Lifelong Intelligence Beyond the Edge using Hyperdimensional Computing

Xiaofan Yu[1], Anthony Thomas[1], Ivannia Gomez Moreno[2], Louis Gutierrez[1], Tajana Šimunić Rosing[1]

[1] University of California San Diego
[2] CETYS University, Campus Tijuana

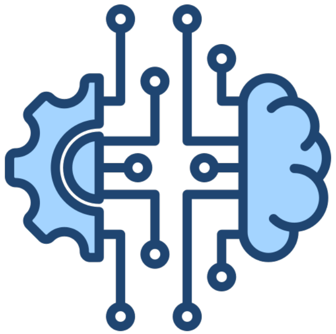
IPSN 2024

System Energy Efficiency Lab
seelab.ucsd.edu

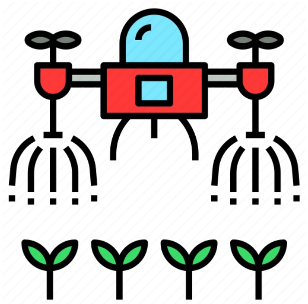
Deploy Edge Intelligence: Current Pipeline

- Current pipelines of designing and deploying edge intelligence include three steps

Collect a dataset



Train a ML model offline

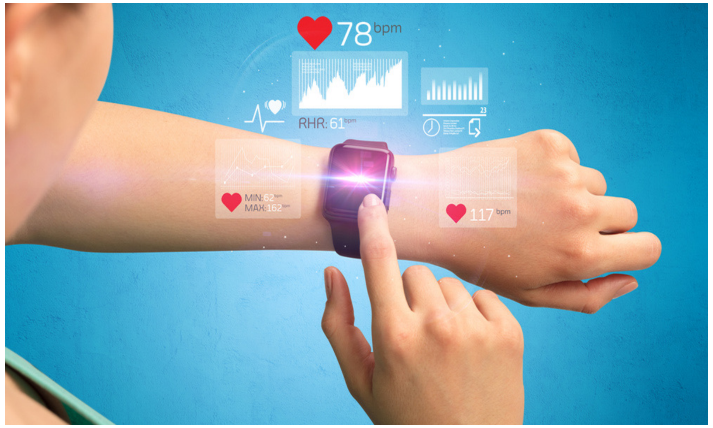


Deploy AI-enabled sensors into the wild

However, this pipeline usually does not work well in practice due to data distribution mismatch!

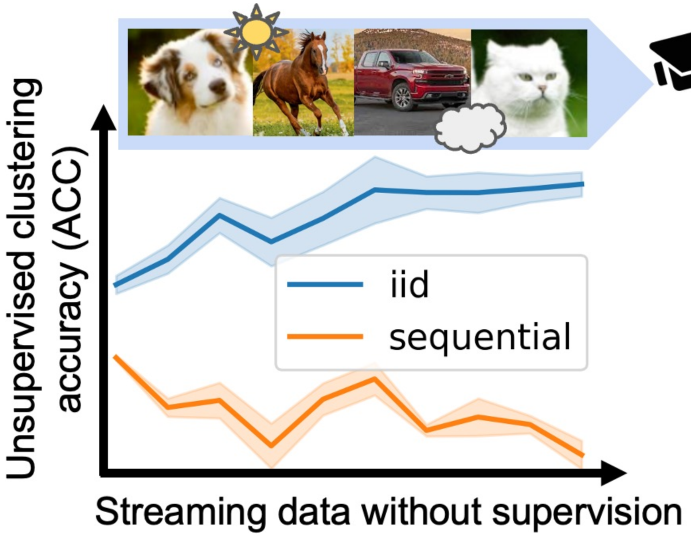
Lifelong (or Continual) Learning on the Device

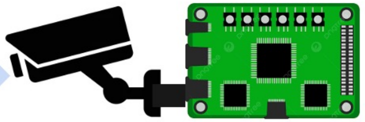
- No prior data collection
- No offline training
- The edge device learns and adapts to a continuously changing environment from its past data
- This learning process continues throughout the lifetime of the edge device



Challenges of Lifelong Learning

- Unique challenges in deploying lifelong edge intelligence
  - Catastrophic forgetting [McCloskey 1989]
  - Lack of supervision in field
  - Limited on-board resources

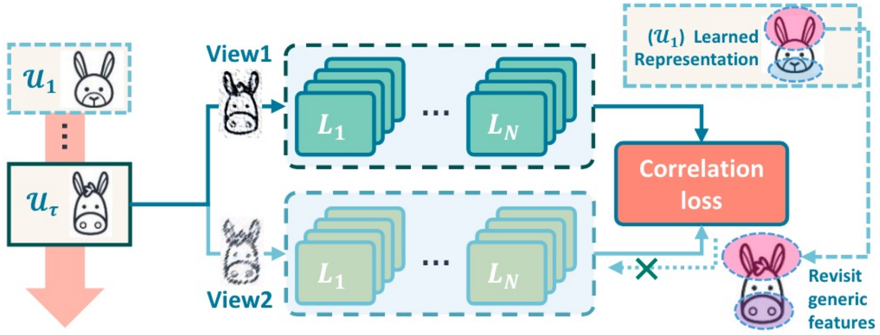




Training latency per batch Image size = 32x32x3 (RGB) Batch size = 32 samples	
Raspberry Pi 4B	17.4 sec
Jetson TX2	4.5 sec

Prior Works

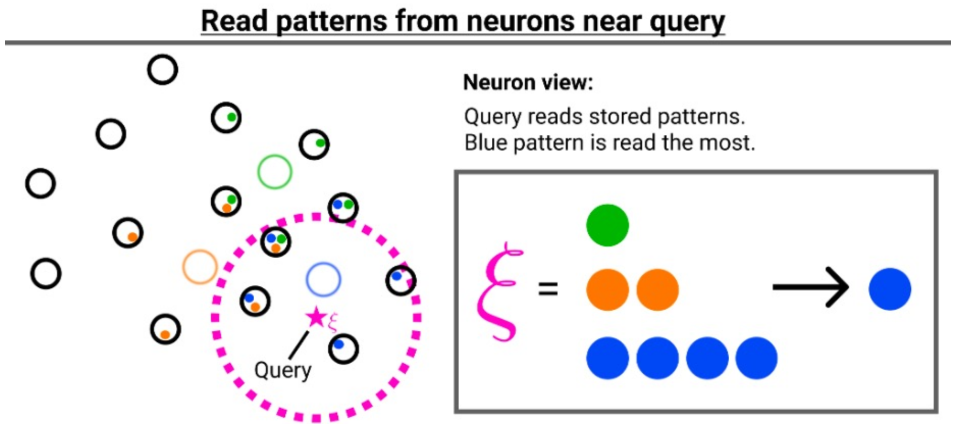
- Unsupervised lifelong learning based on NNs
  - STAM [IJCAI'21]: progressive memory architecture
  - CaSSLe [CVPR'22]: past knowledge distillation
  - LUMP [ICLR'22]: memory replay



