Semiconductor Research Corporation

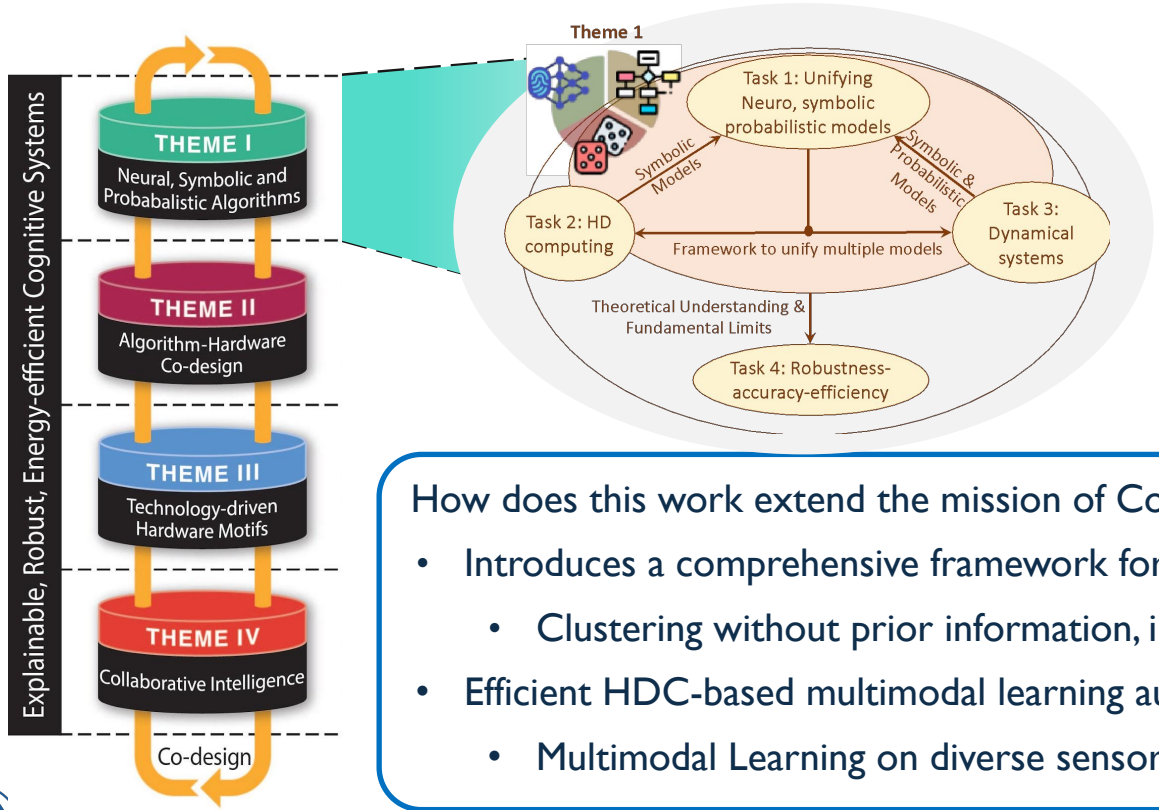
CoCoSys
CENTER FOR THE CO-DESIGN OF COGNITIVE SYSTEMS

Towards Generalized Learning in Hyperdimensional Computing: Density Estimation and Hybrid Models

Quanling Zhao, PI Tajana Rosing

University of California, San Diego (UCSD)

Relevance to the Center Goals



Theme 1 Mission Statement

A principled approach to unified neuro-symbolic-probabilistic algorithms, supported by new information representations, computing models and analysis of fundamental limits, has the potential to radically advance the next generation of cognitive algorithms

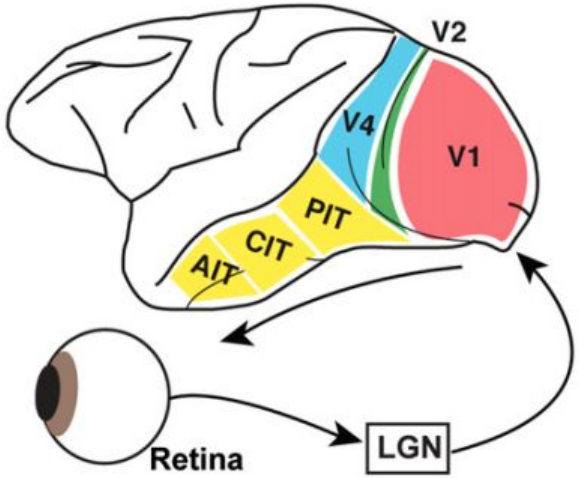
How does this work extend the mission of CoCoSys?

- Introduces a comprehensive framework for density based analysis in HDC
  - Clustering without prior information, image segmentation
- Efficient HDC-based multimodal learning augmented with self-attention
  - Multimodal Learning on diverse sensor data

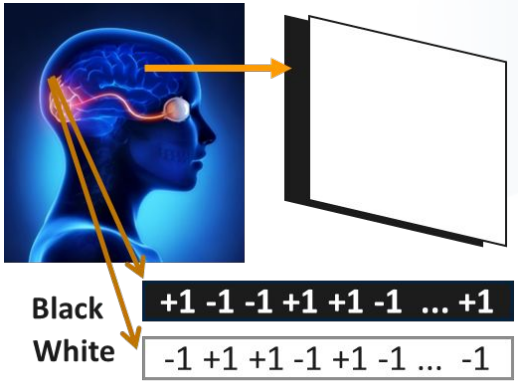
SRC

Hyperdimensional Computing

High dimensional sparse representation



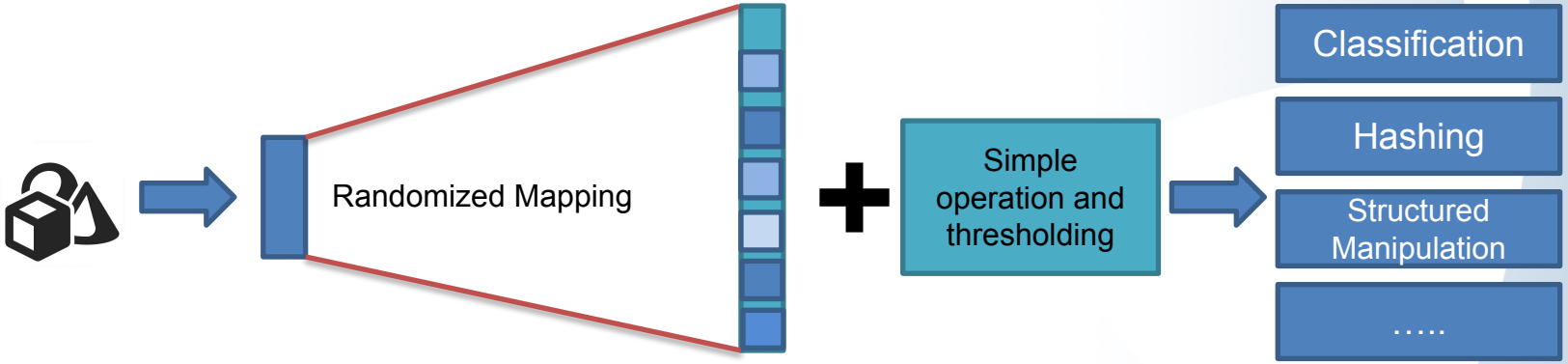
Dense input signal



Hyperdimensional (HD) computing:

- Encodes data into hypervectors
- Leverages full algebra and works on well-defined set of operations that are easy to parallelize
- Fast single-pass training online learning
- Supports symbolic reasoning & explainability
- Energy-efficient & robust to noise

Hyperdimensional Computing

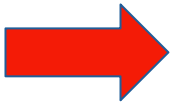


This talk:

- Extending HDC for complex multimodal data
- Enable HDC for density based modeling

This Talk

- **Broaden HDC applications to image segmentation, clustering etc.**



Random Feature Mean-Shift (RFMS)

- Density based analysis HDC representation

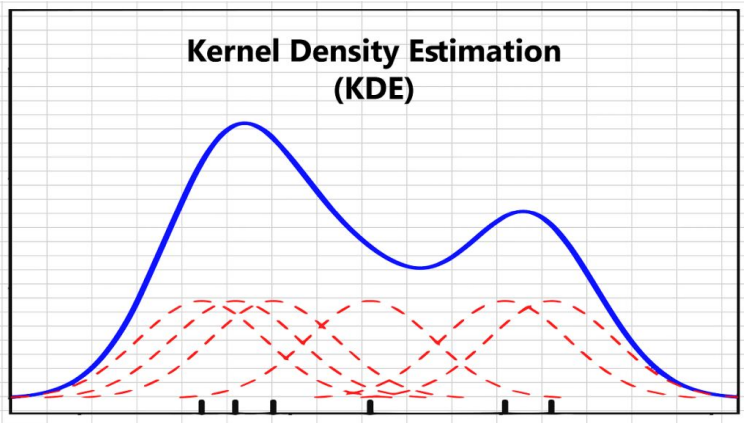
- **Enhance HDC representation, enable learning on diverse sensor data**

MultimodalHD

- NN-HDC hybrid architecture for multimodal data

Background: Density Estimation

- Density estimation is a fundamental tool in many areas of ML
  - Given observed data – how do we assess its likelihood ?



Kernel function

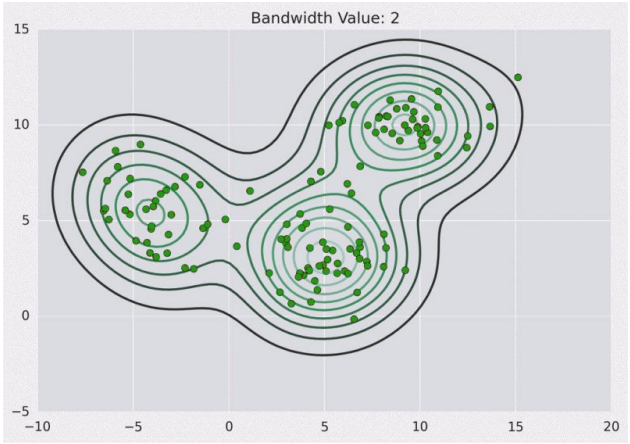
$$\mathrm{KDE}(x) \propto \sum_{i=1}^{n} k\,(x - x_i)$$

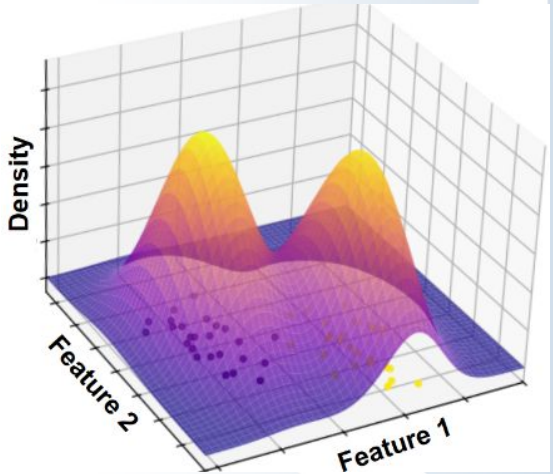
Likelihood

Observation

Background: Mean-Shift

- Mean-Shift: Iteratively shifts each data point towards regions of higher density (as defined by KDE)
  - Converge at modes
  - De-facto algorithm for mode-seeking





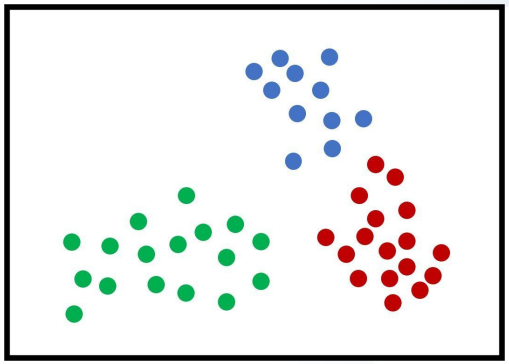
Kernel density estimation over synthetic 2D data(with 2 clusters) using Gaussian kernel.

SRC

Applications of mean-shift

- Locating the modes of a probability density function is a fundamental problem in many areas of machine learning

Image smoothing, segmentation region proposal

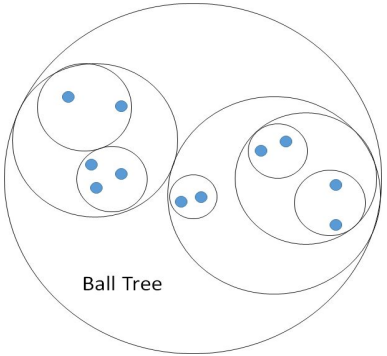


Clustering

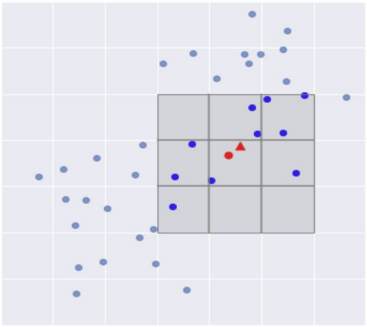
- Mean-Shift: Quadratic complexity with the number of data points observed

SOTA Mean-shift acceleration

- Efficient data structures for distance query
  - Ball-tree, Kd-tree, dual tree [JEA'21]
  - Not improve worst-case complexity



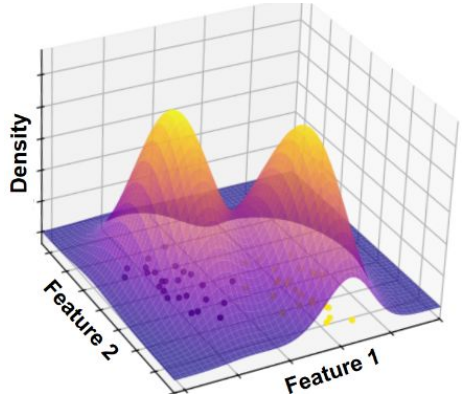
- Discretize representation space
  - MS++ [CVPR'21]
  - Lose resolution
  - Does not scale well into higher dimension



None have succeeded in addressing the complexity bottleneck asymptotically without discretizing data representation in some fashion.

