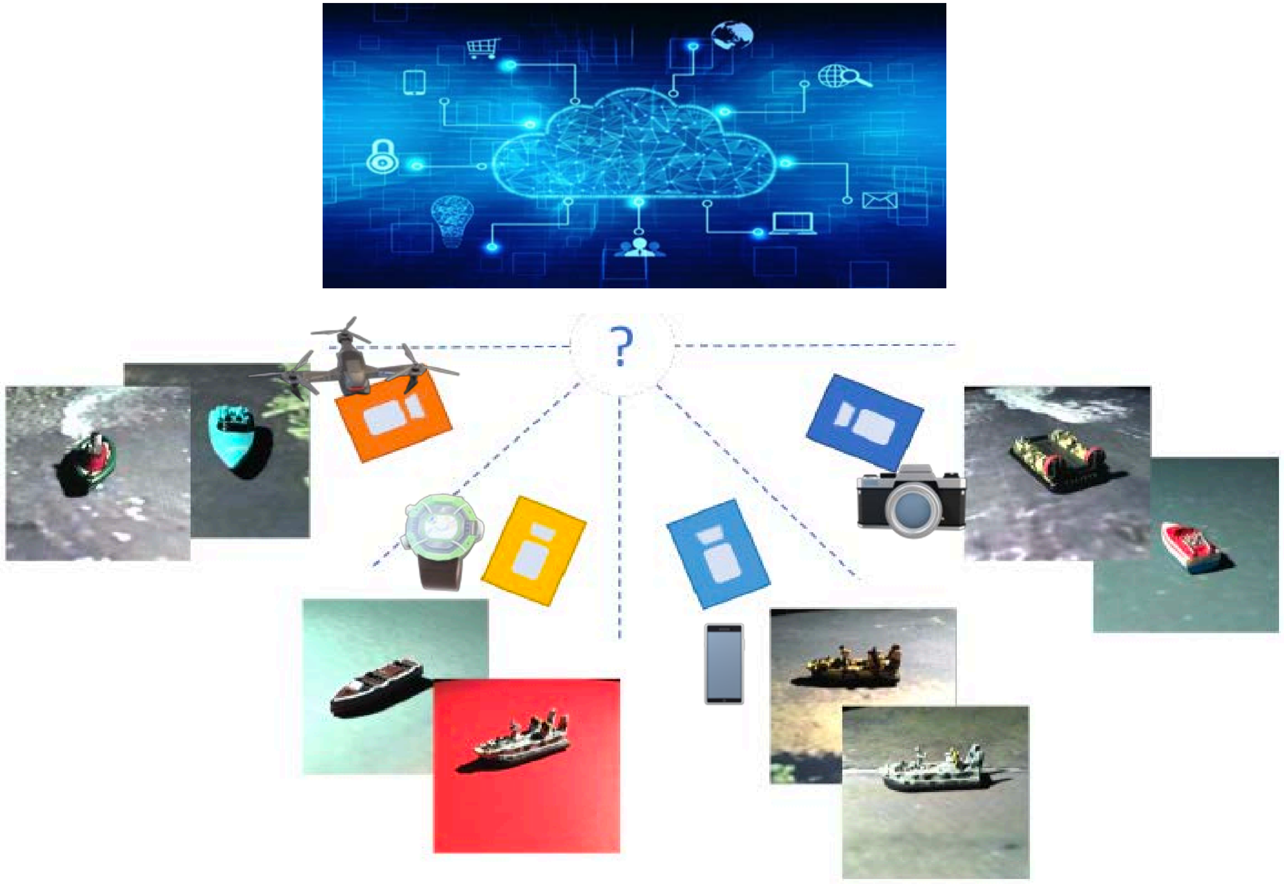
Advances in Distributed Video Analytics

Artifical Intelligence of Things, Singapore

VIJAYKRISHNAN NARAYANAN
THE PENNSYLVANIA STATE UNIVERSITY
SUPPORTED IN PART BY NSF, CRISP AND CBRIC

Object Recognition on Distributed Camera System: Limitations

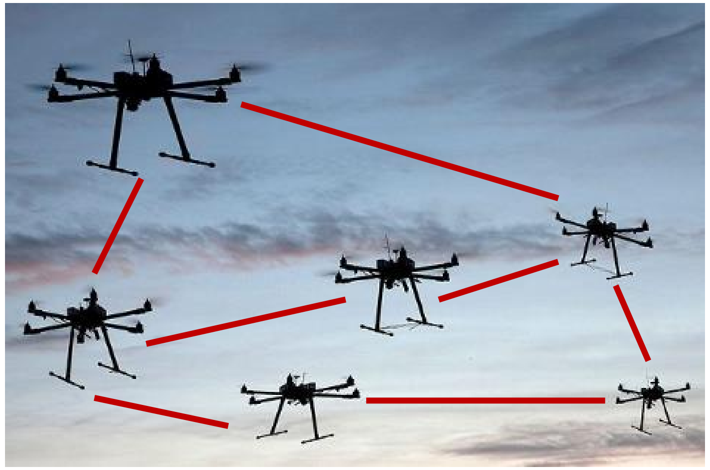