Figures from LUMP [ICLR'22]

(+) Various techniques to mitigate catastrophic forgetting
(-) Intensive resources usage during training

- Neurally-inspired lifelong learning algorithms
  - FlyModel [Shen 2021], SDMLP [ICLR'23]: sparse coding and associative memory

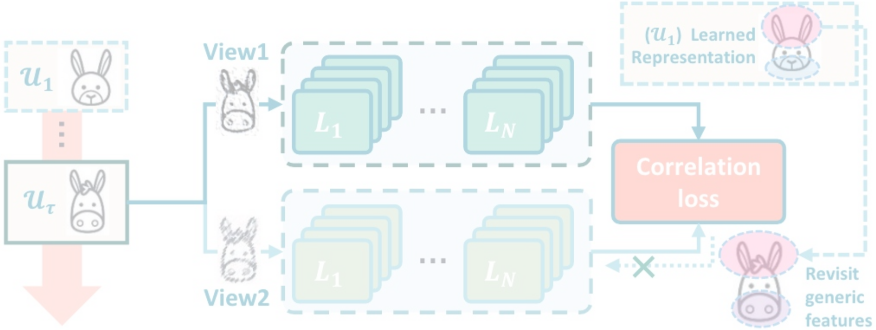


Figures from SDMLP [ICLR'23]

(+) Lightweight training
(-) Need label supervision

Prior Works

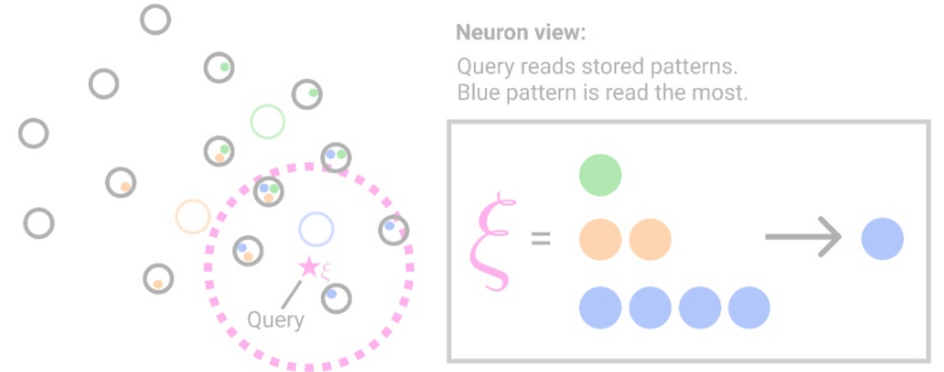
- Unsupervised lifelong learning based on NNs
  - STAM [IJCAI'21]: progressive memory architecture
  - CaSSLe [CVPR'22]: past knowledge distillation
  - LUMP [ICLR'22]: memory replay



Is there any alternative strategies for designing a lightweight and unsupervised lifelong learning algorithm?

- Neurally-inspired lifelong learning algorithms
  - FlyModel [Shen 2021], SDMLP [ICLR'23]: sparse coding and associative memory

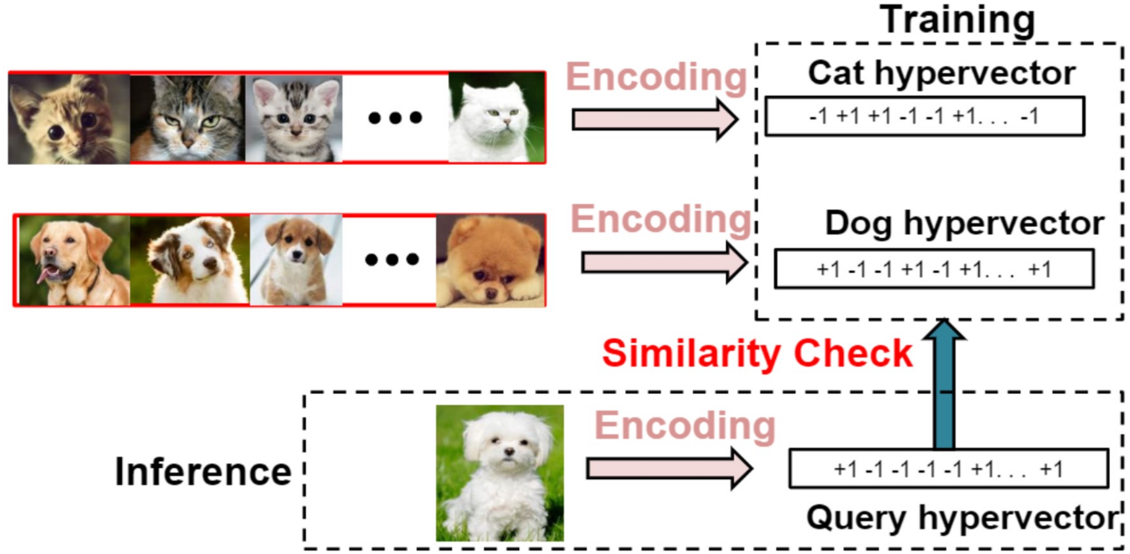
(+) Lightweight training
(-) Need label supervision