Mean-shift as normalized gradient ascent

- Simple geometric intuition behind mean-shift:
  - Gradient ascent over KDE
  - Normalized by local density - decreasing learning rate



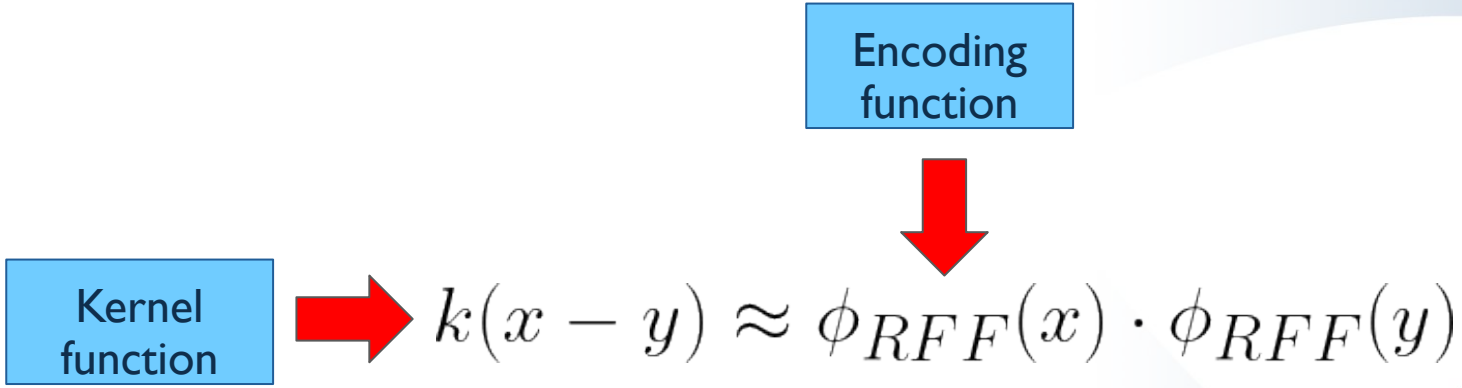
Our Objective

Construct sampling based estimator for
(1) Kernel density
(2) KDE Gradient
That achieves **O(n)** per iteration

$$x^{next} = x^{current} + \eta \frac{\nabla \mathrm{KDE}(x^{current})}{\mathrm{KDE}(x^{current})}$$

HDC as kernel estimator

- How could HDC be used in Mean-Shift
- Important observation: KDE is just a summation of kernel valuesIf



HDC as kernel estimator

- How could HDC be used in Mean-Shift
- Important observation: KDE is just a summation of kernel valuesIf

$$k(x-y) \approx \phi_{RFF}(x) \cdot \phi_{RFF}(y)$$

(Random Fourier Feature)
It is possible to generate HDC encodings such that:

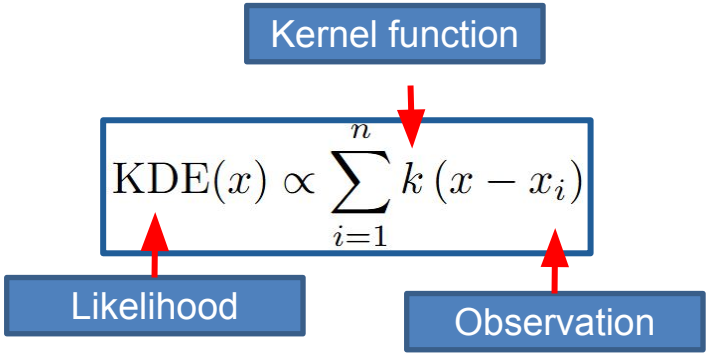
Monte-Carlo

$$k(x-y) = \int_{\mathbb{R}^d} p(\omega) \cos(\omega \cdot (x-y)) d\omega$$

- Often called "Nonlinear Encoding" in the HDC literature
  - Random feature typically drawn from Gaussian and approximates the Gaussian kernel.

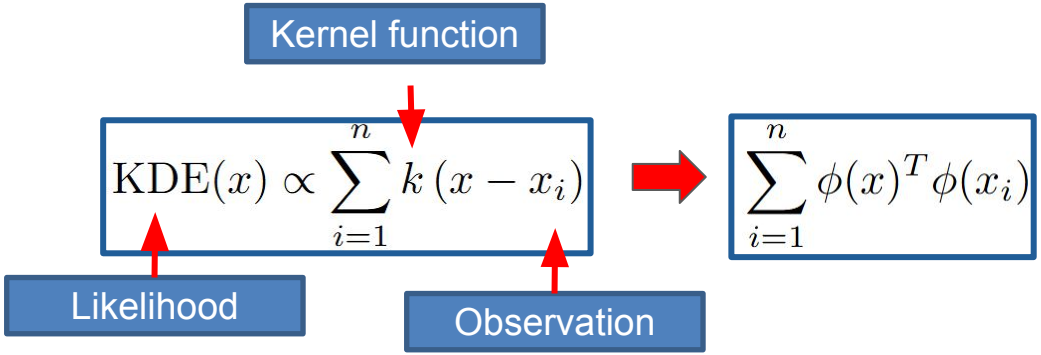
RFMS: Random Feature Density Estimator

- We use HDC encoding function such that: $\langle \phi(x), \phi(y) \rangle \approx k(x, y)$



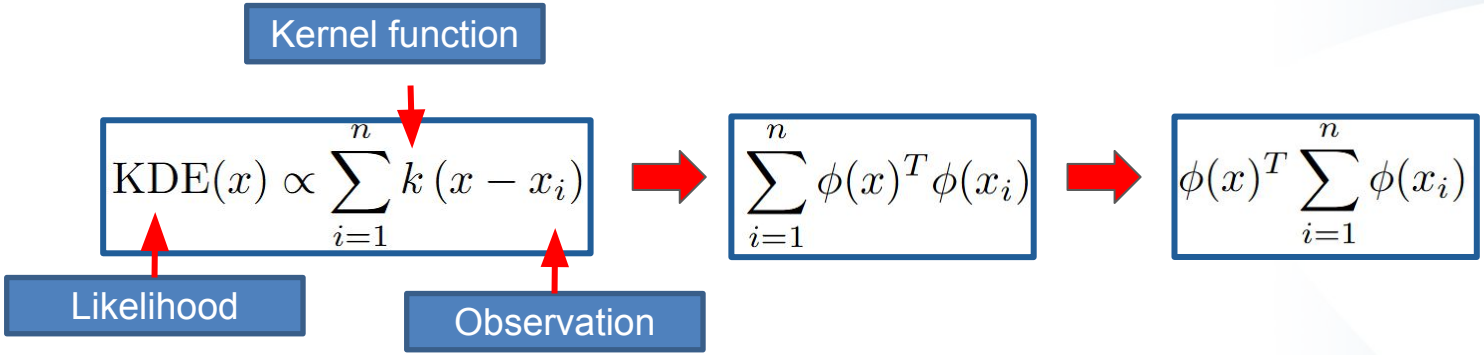
RFMS: Random Feature Density Estimator

- We use HDC encoding function such that: $\langle \phi(x), \phi(y) \rangle \approx k(x, y)$