IoT device limitations: Energy and power constraints

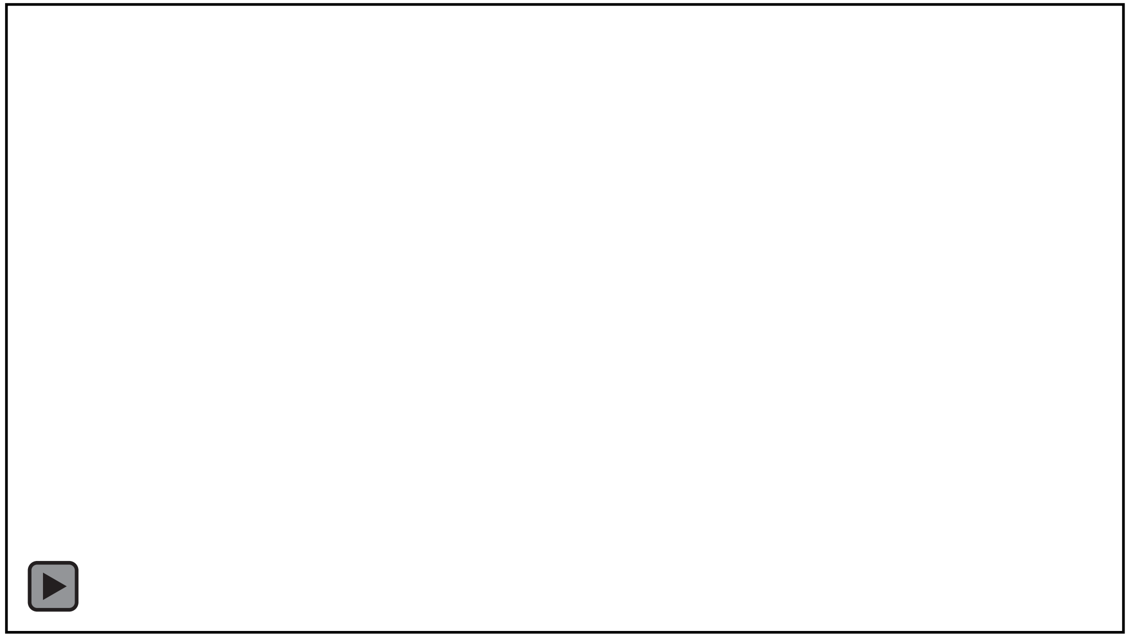
Communication cost bottleneck

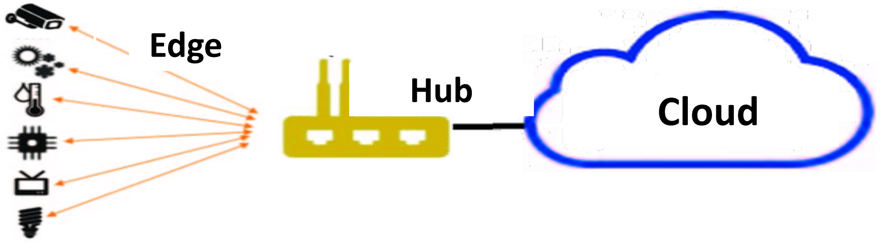
Losing context information

Distributed Intelligence



Peer to Peer

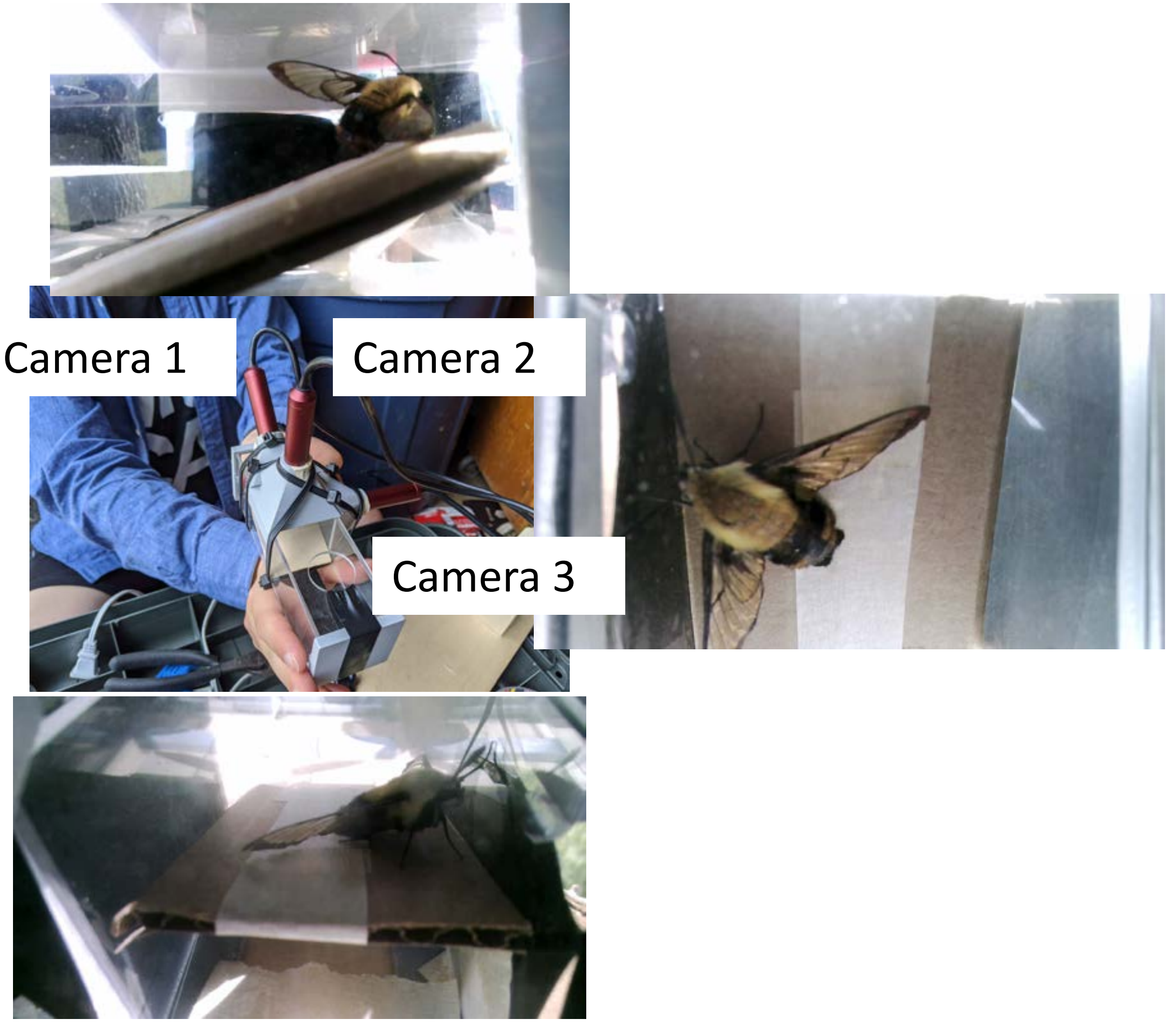




Edge-Cloud Partitioned

PEDRA (Arijit Raychowdhury, CBRIC)