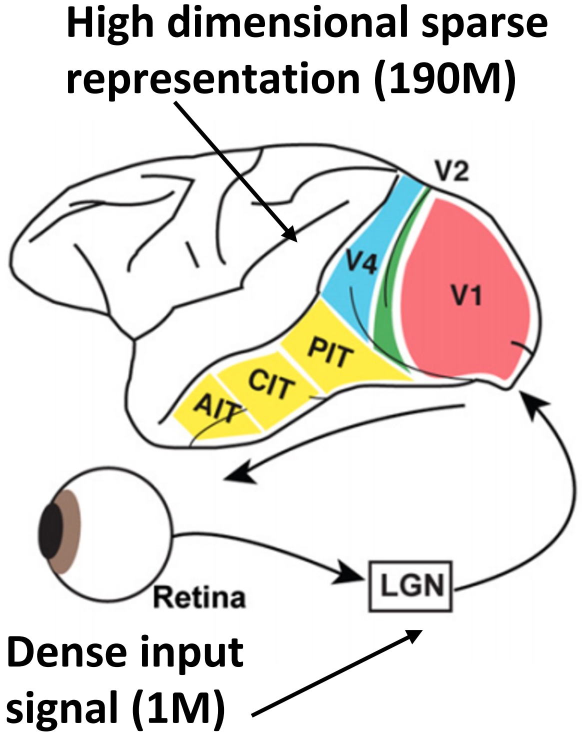


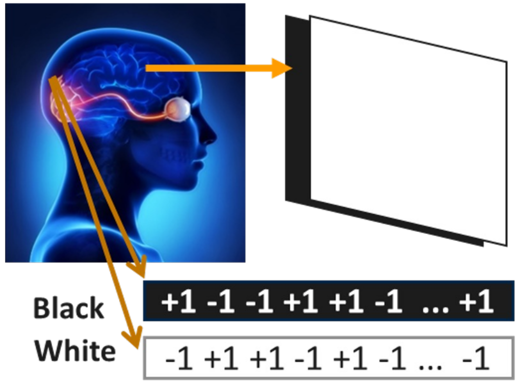
Figures from SDMLP [ICLR'23]

Brain-Inspired Hyperdimensional Computing (HDC)

Dense sensory input is mapped to **high-dimensional sparse representation** on which brain operates [Babadi and Sompolinsky 2014]





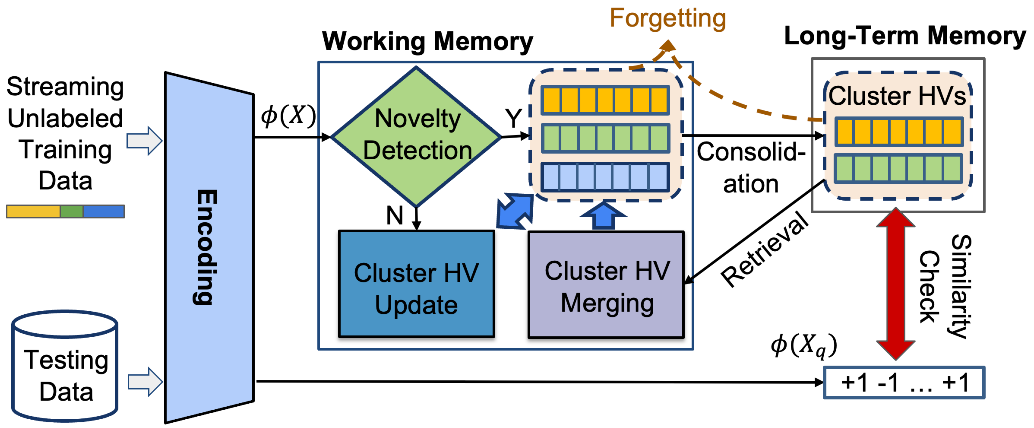


Benefits of HD computing:

- Easy-to-parallelize operations → energy-efficient
- Fast single-pass training
- Connections with biological lifelong learning in fruit flies [Shen 2021]

Our Contribution: LifeHD

- We design LifeHD, the first end-to-end system for on-device unsupervised lifelong learning using Hyperdimensional Computing

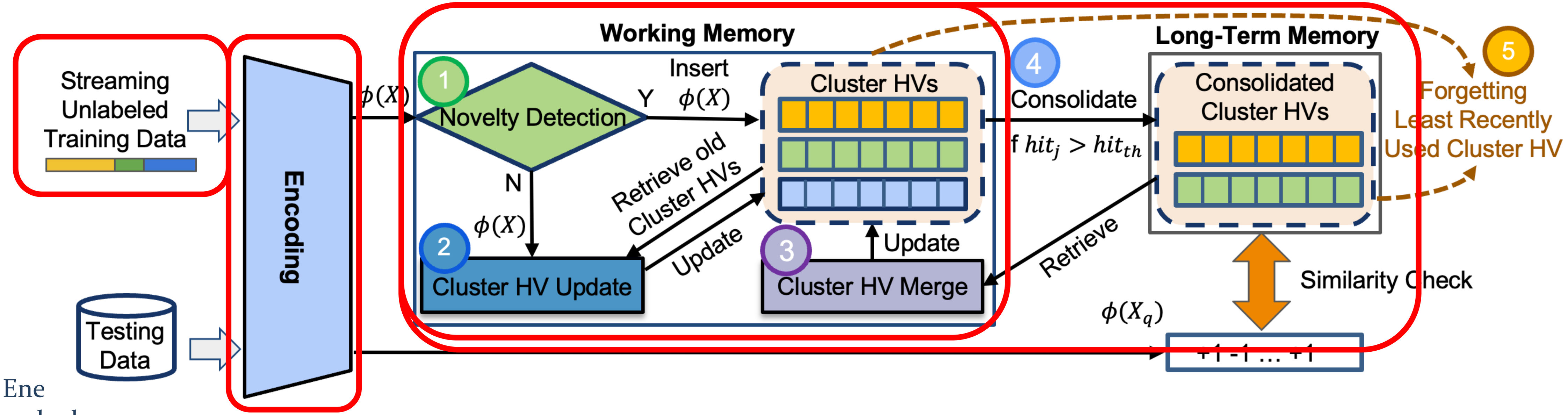


- We propose two variants of LifeHD
  - LifeHD$_{semi}$ deals with **scarce labeled inputs**
  - LifeHD$_{a}$ deals with **power constraints**

- We implement LifeHD on off-the-shelf edge devices and conduct extensive experiments across three typical IoT applications

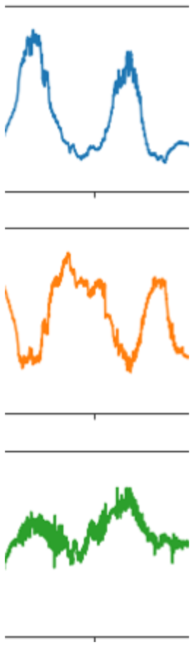
Overview of LifeHD

- Streaming data input
    - Class incremental streams with potential distribution drift
- Encoding projects dense sensor signals into high-dimensional vectors
- Two-tier associative memory design for mitigating catastrophic forgetting
- Three key components in LifeHD's working memory
    - (1) Novelty detection, (2) Cluster HV update, (3) Cluster HV Merge