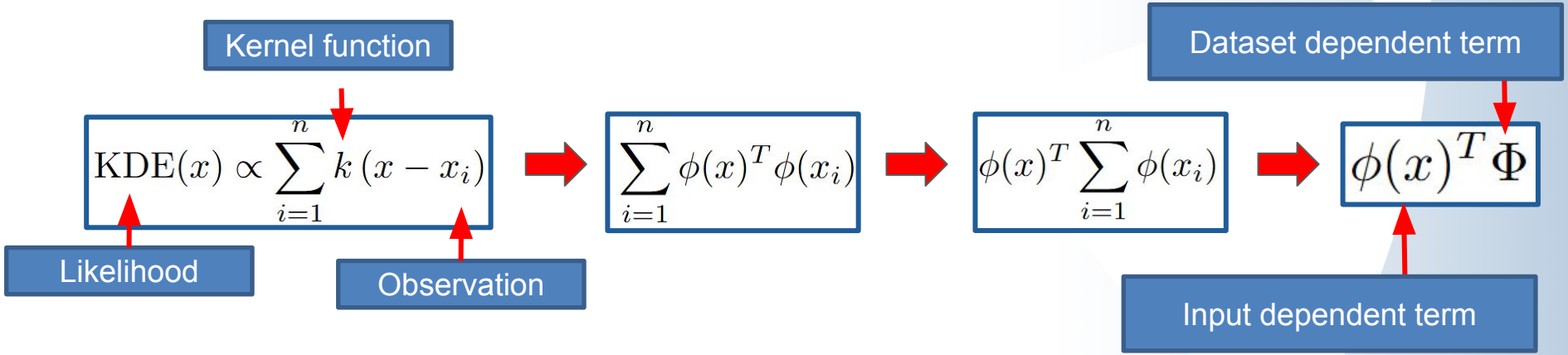
RFMS: Random Feature Density Estimator

- We use HDC encoding function such that: $\langle \phi(x), \phi(y) \rangle \approx k(x, y)$



RFMS: Random Feature Density Estimator

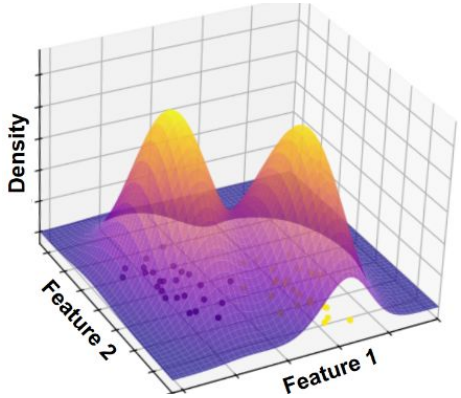
- We use HDC encoding function such that: $\langle \phi(x), \phi(y) \rangle \approx k(x, y)$



- Given we stores dataset dependent term in advance, cost density estimation reduced to **O(1) form O(n)**

RFMS: Zeroth-order Gradient Ascent

- **Context**



$$x^{next} = x^{current} + \eta \frac{\nabla \mathrm{KDE}(x^{current})}{\mathrm{KDE}(x^{current})}$$

- **Needs**

(1) Kernel density ✅
(2) KDE Gradient

- Consider density estimator a oracle
- We can estimate gradient by making 2 queries to the density estimator

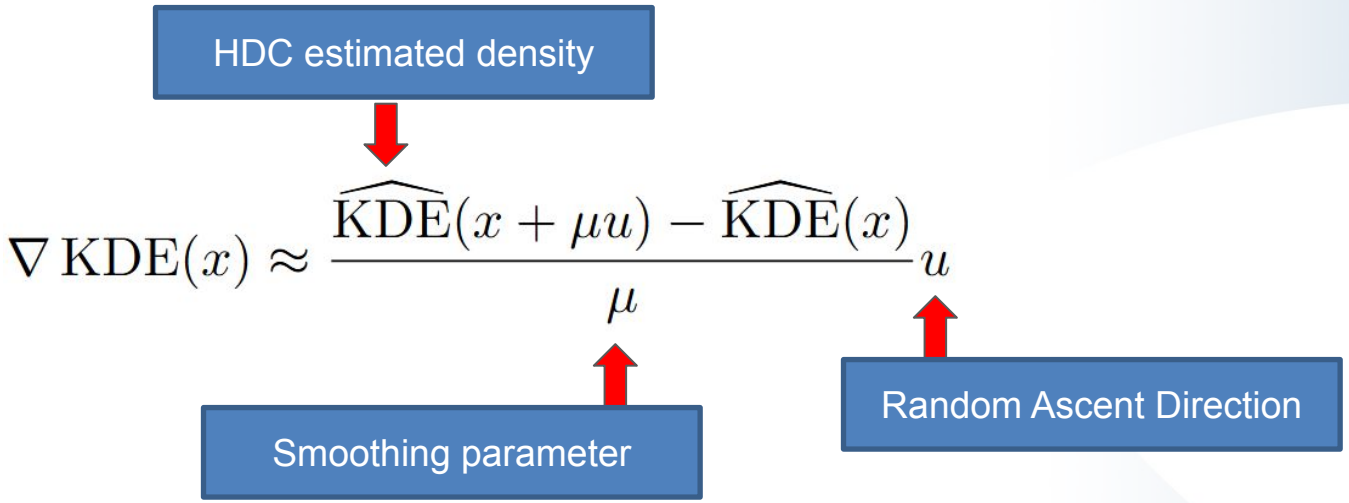
SRC

RFMS: Zeroth-order Gradient Ascent

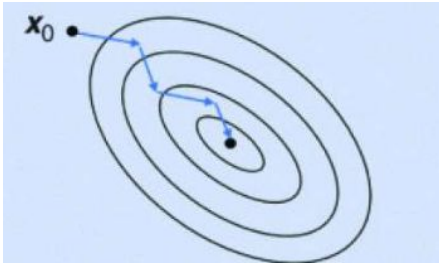
HDC estimated density

$$\nabla \mathrm{KDE}(x) \approx \frac{\widehat{\mathrm{KDE}}(x + \mu u) - \widehat{\mathrm{KDE}}(x)}{\mu} u$$

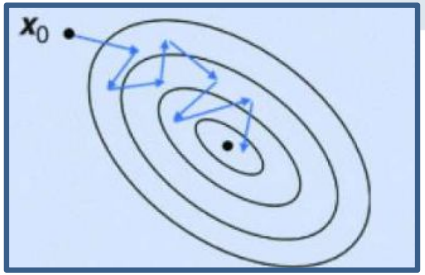
RFMS: Zeroth-order Gradient Ascent



RFMS: Zeroth-order Gradient Ascent



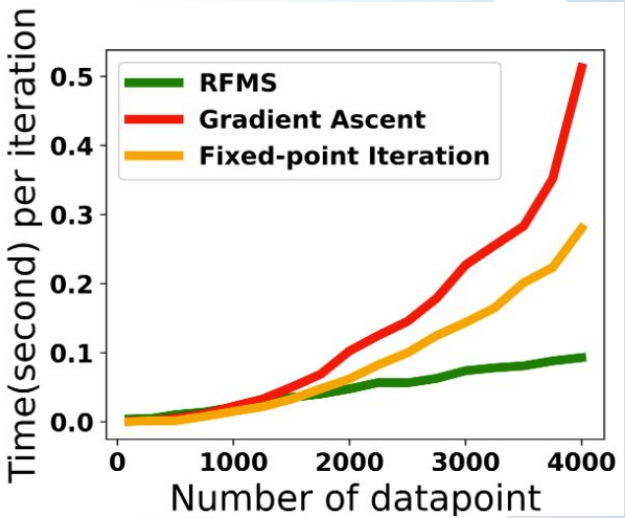
First order gradient



Zeroth order gradient

RFMS: Complexity

Operation	Mean-Shift	RFMS
Density (per point)	O(nd)	O(D)
Gradient (per point)	O(nd)	O(D)
Shift (per iteration)	O(n^2d)	O(nD)



- RFMS estimates the mean-shift update date via two random feature density estimations
- Sampling based estimator, different from data structure and hashing based methods.