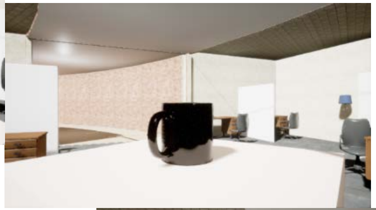
Eco-Friendly Pollinator Trackers



In Store Camera Networks

Visual Assistance System



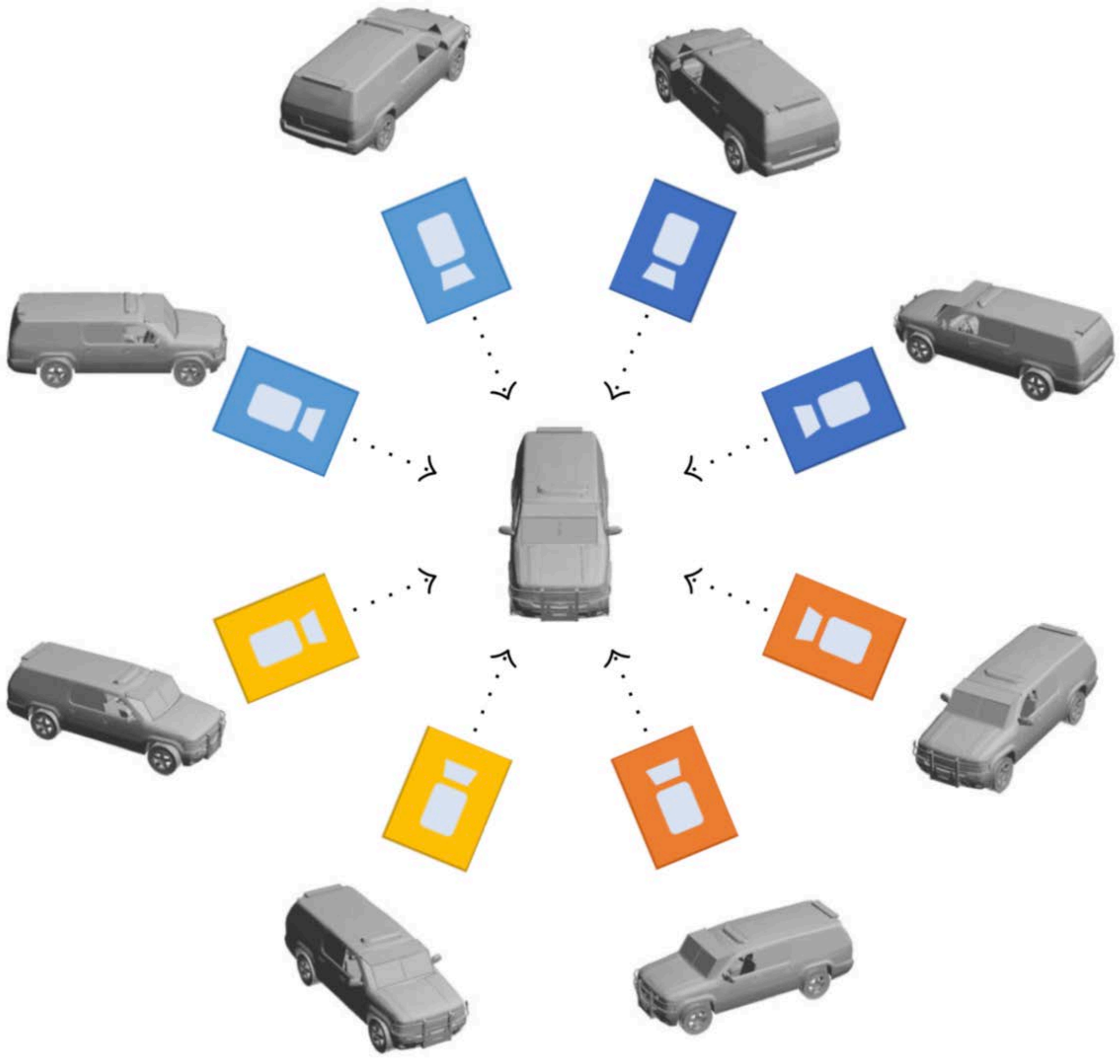


Which views are most useful to recognize object ?
How many views do we need?





Tradeoff - Accuracy & Communication Costs



Test # views	Accuracy
1	70.0%
2	71.2%
4	91.1%
8	93.1%

8x Comm. Traffic increase
before feature aggregation (pooling)

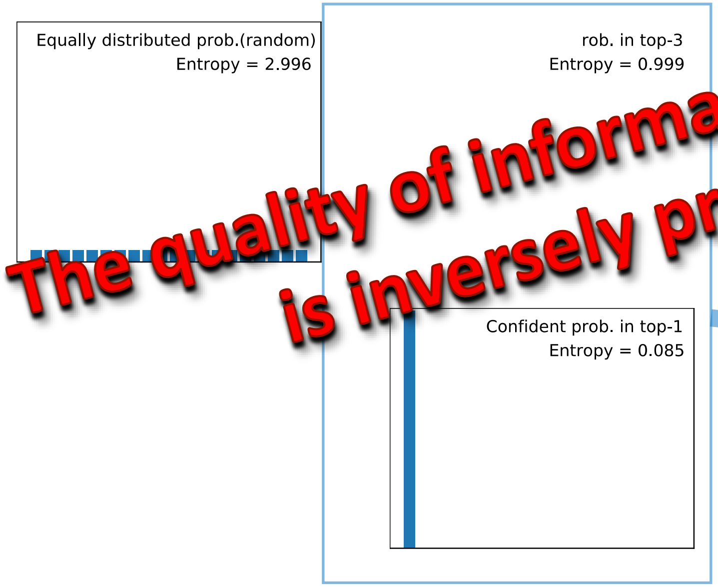
Significant communication traffic incurs
on the side of back-end feature aggregation

*Examples are reproduced from Table 2,
Yifan Feng, et al. "GVCNN: Group-View Convolutional Neural Networks for 3D Shape Recognition," CVPR, 264–272, 2018.

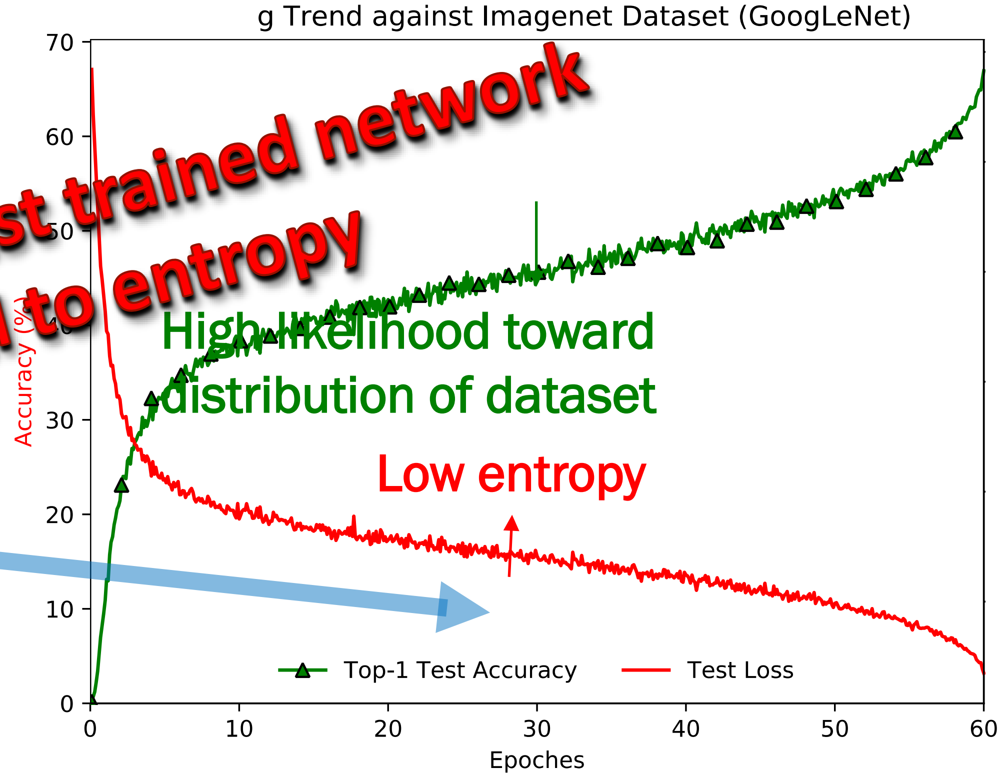
Leveraging Entropy for Context-Awareness