LifeHD Encoding

- Encoding is the first and the most important step in HDC
- We use the Spatiotemporal HDC encoding [Nature Electronics'21]

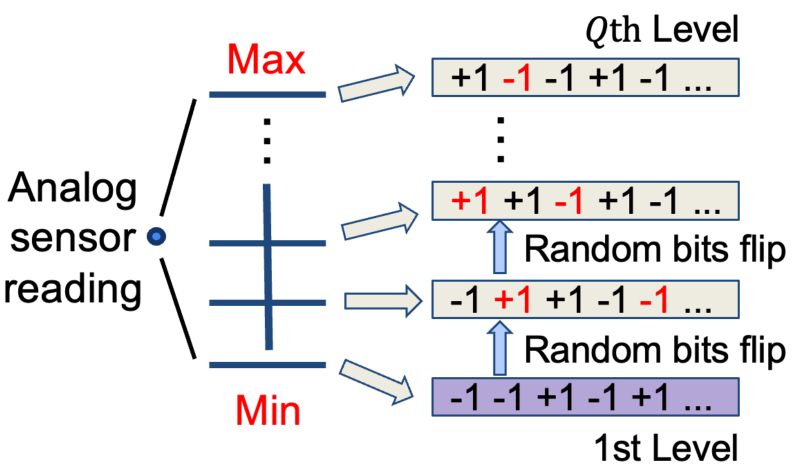


d sensors

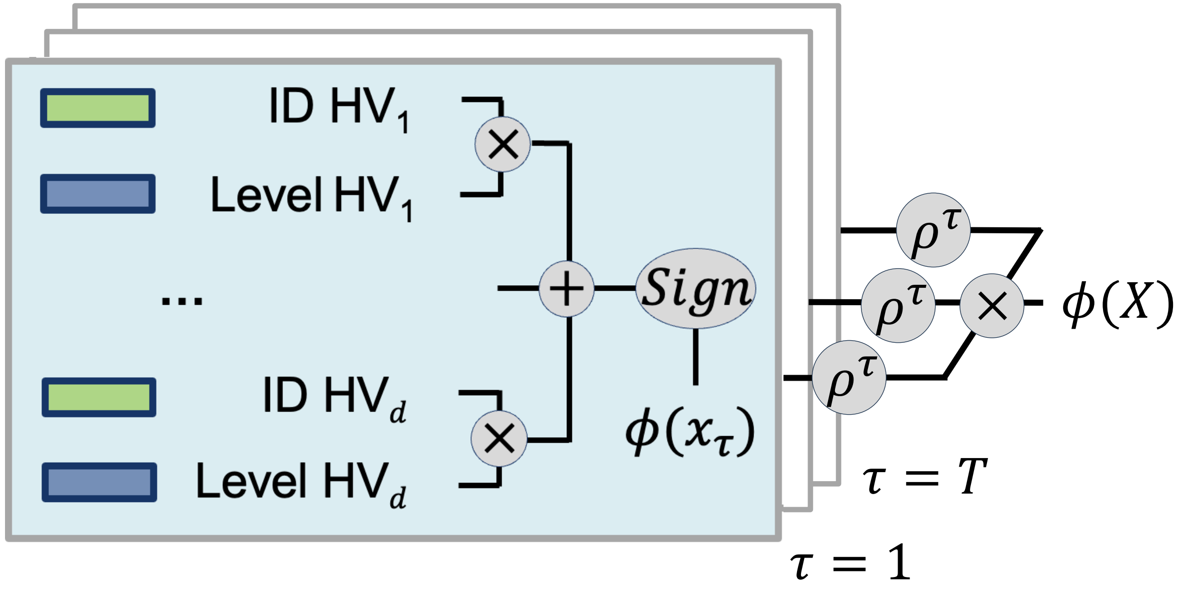
Generating Sensor ID HVs

Randomly generate d HVs

Generating Level HVs



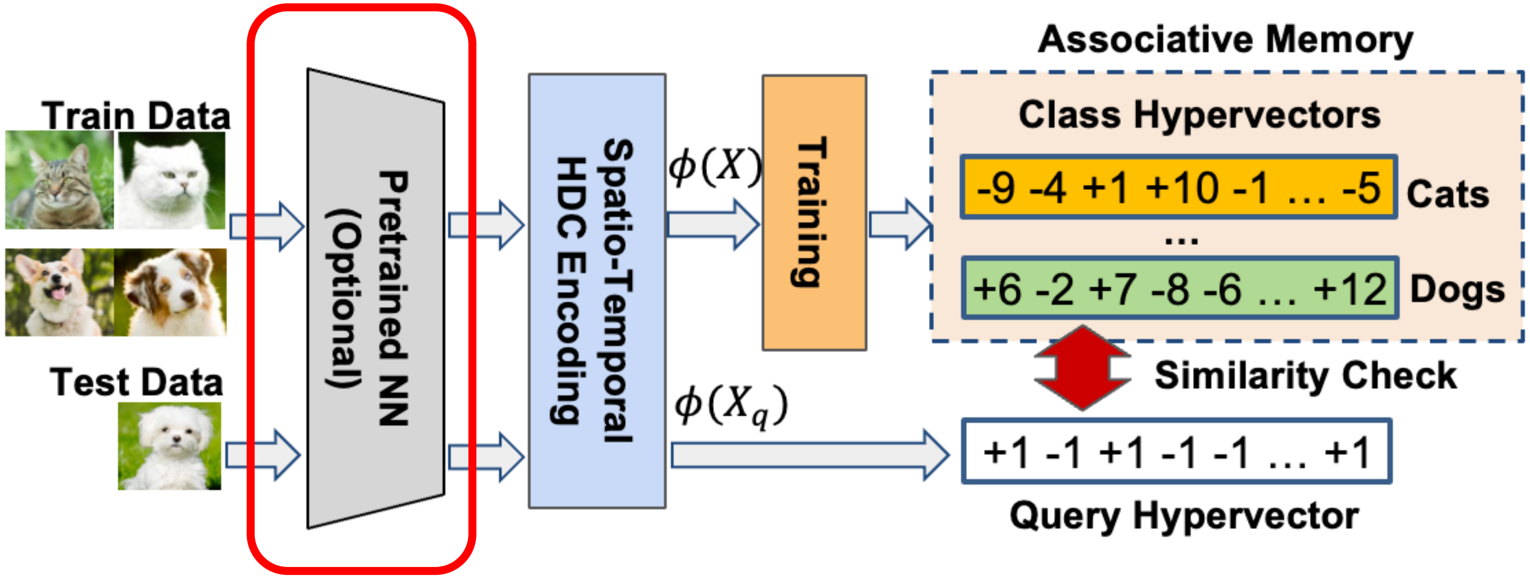
Encoding Time-Series Sensor Data



ρ^τ: Permutation $\phi(x)$: Encoded HV

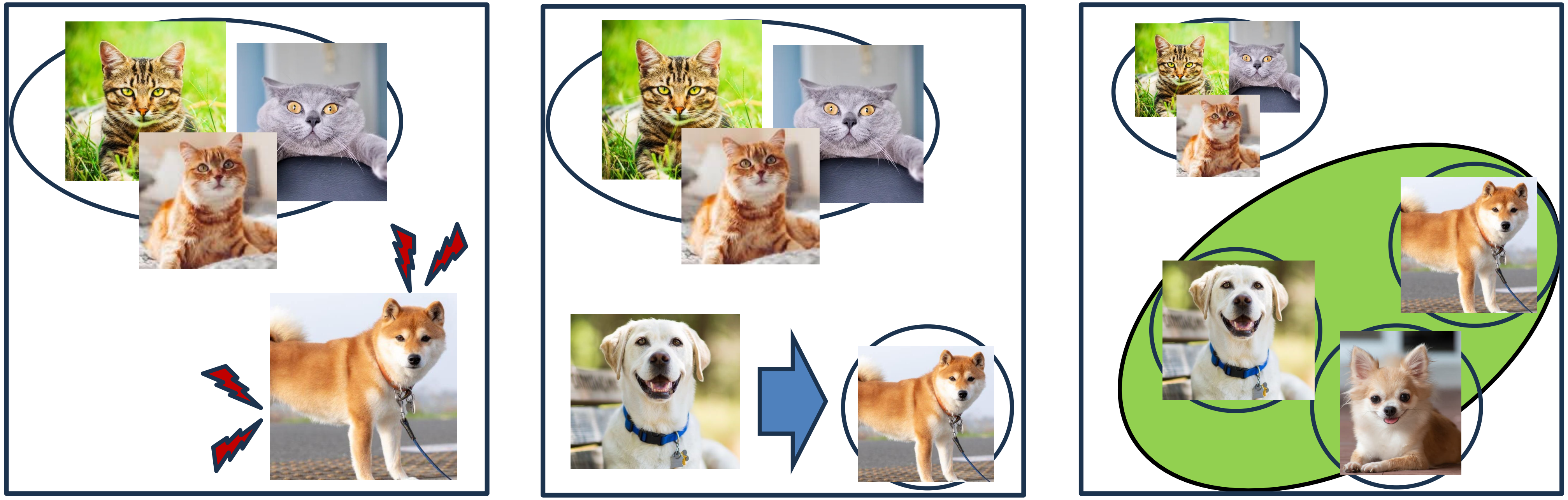
HDnn Encoding

- We use HDnn encoding [GLVLSI'22] for more complex data such as sound and images
  - A pretrained and frozen NN for feature extraction



Intuition of LifeHD's Working Memory Designs

- LifeHD's designs draw inspiration from human cognitive processes