Mode Estimation Bounds

- For sufficiently large dimension, conditions will be satisfied almost surely

$$\hat{x}^* \in B\left(x^*, \frac{\epsilon_1}{\lambda_{min}\left(\nabla^2 \hat{f}_{k_h}(x^*)\right) - \epsilon_2}\right)$$

Mode "closeness" based on dimensionality and mode "flatness"

- For every mode of original KDE, there will be a mode of approximated KDE nearby

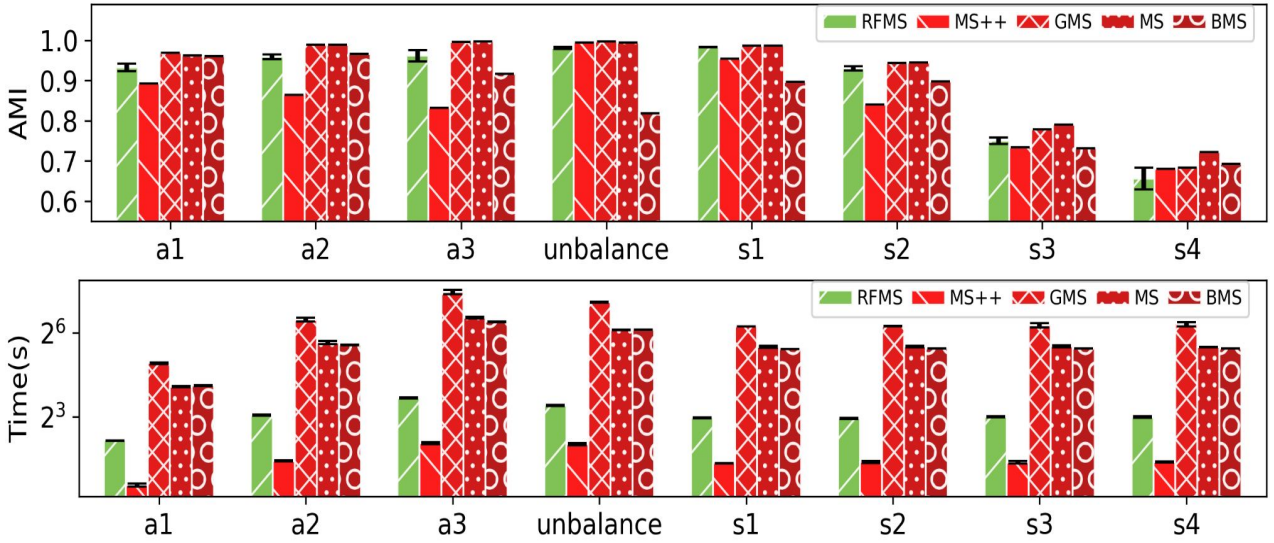
Intuition: Bounded difference for gradient and hessian implies mode stability (Morse theory). Apply Newton-Kantorovich theorem for the concrete bound

Evaluation Setup

- Baselines
  - Mean-Shift (MS) [JPDC'19,IEEE Access'22] - De-facto choice for mode-seeking
  - Gradient Mean-Shift (GMS) [IDA'07,TIT'75]
  - Blurring Mean-Shift (BMS) [AAA1'21,JoC'16]
  - MS++ [CVPR'21]
  - QuickShift [CEA'23,ECCV'08]
- Dataset
  - 8 benchmarking datasets for clustering [SoftwareX'22]
  - Berkeley Segmentation Dataset Benchmark [CVPR'21,ICCV'01]
    - 150k data per points, impractical for normal mean-shift
- For all methods we run 100 iterations with algorithm for clustering assignment
- For RFMS we set D=300 for clustering and D=10 for image segmentation
  - D controls trade-offs between approximation quality and efficiency

Clustering Results

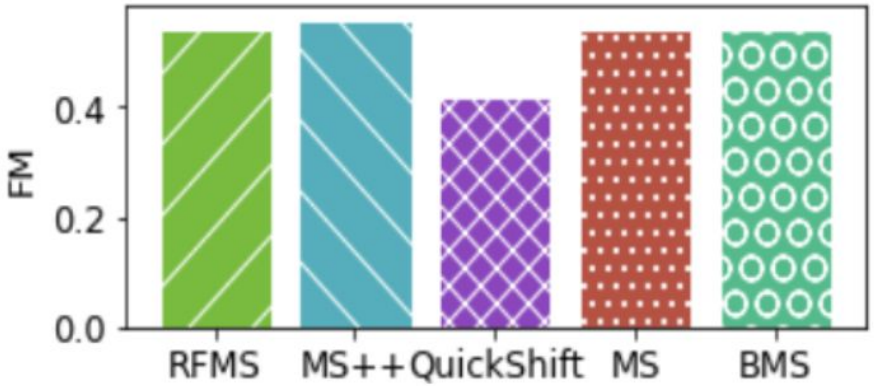
- Density based clustering
  - Differ from previous HDC clustering method where the number of cluster needs to be known in advance

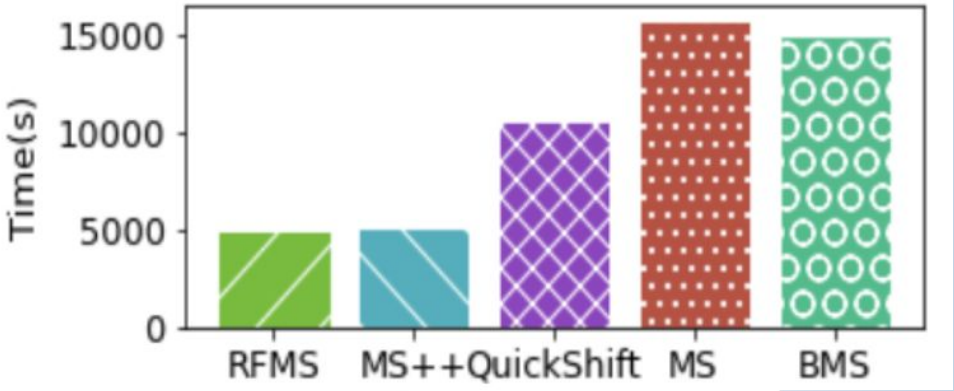


- Evaluate quality with Normalized Mutual Information (NMI) and time for 100 iteration (in second)
- Significantly more efficient than conventional mean-shift algorithms (MS, BMS, and GMS), with up to a 14x speed-up
- Slightly slower than MS++ (On average 5.6s slower), RFMS produced better clustering quality in 6 out of 8 datasets