- Against DNN model $y = P(x; \Theta)$,
  Entropy can be defined as

$$\sum_c \sigma\left(-y_c \log(y_c + \varepsilon)\right)$$

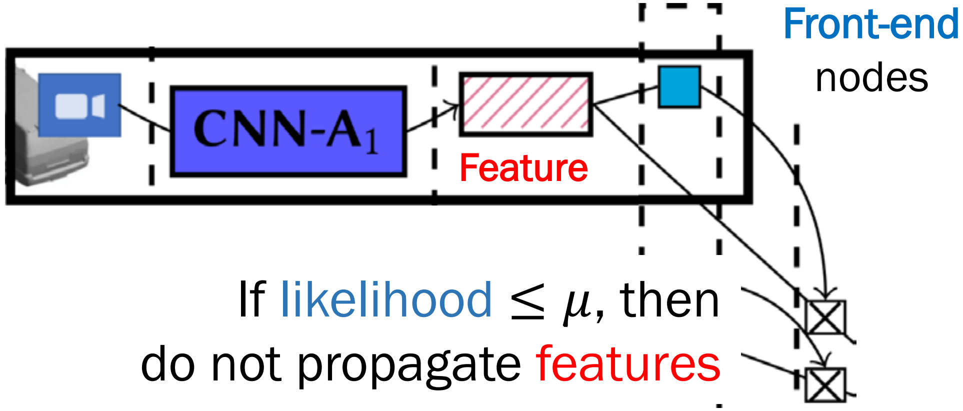


The quality of information against trained network is inversely proportional to entropy

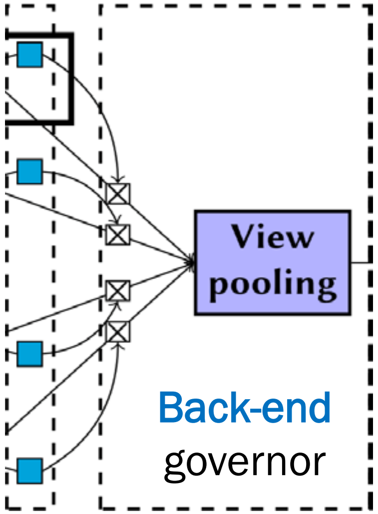


Selective Feature Normalization

- How to give importance?
  - $\delta = \max\left({}^{1}/_{(entropy+\rho)} - \mu, 0\right)$
  - If likelihood $\leq \mu$, then signal back-end to **zero** importance.

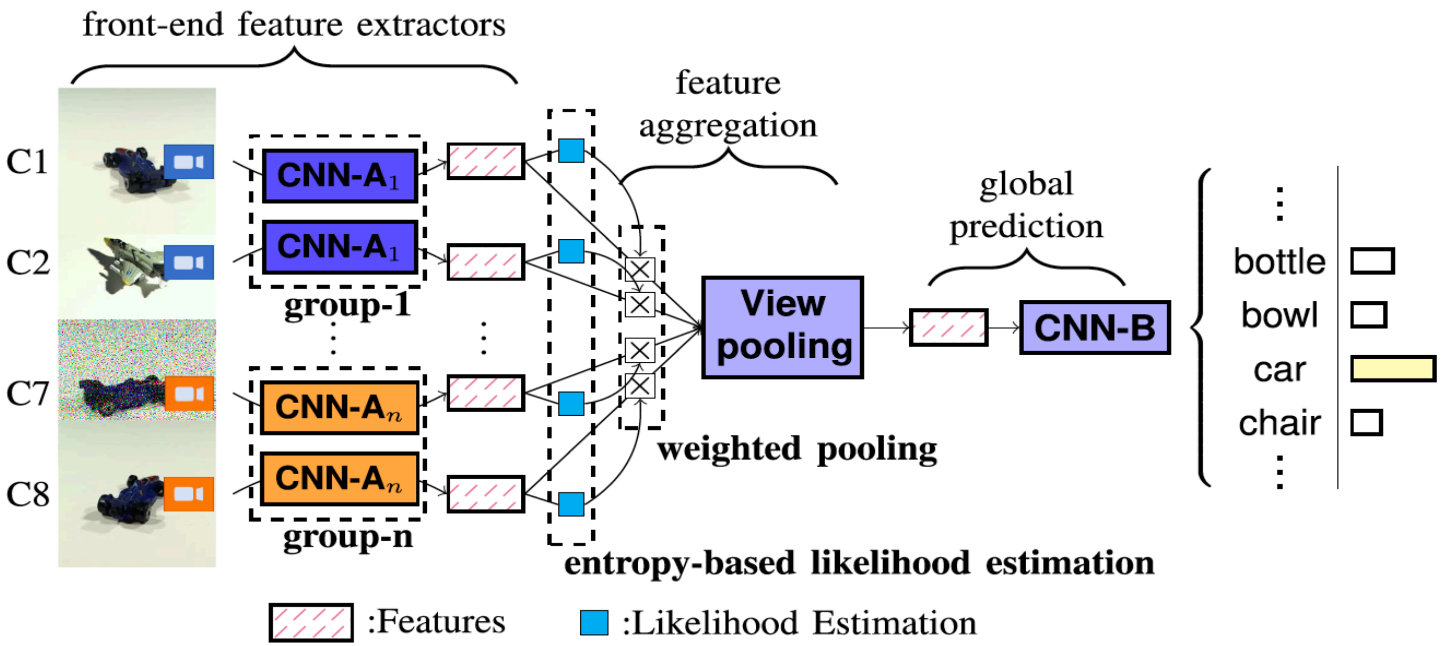


- Feature aggregation in wavg
  - $\sum_i \frac{\delta_i}{\sum_\phi \delta_\phi} X_i$ (skip X_i if $\delta_i = 0$)



- If importance is 0, then skip computation
- Otherwise, accumulate features with respect to normalized importance