- *Question:* How does a baby continually improve knowledge without supervision?

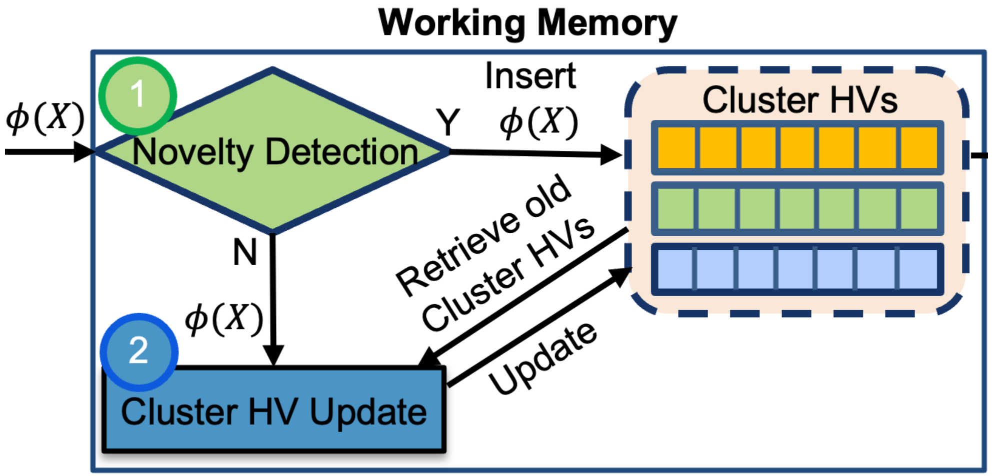


Novelty Detection | Cluster Update | Cluster Merge

Novelty Detection and Cluster HV Update

- Novelty Detection
  - If the new incoming HV $\phi(x)$ is very dissimilar from all existing cluster HVs m_j

If $\cos\left(\phi(X), m_j\right) < \mu_j - \gamma\widehat{\sigma}_j$, then flag novel



- Online Cluster HV Update
  - Update the assigned cluster HV m_j
  - Update params in a moving average manner

$$m_j \leftarrow m_j \oplus \phi(X)$$

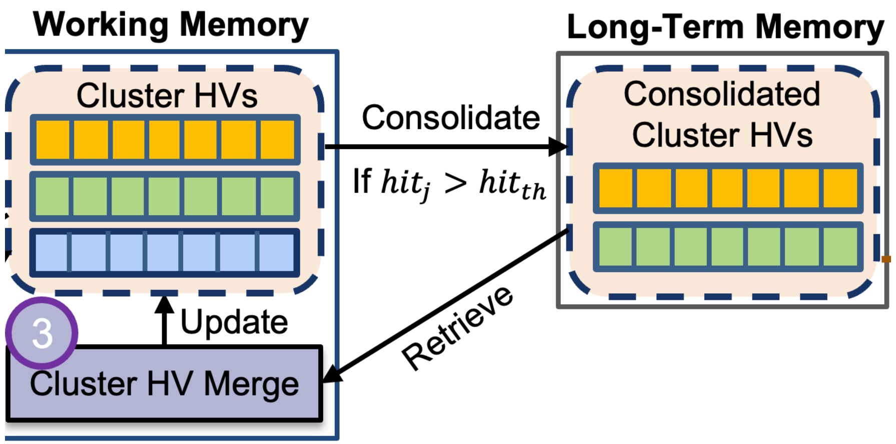
$$\mu_j \leftarrow (1-\alpha)\mu_j + \alpha\cos(\phi(X), m_j)$$