SRC

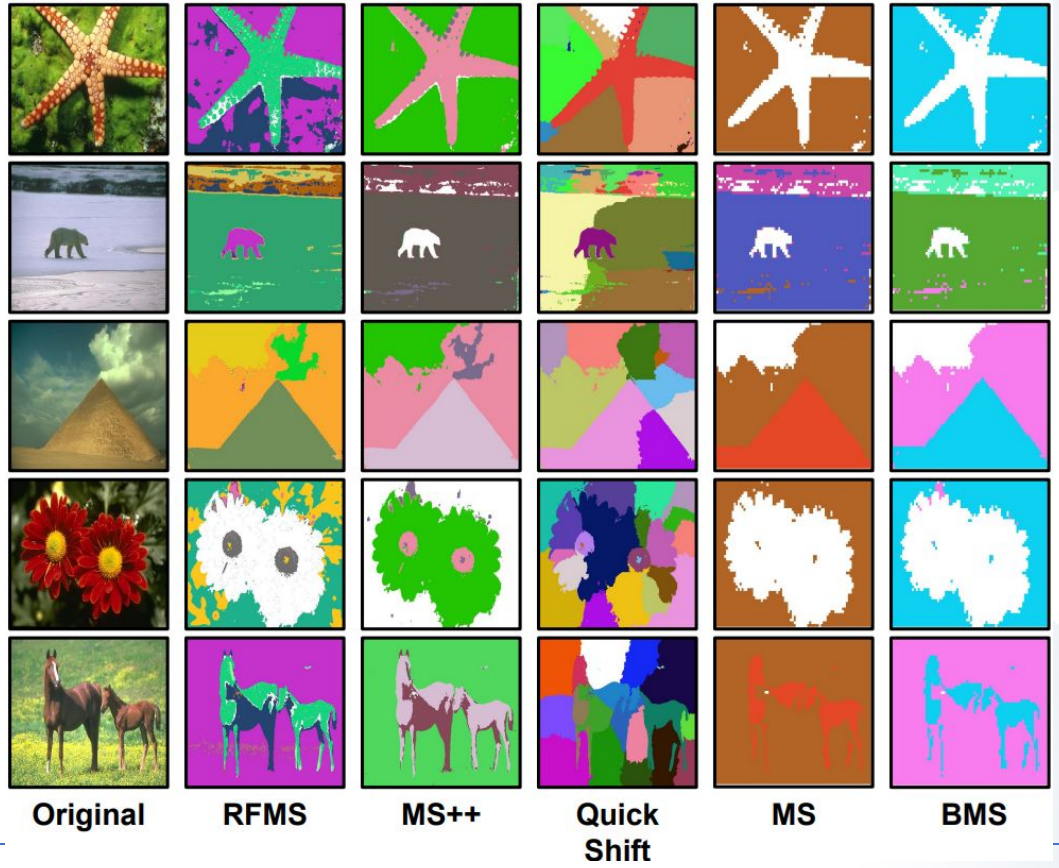
Image Segmentation Results





- Each image is a dataset of 154k points
- MS and BMS were ran on down-sampled image (1/36 size)
- Despite MS and BMS being run on lower-resolution sampled images, RFMS still achieves 3x speedup when compared with MS and BMS, and 2x speedup when compared to QuickShift

Image segmentation (Qualitative)



SRC

This Talk

- **Broaden HDC applications to image segmentation, clustering etc.**

Random Feature Mean-Shift (RFMS)

- Density based analysis HDC representation

- **Enhance HDC representation, enable learning on diverse sensor data**

MultimodalHD

- NN-HDC hybrid architecture for multimodal data

Challenges to Multimodal Learning

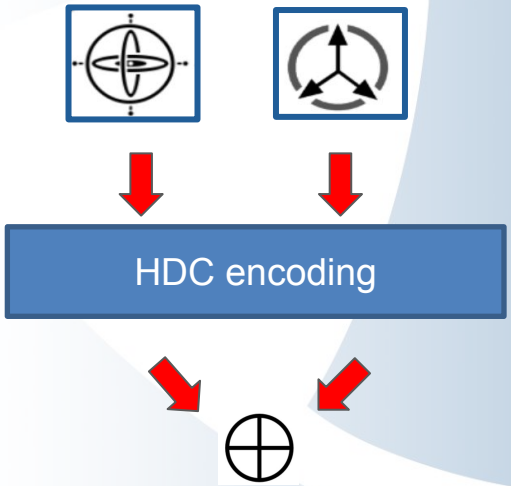
HDC Advantages

- Efficient processing of multimodal sensor signals
- Computational efficient with less memory usage
- How to combine different modalities

Challenges

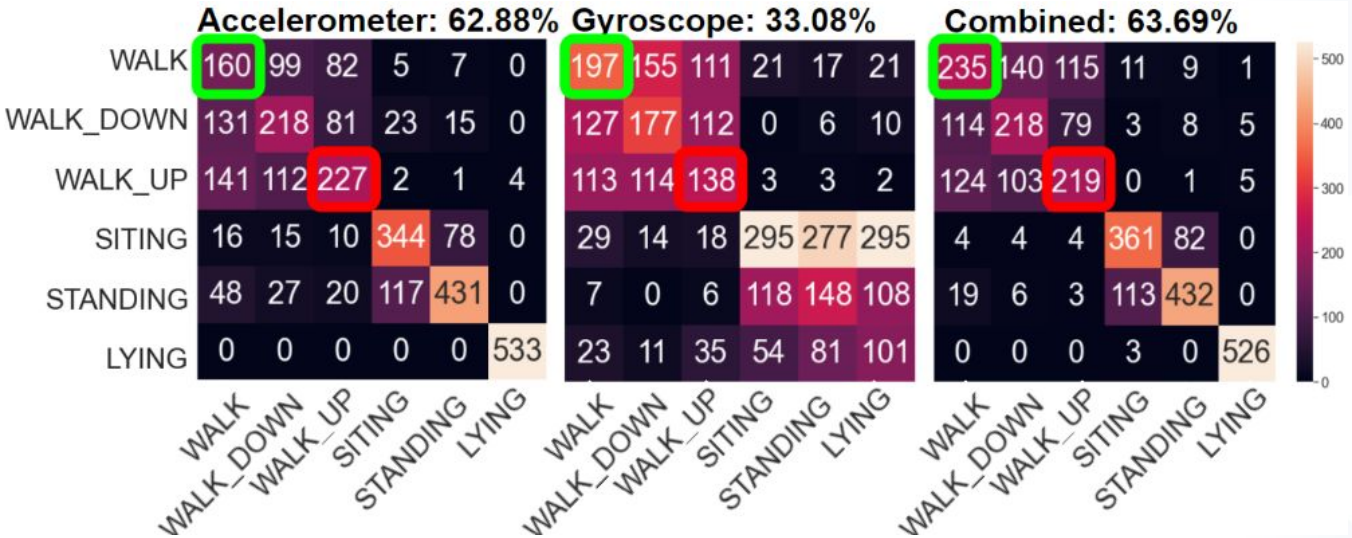
- Data heterogeneity
- Multimodal fusion: how to aggregate the sensor data of same and different modalities

Multimodal Time-series:
[AICAS'19],[IEEE IV'21]



Fusion via boundling

Challenges of using HDC in multimodal data



Green:
Positive Aggregation

Red:
Negative Aggregation

- Main challenge for MFL: learning joint representation by fusing the information from different sensing modalities
- Previous HDC literature superimposes hypervectors from different modality
  - Can lead to negative fusion results

SRC

SOTA Multimodal Federated Learning (MFL)

Method	Modality Heterogeneity	Personalization	Hardware Efficiency
Multimodal FL	✓	×	×
MMFed	limited	✓	×
FedMSplit	✓	✓	×
MultimodalHD (this paper)	✓	✓	✓

Multimodal-FL [Neurcomp'22]

- **Multimodal Split Autoencoder**
  - Employs a split autoencoder on each client to learn from multiple modalities without supervision

MMFed [IoTDI'22]