Context-Aware Multi-View Camera System

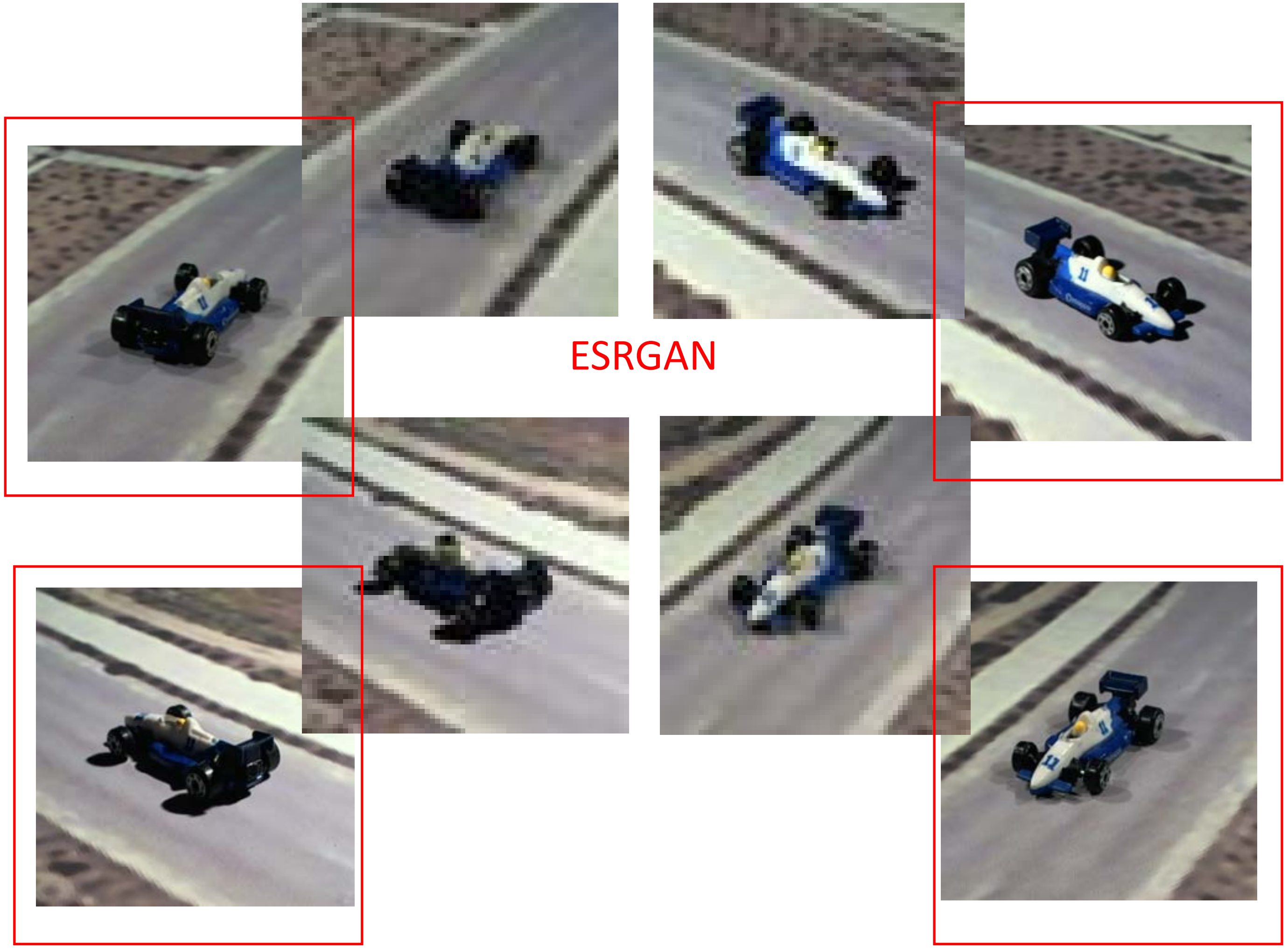


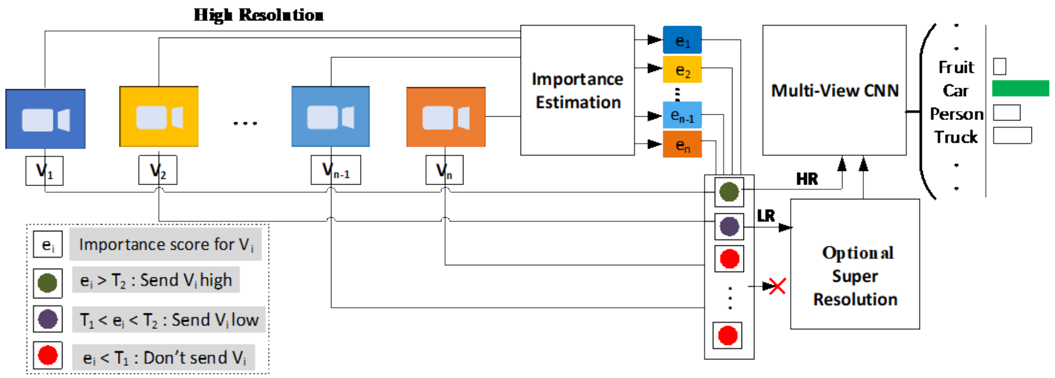
1. Likelihood estimation from entropy
2. Average pooling based on Likelihood weights in feature aggregation