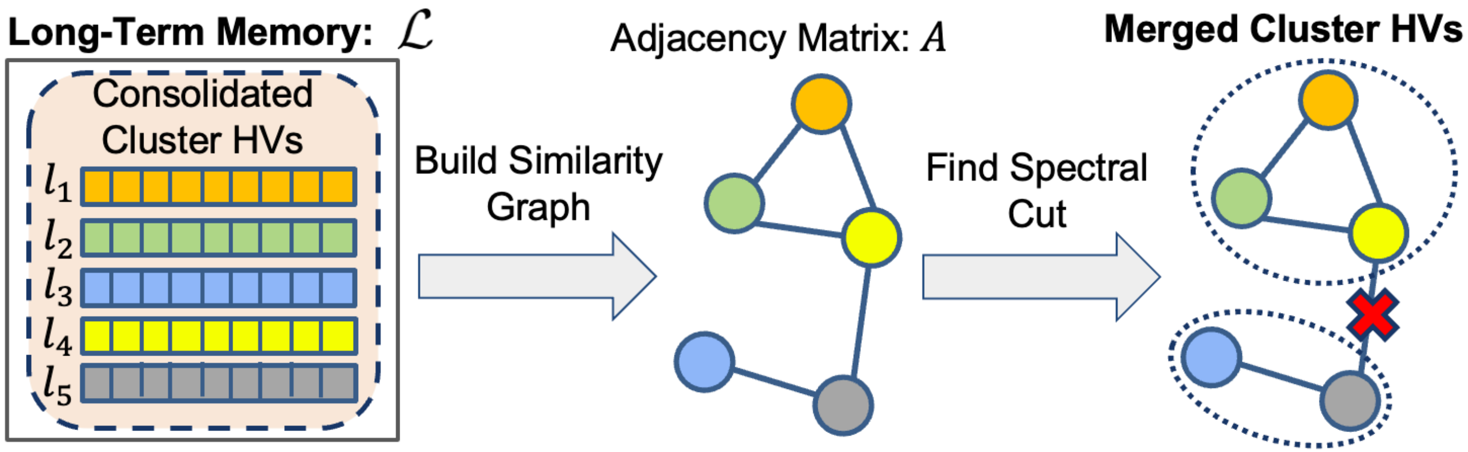
$$\hat{\sigma}_j \leftarrow (1-\alpha)\hat{\sigma}_j + \alpha|\cos(\phi(X), m_j) - \mu_j|$$

Sym.	Meaning
$\phi(x)$	Incoming encoded HV
m_j	The jth stored cluster HV
$\mu_j, \widehat{\sigma}_j$	Mean and standard difference of similarity threshold
γ, α	hyperparameters

Cluster HV Merge

- Analyze the global similarity relationship between long-term cluster HVs
- Group "similar" cluster HVs into a "coarser" one if appropriate
- Update the working memory





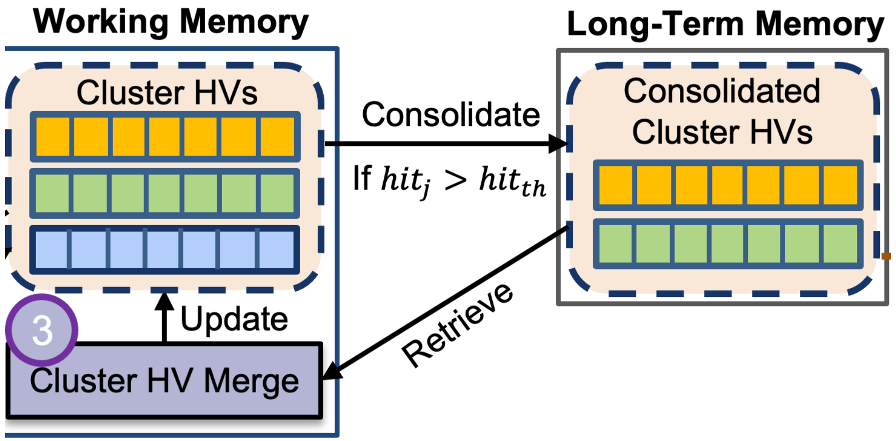
Step 1: Build a similarity graph

Step 2: Compute the eigendecomposition of the similarity matrix

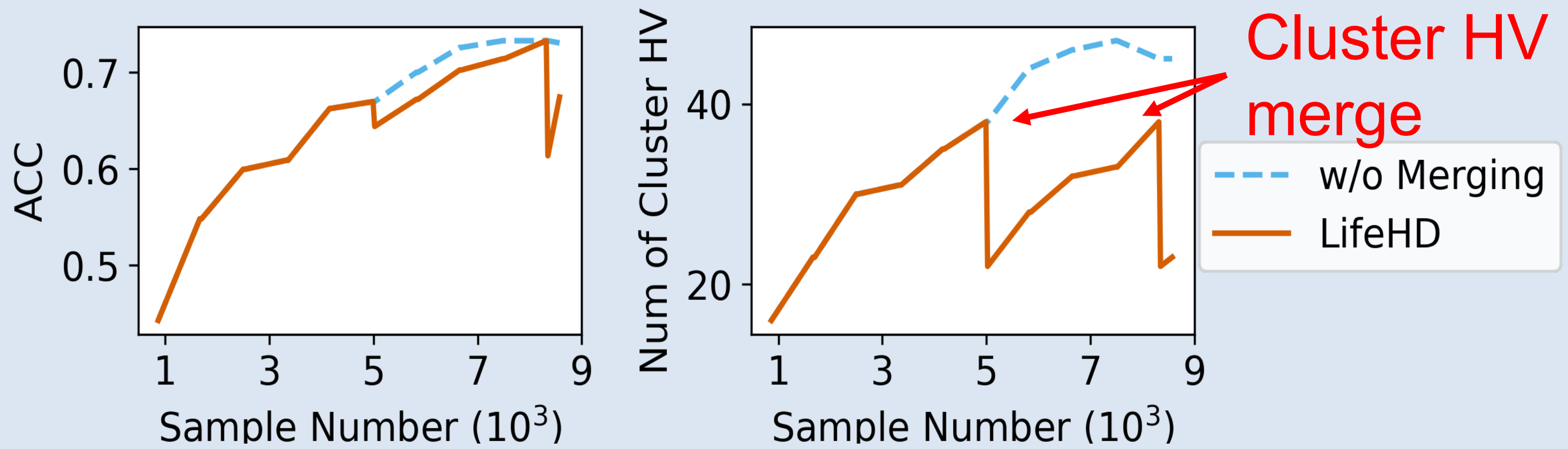
Step 3: Group the cluster HVs by running K-Means on eigenvectors