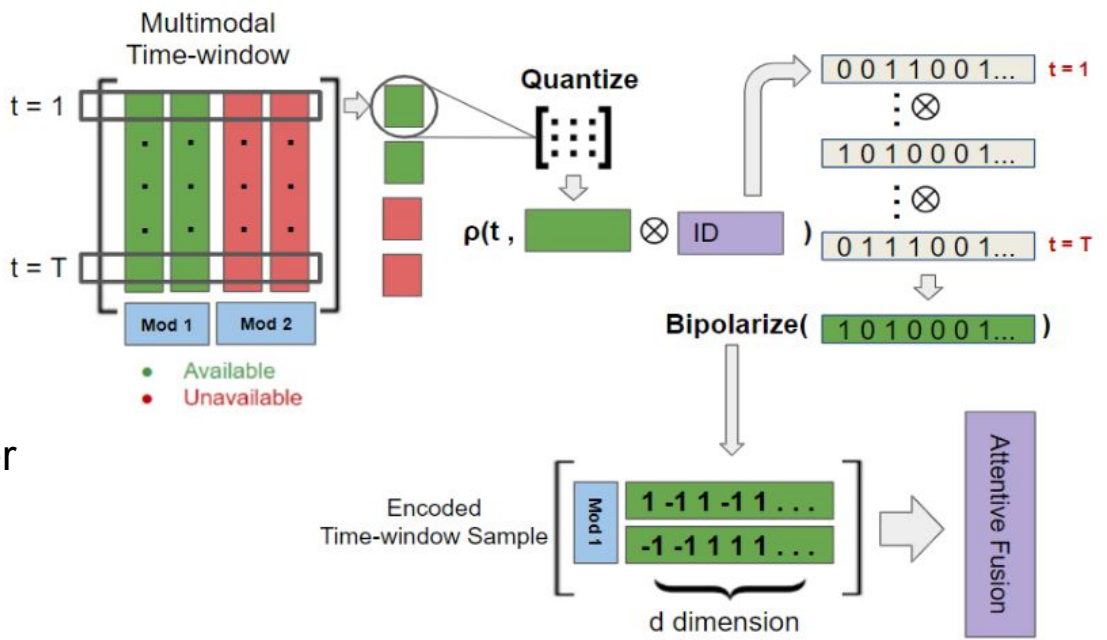
- **Modality co-attention**
  - Uses co-attention mechanism between modalities

FedMSplit [SIGKDD'22]

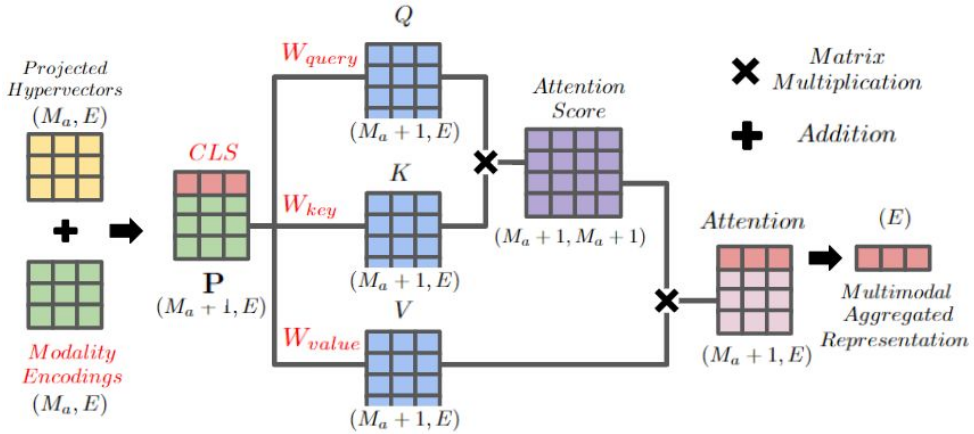
- **Split neural network**
  - Dynamic graph structure to adaptively capture the relationships among different types of clients

Encoding time-series sensor data with HDC

- Encoding requires no training
- ID hypervectors: encode sensor type
- Level hypervectors: encode real valued sensor readings to hypervector
- Use permutation to encode temporal information
- Supports modality heterogeneity
- All clients have the same encoder



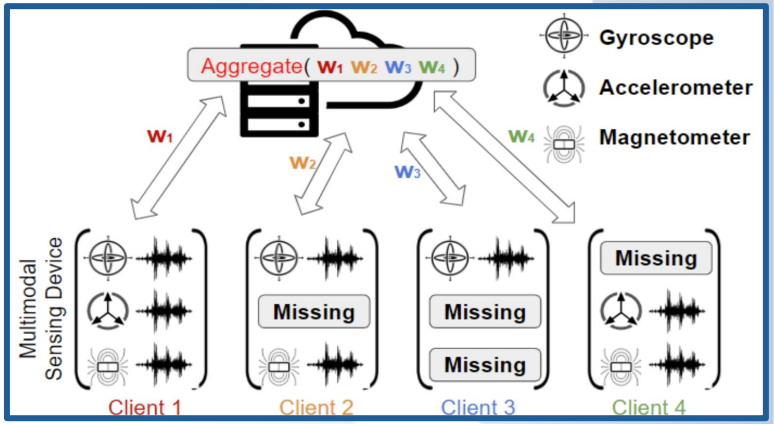
Tokenizing modality information and Attention



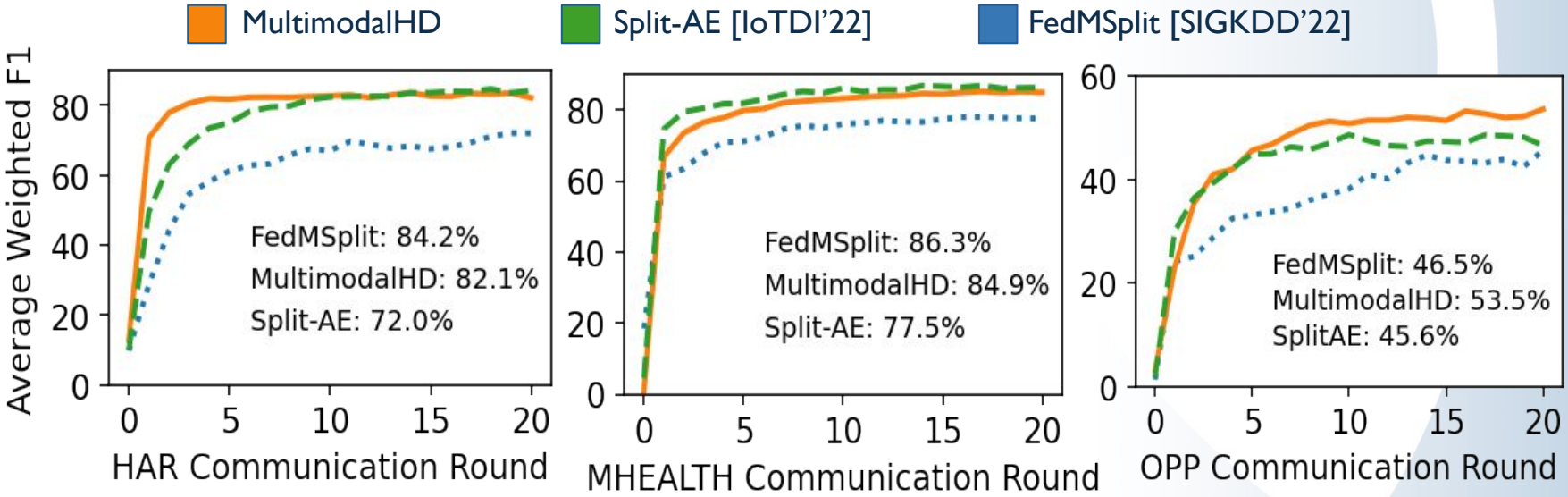
- Adapting intermodal attention to learn a fused representation
- Modality encodings to encourage the model to learn information associated with each modality itself rather than the data from that modality
- Uses CLS token as an attentively aggregated representation of all modalities

Evaluation Setup

- **Baselines**.
  - Split-AE[IoTDI'22]: Uses split-autoencoder to learn and extract correlated representations from different modalities.
  - FedMSplit[SIGKDD'22]: Employ separate blocks for available modalities on the clients and updates the global model based on a dynamically learned graph.
- **Evaluation Platform**: Raspberry Pi 4b
  - Training efficiency measure as training time.
- **Metric**: Following previous studies we use the weighted F1 score as the evaluation metric
  - The overall performance is evaluated by the average F1 score across all client.