Communication cost reduction by a factor of 3X, while achieving high accuracy

Achieve 8.5%–13.4% accuracy recovery against a large portion of camera failures

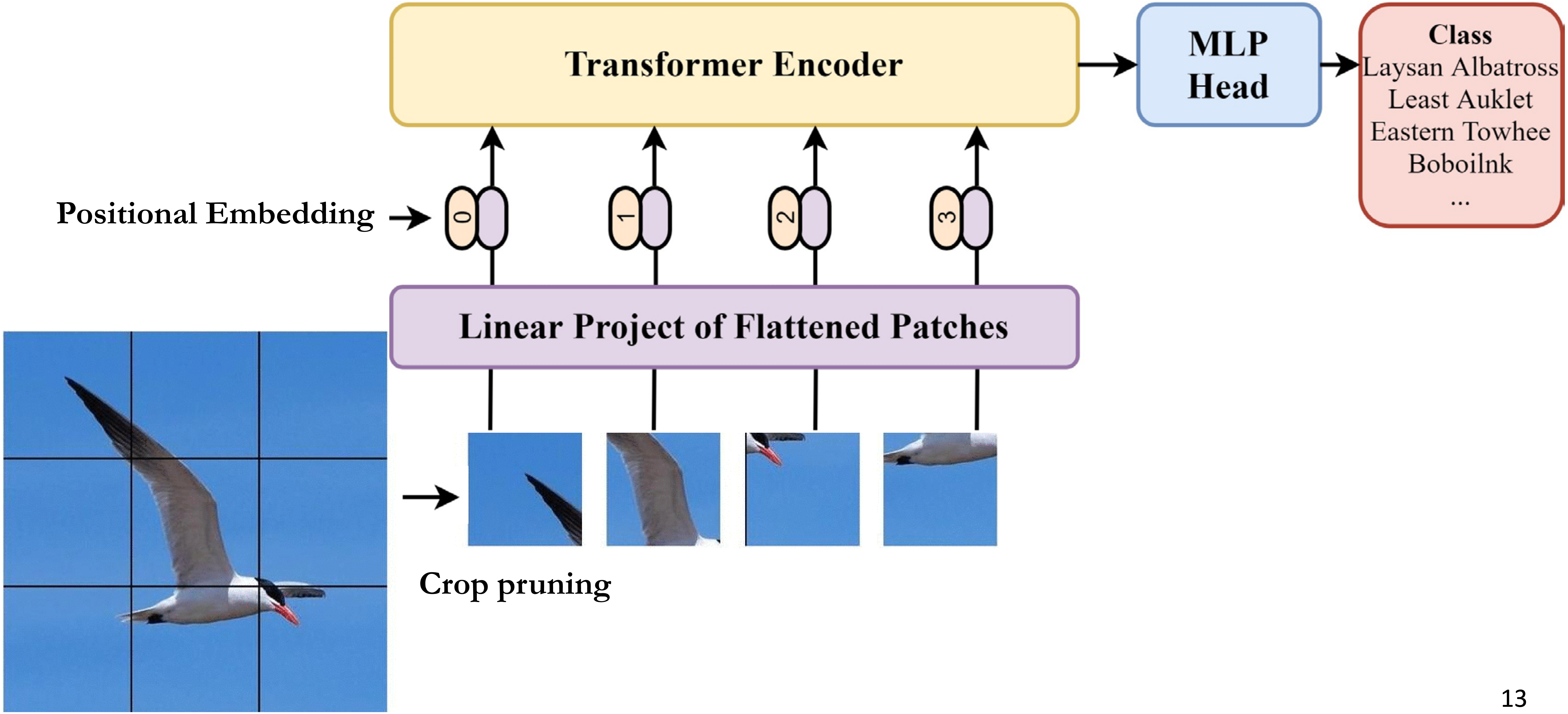
Varying Resolution of Transmitted Image/Feature



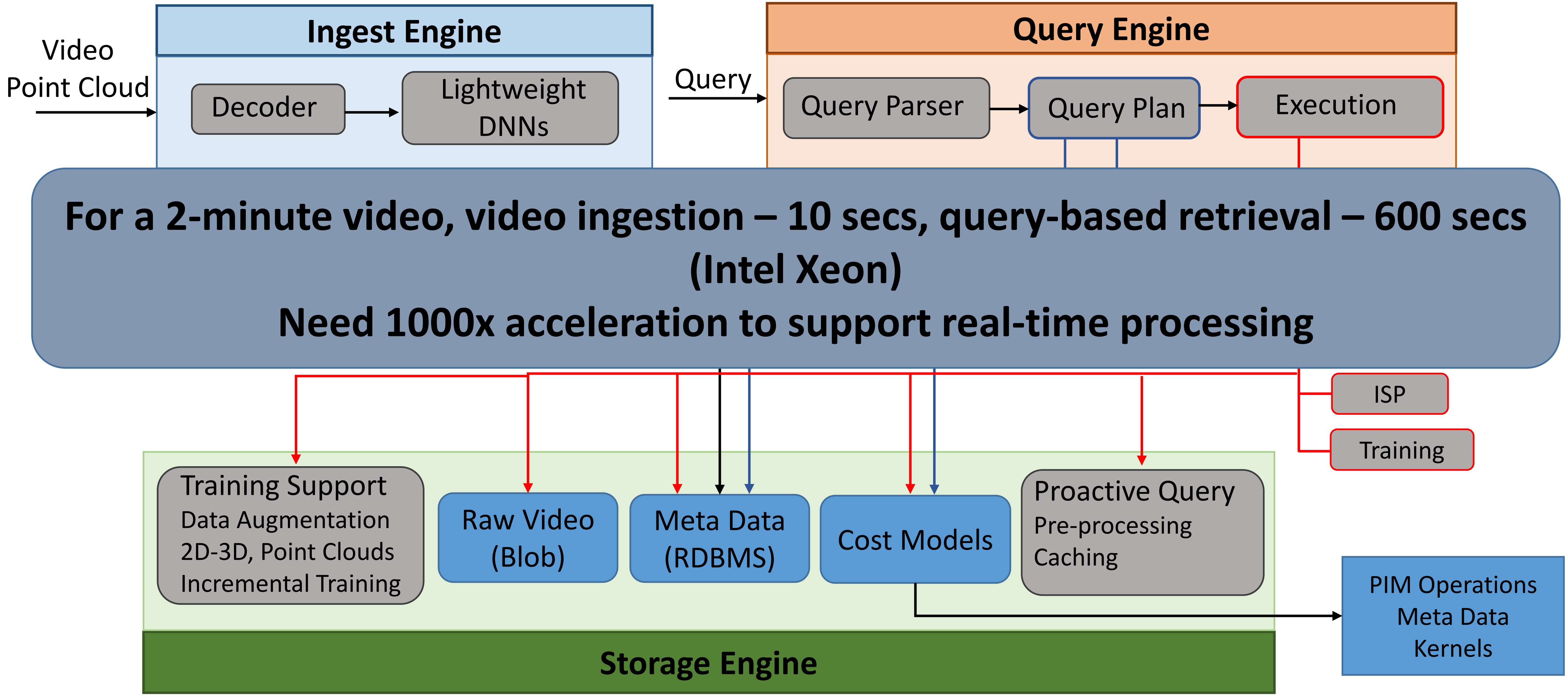


5X reduction in communication energy for ~2% loss in accuracy

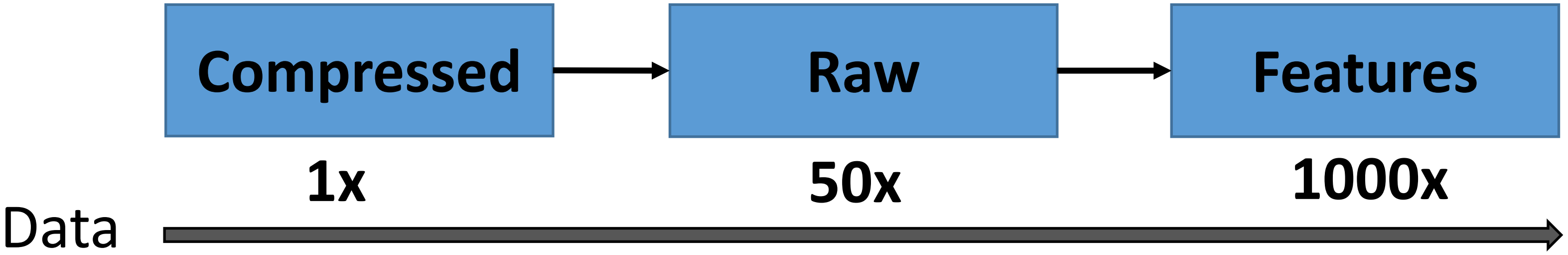
Selective Transmission of High Resolution Images