Cluster HV Merge

- Analyze the global similarity relationship between long-term cluster HVs
- Group "similar" cluster HVs into a "coarser" one if appropriate
- Update the working memory



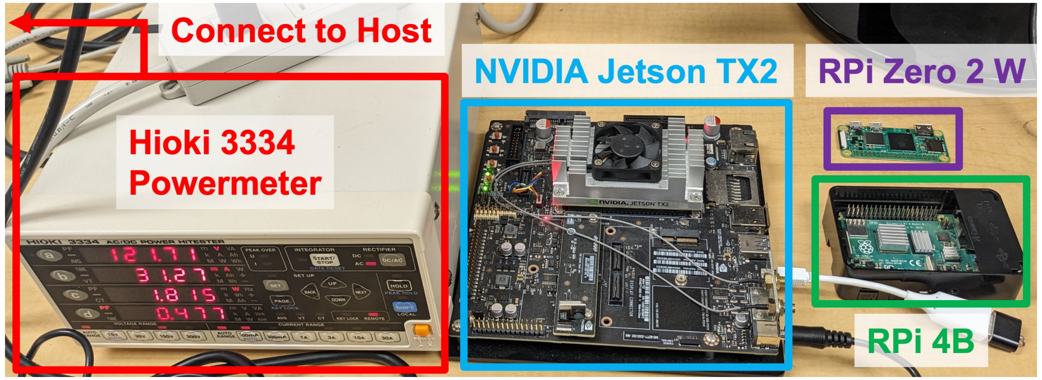
Intuitive Visualization



Ablation study of LifeHD on MHEALTH

Experimental Setup

- We implement LifeHD in Python and PyTorch on
  - Raspberry Pi Zero 2W
  - Raspberry Pi 4B
  - NVIDIA Jetson TX2 (w/ GPU)



- We test on three typical IoT applications

Dataset	Application	Classes (Balanced?)	Total Samples	Pretrained Neural Network in HDnn
MHEALTH [1]	Human activity recognition	12 (N)	9K	/
ESC-50 [2]	Sound recognition	50 (Y)	2K	ACDNet [4]
CIFAR-100 [3]	Image classification	20 (Y)	60K	MobileNet [5]

[1] Karol J Piczak. ESC: Dataset for environmental sound classification. 2015
[2] Garcia Rafael Banos, et al. MHEALTH Dataset. UCI Machine Learning Repository. 2014
[3] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009
[4] Md Mohaimenuzzaman et al. Environmental Sound Classification on the Edge: A Pipeline for Deep Acoustic Networks on Extremely Resource-Constrained Devices. Pattern Recognition 133 (2023), 109025.
[5] Mark Sandler et al. Mobilenetv2: Inverted residuals and linear bottlenecks. CVPR'18.

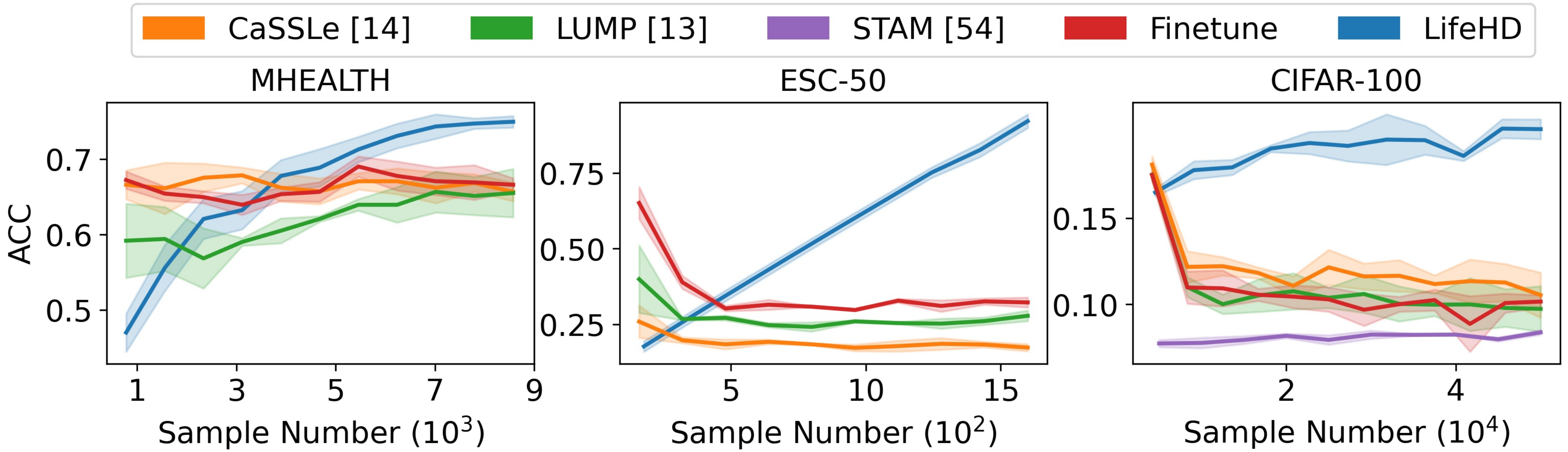
Experimental Setup (Cont.)