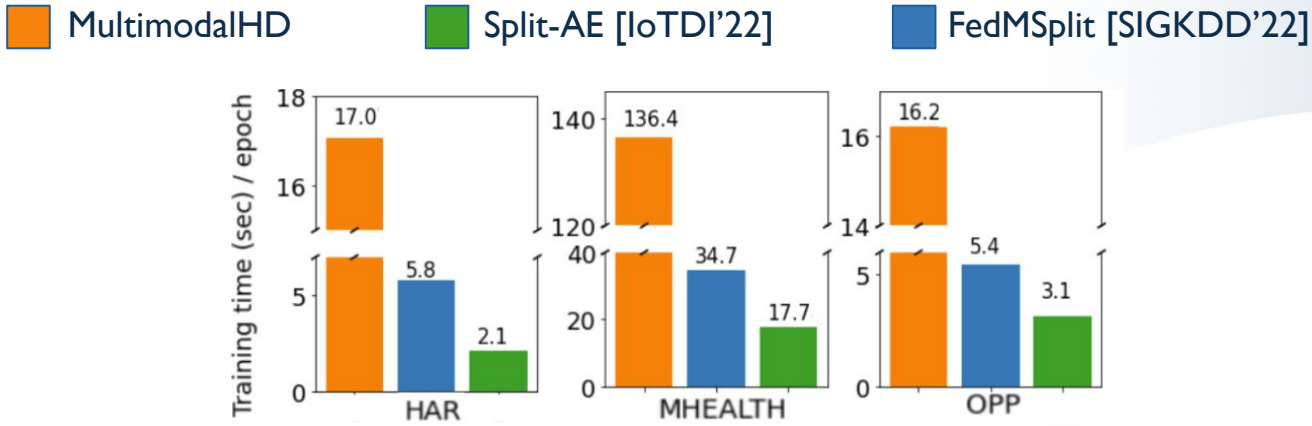


Results: MFL Accuracy



MultimodalHD achieves better or comparable accuracy to SOTA MFL baselines

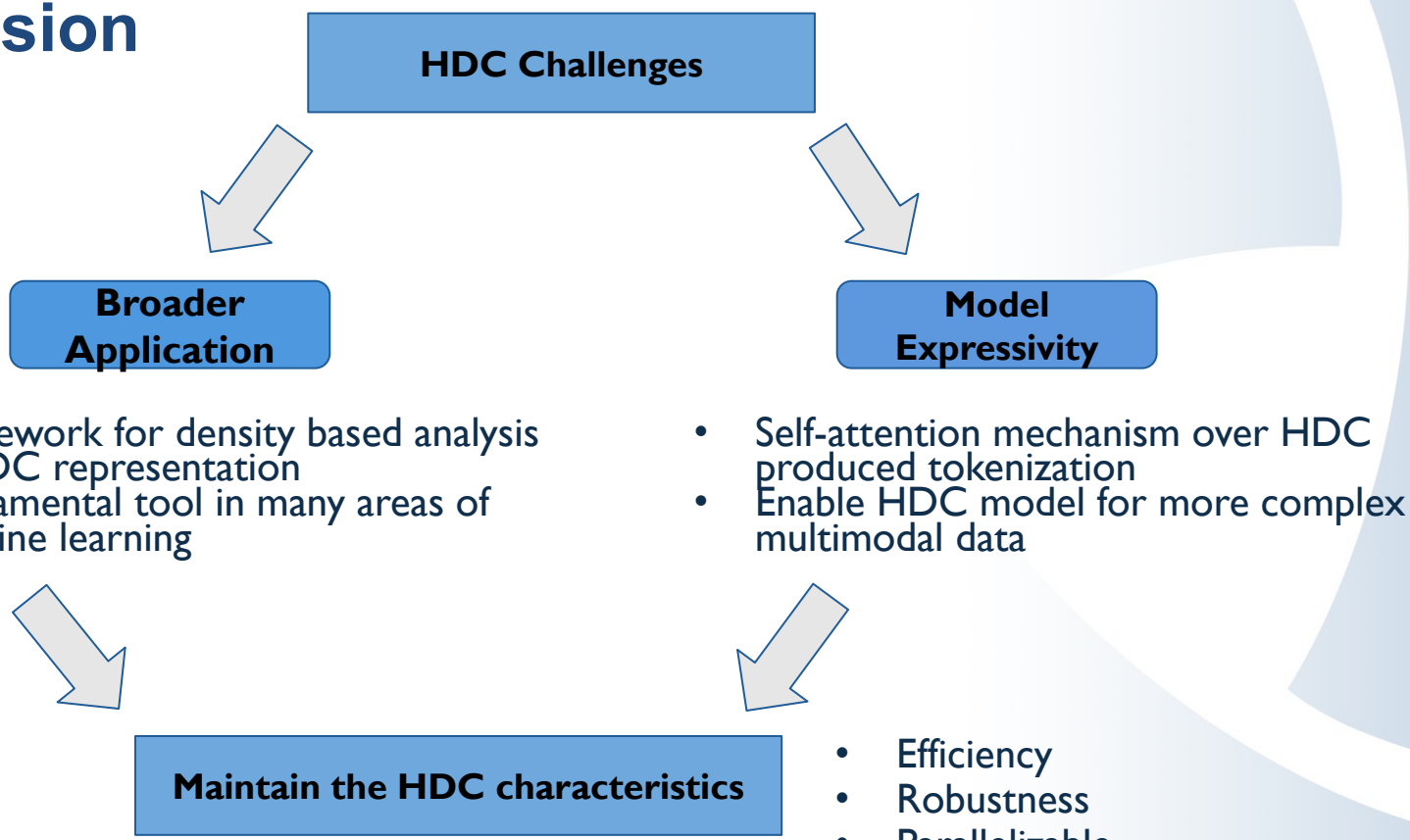
Results: Efficiency



MultimodalHD has up to 8x faster training

- HDC-based methods avoid slow and expensive sequential RNN computation

Conclusion


HDC Challenges
Broader Application
Model Expressivity
• Framework for density based analysis in HDC representation
• Fundamental tool in many areas of machine learning
• Self-attention mechanism over HDC produced tokenization
• Enable HDC model for more complex multimodal data
Maintain the HDC characteristics
• Efficiency
• Robustness
• Parallelizable