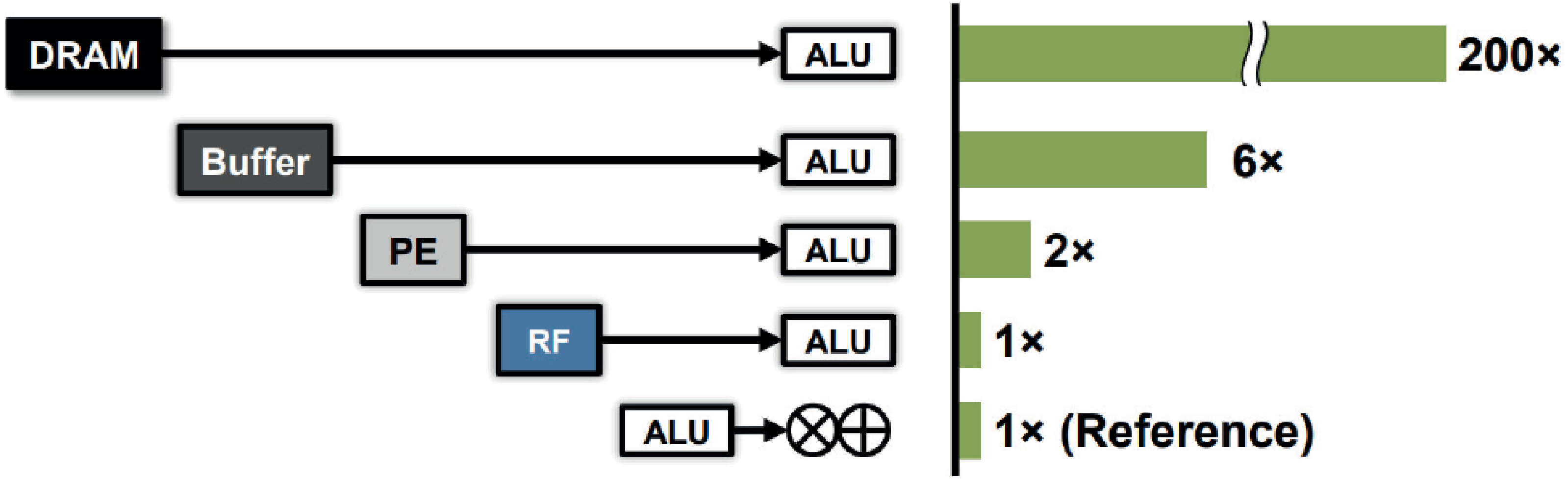
Computing Demands - Visual Analytics System



