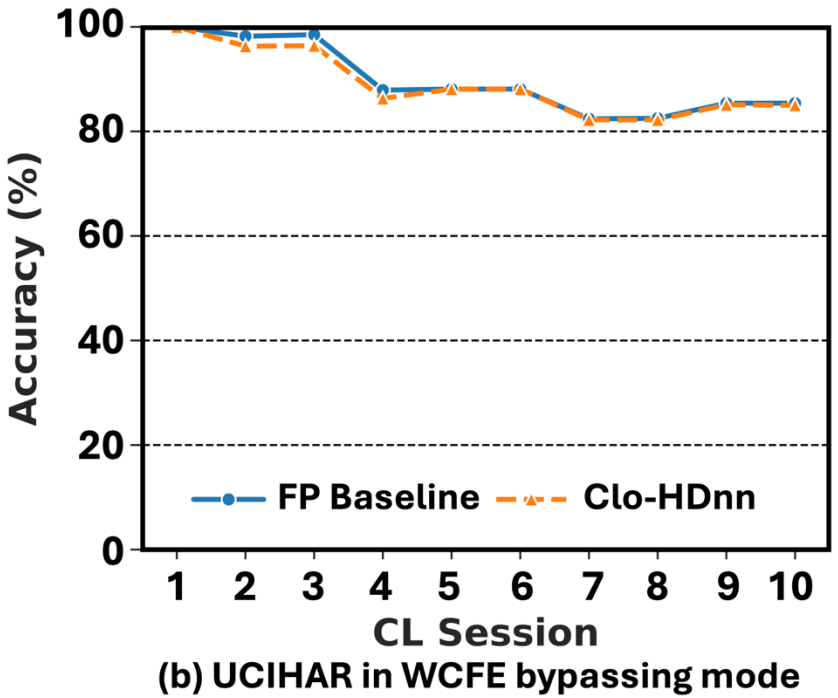



**<Clo-HDnn prototype in TSMC N40>**

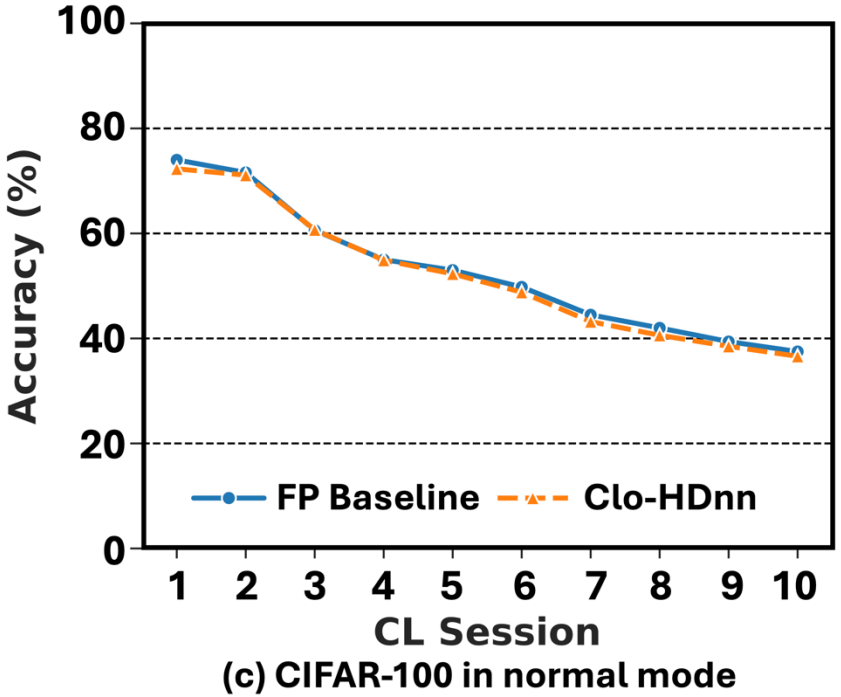
# Accuracy Results



(a) ISOLET in WCFE bypassing mode



(b) UCIHAR in WCFE bypassing mode

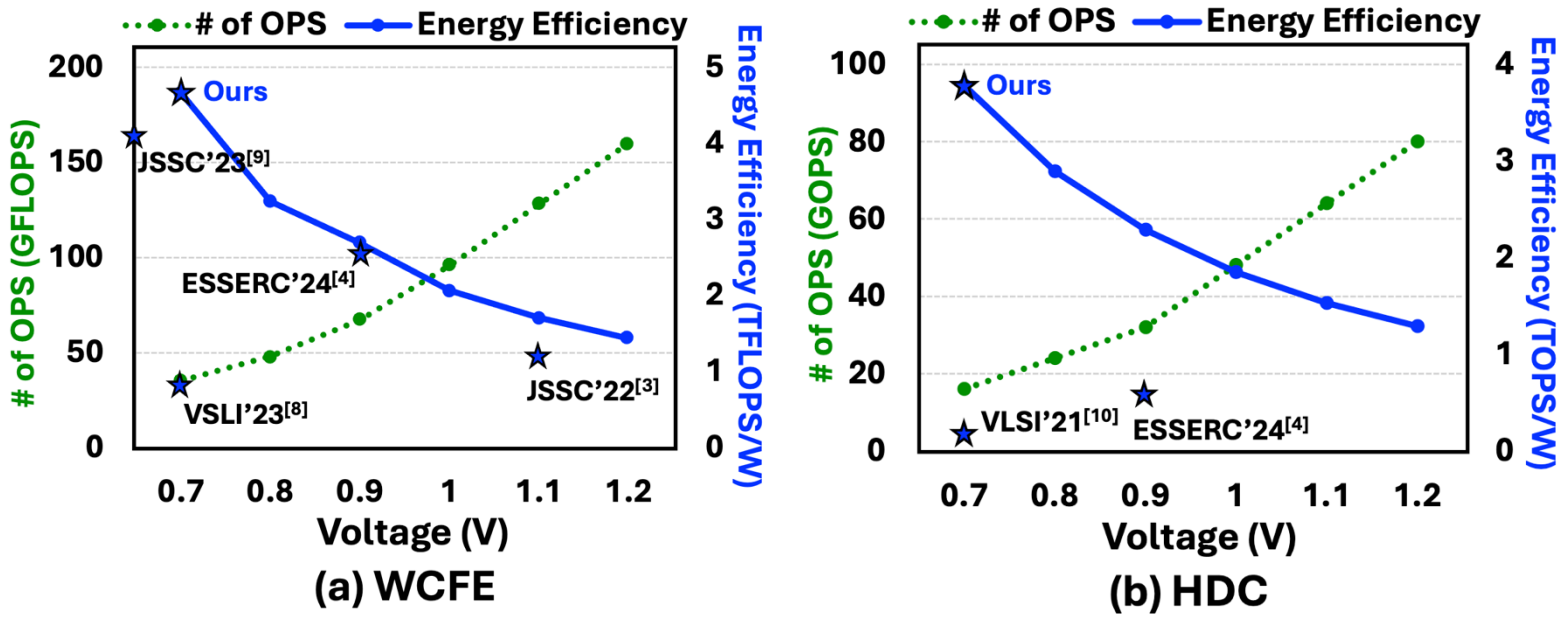


(c) CIFAR-100 in normal mode

- **Bypassing mode**
  - ISOLET& UCIHAR: Negligible accuracy degradation vs. FP Baseline [IPSN'24]
- **Normal mode**
  - CIFAR-100: Negligible accuracy degradation vs. FP Baseline [IPSN'24]

# Measured Performance

| Chip Summary Table | |
|---|---|
| Technology | 40nm CMOS |
| Die Size | 14.4 $mm^2$ |
| Capacity (kB) | SRAM: 168 (WCFE), 32 (HDC) |
| Supply Voltage | 0.7V–1.2V |
| Frequency | 50MHz – 250MHz |
| Model | CNN (WCFE) + HDC |
| Precision | BF16 (CNN)<br>INT1-8 (HDC inference)<br>INT8 (HDC training) |
| Feature Dimension (*F*) | 8-1024 |
| HDC Dimension (*D*) | 1024-8192 |
| Max # of Class | 128 |
| Peak Energy Efficiency | CNN (WCFE): 1.44-4.66 TFLOPS/W<br>HDC: 1.29-3.78 TOPS/W |