- Baselines
  - We compare with SOTA neural network-based unsupervised lifelong learning
    - STAM [IJCAI'21]: progressive memory architecture
    - CaSSLe [CVPR'22]: past knowledge distillation
    - LUMP [ICLR'22]: memory replay
  - We also compare with the fully Supervised HDC baseline

- Metrics
  - Unsupervised Clustering Accuracy (ACC)
    - ACC computes the accuracy under the "best" mapping between clusters and labels
  - Training time per batch
  - Energy consumption per batch
  - Memory usage

On all platforms

LifeHD vs. SOTA Neural Network-based Baselines



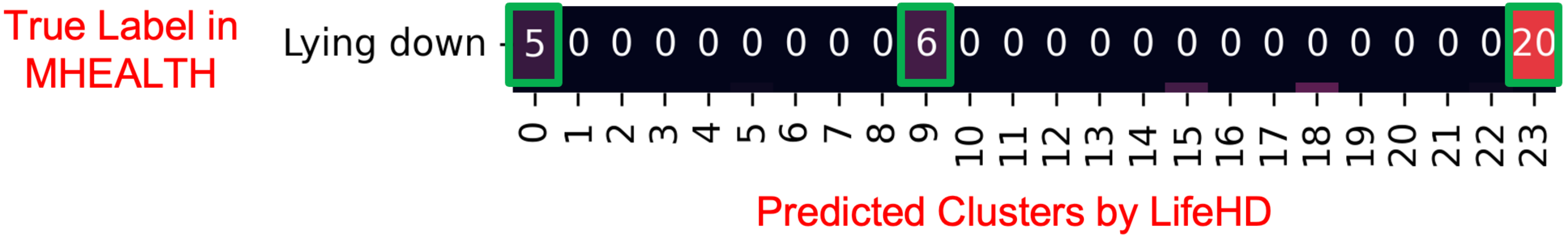
- All NN-based baselines start from higher ACC but experience forgetting
- LifeHD achieves up to **9.4%, 74.8% and 11.8%** accuracy increase on MHEALTH, ESC-50 and CIFAR-100 compared to NN-based baselines

LifeHD vs. Supervised HDC

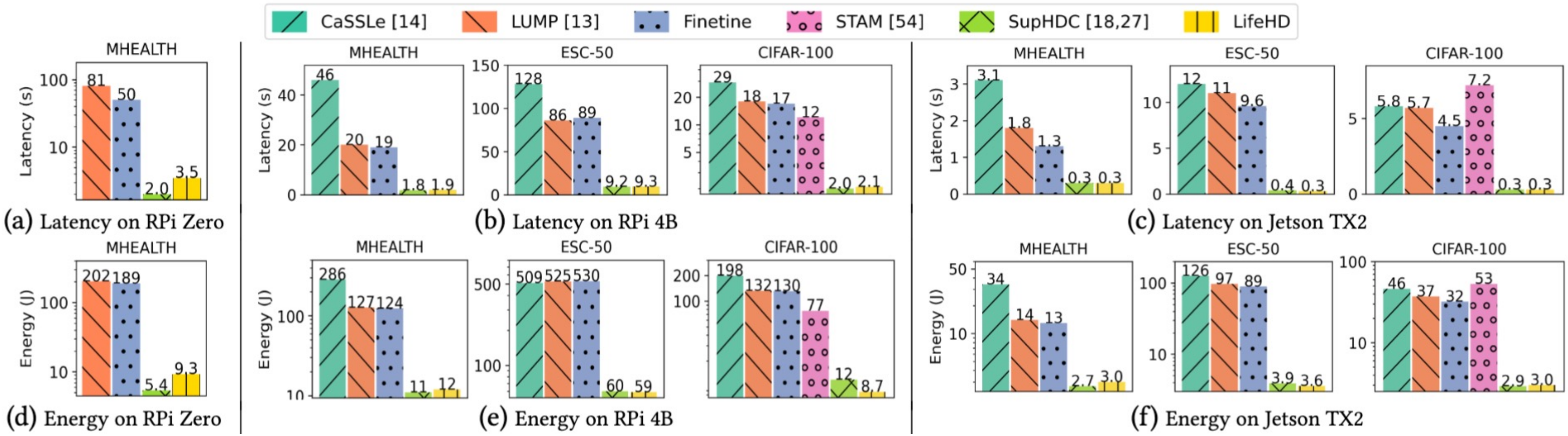
The gap of final ACCs: LifeHD vs. Supervised HDC

Method	MHEALTH	ESC-50	CIFAR-100
LifeHD	0.75	0.92	0.2
Supervised HDC	0.9	0.95	0.26
Gap	-0.15	-0.03	-0.06

- LifeHD approaches the ACC of supervised HDC with a gap of **15%, 3% and 6%** on MHEALTH, ESC-50 and CIFAR-100
- Visualization of a valid learning outcome



Training Latency and Energy



- LifeHD vs. NN-based baselines
  - Up to 23.7x, 36.5x and 22.1x faster to train on RPi Zero, RPi 4 and Jetson TX2
  - Up to 22.5x, 34.3x and 20.8x more energy efficient on RPi Zero, RPi 4 and Jetson TX2

Conclusion

- On-device lifelong learning should be the future of edge intelligence
- Prior works require label supervision or intensive resources to train
- We design and implement LifeHD, the first end-to-end system for on-device unsupervised lifelong learning using Hyperdimensional Computing
- We further propose two variants of LifeHD to deal with practical scenarios

- LifeHD improves ACC by up to 74.8% compared to the SOTA NN-based unsupervised lifelong learning baselines with as much as 34.3x better energy efficiency on Raspberry Pi 4B
- Our code is available at https://github.com/Orienfish/LifeHD

References

- McCloskey, Michael, and Neal J. Cohen. "Catastrophic interference in connectionist networks: The sequential learning problem." Psychology of learning and motivation. Vol. 24. Academic Press, 1989. 109-165.
- Enrico Fini, et al. Self-Supervised Models are Continual Learners. CVPR'22
- Divyam Madaan, et al. Representational Continuity for Unsupervised Continual Learning. ICLR'22
- James Smith, et al. Unsupervised Progressive Learning and the STAM Architecture. IJCAI'21
- Shen, Yang, Sanjoy Dasgupta, and Saket Navlakha. "Algorithmic insights on continual learning from fruit flies." arXiv preprint arXiv:2107.07617 (2021)
- Bricken, Trenton, et al. "Sparse distributed memory is a continual learner.", ICLR'23
- Moin, Ali, et al. "A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition." Nature Electronics 4.1 (2021): 54-63
- Dutta, Arpan, et al. "Hdnn-pim: Efficient in memory design of hyperdimensional computing with feature extraction." Proceedings of the Great Lakes Symposium on VLSI 2022. 2022.
- Imani, Mohsen, et al. "Semihd: Semi-supervised learning using hyperdimensional computing." ICCAD'19
- Khaleghi, Behnam, Mohsen Imani, and Tajana Rosing. "Prive-hd: Privacy-preserved hyperdimensional computing." DAC'20