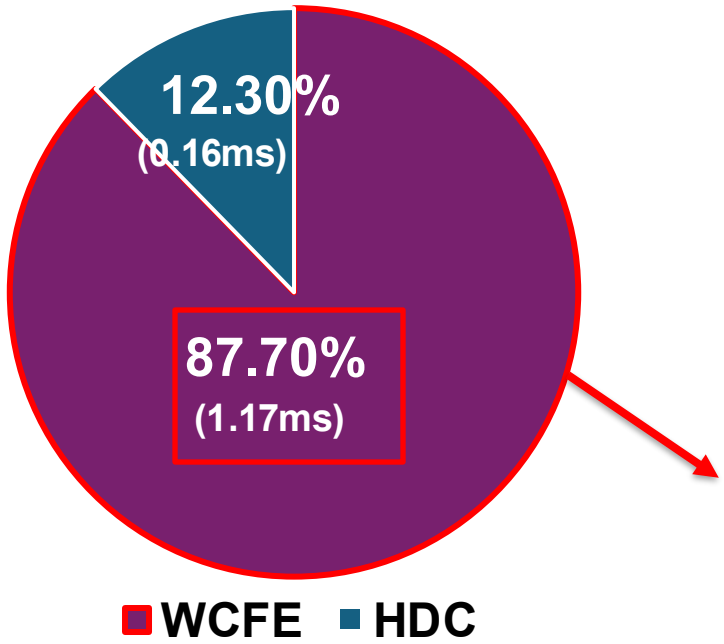



@50-250 MHz across 0.7-1.2V

- **WCFE (Feature Extractor)**
  - ~4.66 TFLOPS/W which is **1.73-7.77x** higher energy efficiency vs. SOTA [JSSC'23, VLSI'23]
- **HDC (Classifier)**
  - ~3.78 TOPS/W which is **4.85x** higher energy efficiency vs. SOTA [ESSERC'24]

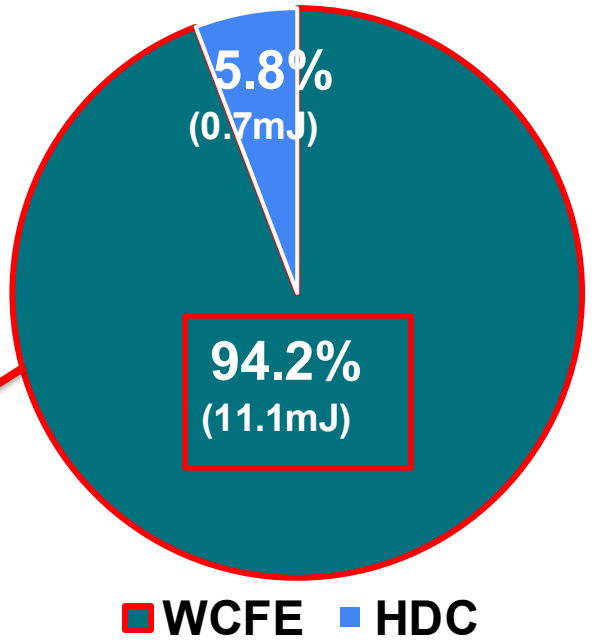
# Latency & Energy Breakdown

**Latency breakdown @1.1V, 200MHz**




**Can be reduced by bypassing mode**

**Energy breakdown @1.1V, 200MHz**




- **94.2%** of total energy consumption & **87.7%** of the latency can be reduced by WCFE bypassing mode

# Comparison Table

| | Our work | ESSERC'24 [4] | VLSI'23 [8] | JSSC'23 [9] | JSSC'22 [3] | VLSI'21 [10] |
|---|---|---|---|---|---|---|
| **Technology** | 40nm | 40nm | 28nm | 28nm | 40nm | 40nm |
| **Learning Mode** | CL HDC | FSL HDC | LET | Sparse BP | Low-rank BP | OSL |
| **Design** | Digital | Digital | Digital + CIM | Digital | Digital + CIM | ReRAM CIM |
| **Encoder Type** | Kronecker | cRP-based | - | - | - | - |
| **Precision** | BF16/INT1-8 | BF16/INT16 | BF16 | FP8/16 | INT8 | FP32 |
| **On-chip Mem. (kB)** | SRAM: 200 | SRAM: 424 | SRAM: 329 | SRAM: 1280 | ReRAM: 204<br>SRAM:512 | ReRAM: 8 |
| **Area ($mm^2$)** | 14.4 | 11.3 | 5.8 | 16.4 | 29.2 | 0.2 |
| **Frequency (MHz)** | 50-250 | 100-250 | 20-450 | 75-340 | 200 | 200 |
| **Supply voltage (V)** | 0.7-1.2 | 0.9-1.2 | 0.56-1.05 | 0.6-1.1 | 1.1 | - |
| **Scaled EE (TFLOPS/W) (CNN)** | 4.66<br>@ResNet18 | 2.69<br>@VGG16 | 0.6-0.87 | 4.1@<br>ResNet20 | 1.1* @ResNet18<br>(2.2 TOPS/W) | - |
| **Scaled EE (TOPS/W) (Classifier)** | 3.78 (HDC) | 0.78 | - | - | - | 0.12 |

All the energy efficiency (EE) is scaled to 40nm technology, * Scaled INT8 (TOPS/W) to BF16 (TFLOPS/W)

- Clo-HDnn achieves 4.66 TFLOPS/W for feature extractor by using efficient weight clustering feature extractor, showing **7.77x** higher energy efficiency compared to SOTA [VLSI'23]
- Clo-HDnn achieves 3.78 TOPS/W for classification by optimizing HDC module, showing **4.85x** higher energy efficiency compared to SOTA [ESSERC'24]

# Conclusion

- Clo-HDnn is a 40nm CMOS accelerator designed for Continual Learning for HDC tasks and is the first chip to support end-to-end process
- It leverages energy-efficient CNN-based feature extractor (WCFE) and HDC for classification with efficient progressive search
- The accelerator shows 4.66 TFLOPS/W (**7.77x** energy efficiency vs. SOTA) for feature extractor, and 3.78 TOPS/W (**4.85x** energy efficiency vs. SOTA) for classifier
- It leverages Kronecker HD encoder which supports lighter computation and less memory consumption compared to conventional HD encoder

# References


[1] W. Xu, et al., ”FSL-HD: Accelerating Few-Shot Learning on ReRAM using Hyperdimensional Computing,” DATE, Antwerp, Belgium, 2023.

Behnam Khaleghi, et al., 2022. PatterNet: explore and exploit filter patterns for efficient deep neural networks. DAC, 2022.

[2] Y. -H. Chen, et al., ”Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks,” JSSC, 2017.

[3] K. Prabhu, et al., JSSC, vol. 57, no. 4, pp. 1013-1026, 2022.

[4] H. Yang et al., ESSERC, 2024, pp. 33-36.

[5] X. Yu et al., IPSN, 2024.

[6] J. Park, at al., ”9.3 A 40nm 4.81TFLOPS/W 8b Floating-Point Training Processor for Non-Sparse Neural Networks Using Shared Exponent Bias and 24-Way Fused Multiply-Add Tree,” ISSCC, 2021.

[7] K. Prabhu et al., ”CHIMERA: A 0.92-TOPS, 2.2-TOPS/W Edge AI Accelerator With 2-MByte On-Chip Foundry Resistive RAM for Efficient Training and Inference,” JSSC, 2022.

[8] J. -H. Kim, et al., Symp. on VLSI, 2023, pp. 1-2.

[9] S. K. Venkataramanaiah et al., JSSC, vol. 58, no. 7, 2023.

[10] H. Li et al., Symp. on VLSI, 2021, pp. 1-2.

[11] H. Li et al., ”SAPIENS: A 64-kb RRAM-Based Non-Volatile Associative Memory for One-Shot Learning and Inference at the Edge,” in IEEE Transactions on Electron Devices (T-ED), 2021.

[12] A. Zhou, et al., “Incremental network quantization: Towards lossless cnns with low-precision weights,” arXiv:1702.03044, 2017.

[13] K. Hegde et al., “Ucnn: Exploiting computational reuse in deep neural networks via weight repetition,” ISCA, 2018.

[14] J. Snell et al., “Prototypical networks for few-shot learning,” NeurIPS, 2017.

# Thank you!

**For more information, please visit**

- **https://www.ucsdvvip.com/**
- **http://seelab.ucsd.edu/**

**or contact**

- **cesong@ucsd.edu**

**Chang Eun (Paul) Song**

**About Me!**

**Scan me!**

## Acknowledgements

This work was supported by TSMC and in part by PRISM and CoCoSys, centers in JUMP 2.0, an SRC program sponsored by DARPA. #455140