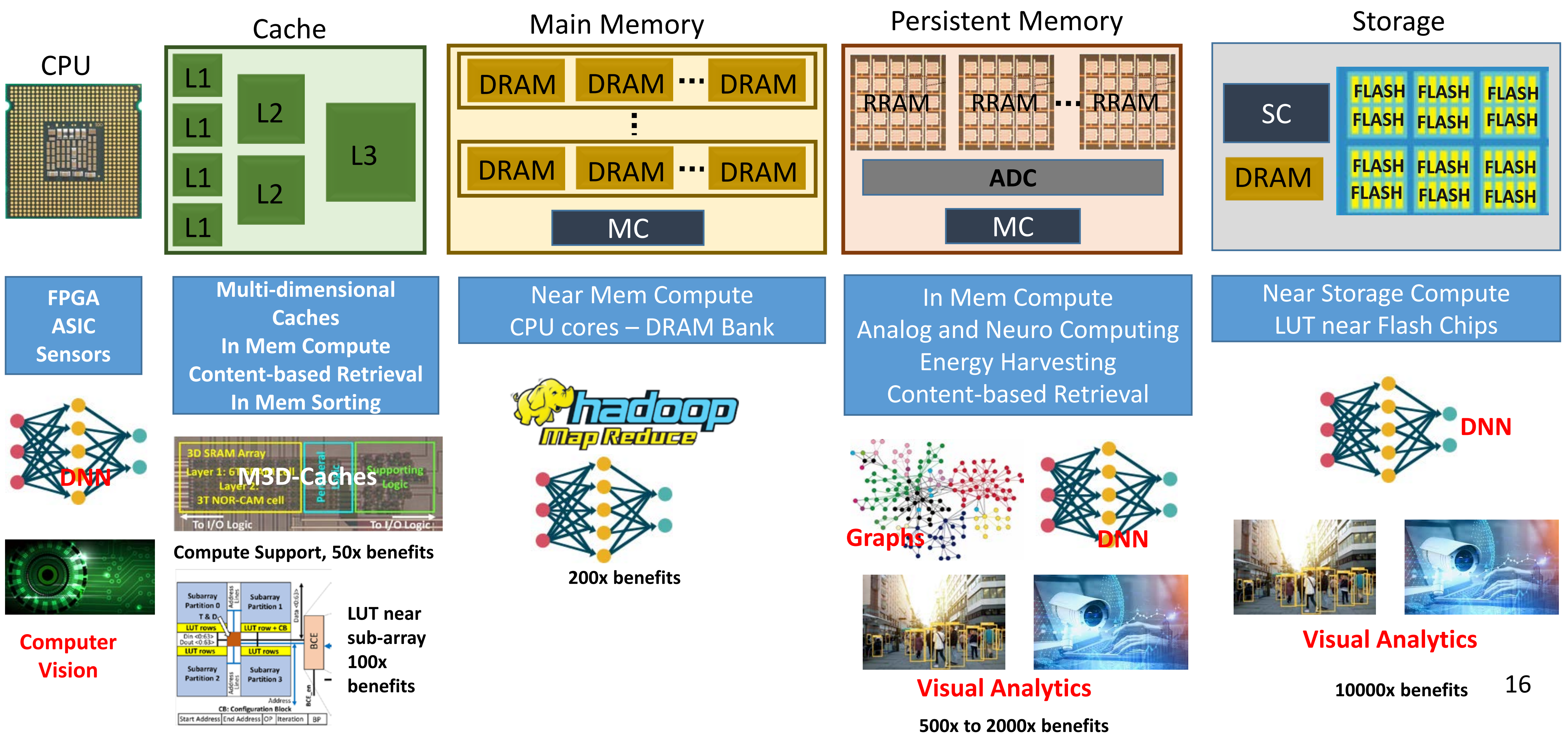
Latency-Storage Tradeoff



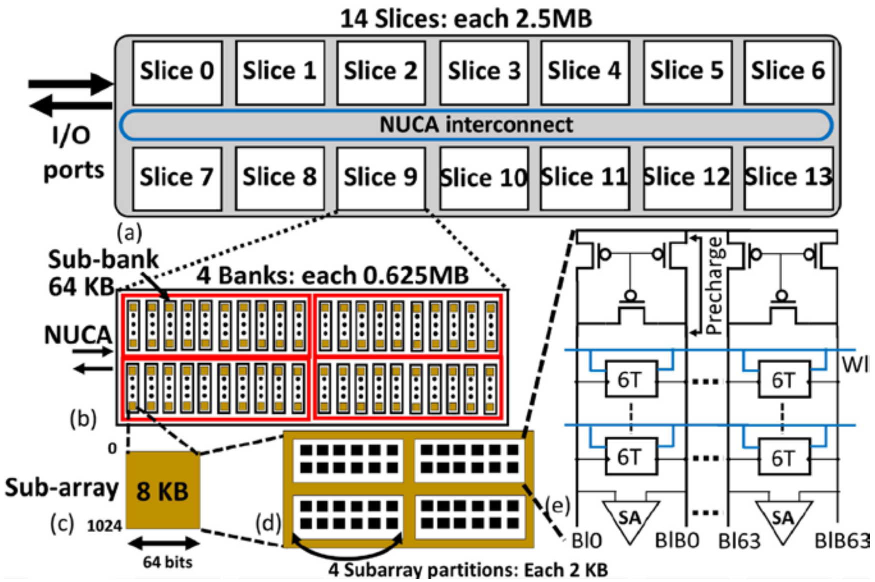
Efficient algorithms, Hardware Acceleration, High density memory/storage, Compute near memory/storage, 3D Integration

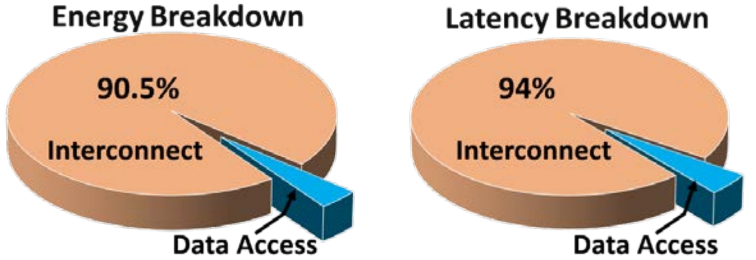


Compute-Memory-Storage Hierarchy



SRAM Based Accelerators

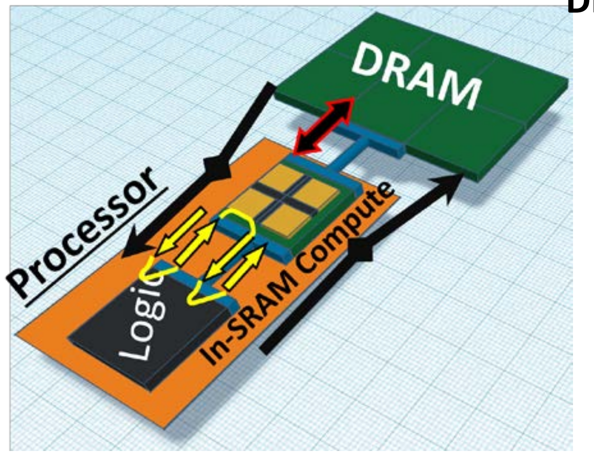




Cost of sub-array access
Cost of interconnect

The data movement cost is majorly spent on transporting the data from sub-array port to the Bank's port.

Computing in-SRAM

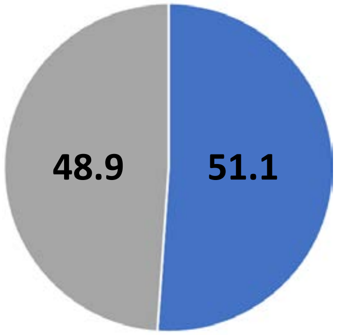


Computing within-SRAM alleviates the data movement costs between the logic and cache. But, suffers from data movement between DRAM and SRAM.

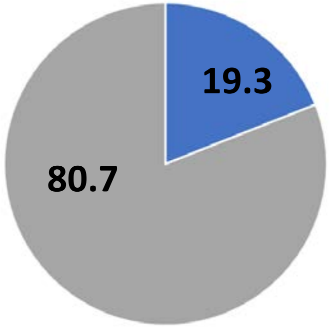
This stresses the need for computation in next level memory, i.e., across different memory hierarchy.

Loading from DRAM
In-SRAM Computation

Latency split-up executing Inception-V3



Energy split-up



Processing in Memory Approaches

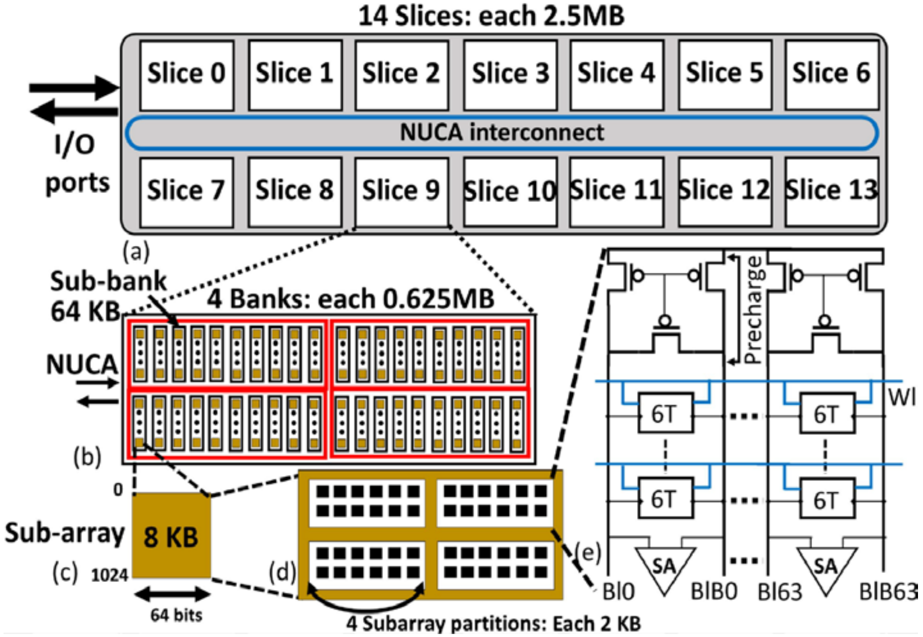
- ✘ Analog based PIM: Requires costly ADCs, and very prone to PVT variations.
- ✘ Digital based PIM: Requires changes to the tightly built custom-layout sub-arrays.
  Repeated bitline (dis)charging used for the compute.

Goal: Place compute logic near each sub-array without any perturbation to the sub-array.

- ✘ Multiplier based logics are area expensive and energy consuming.
- ✓ Look-up table-based compute engines requires lesser area and are more energy efficient.

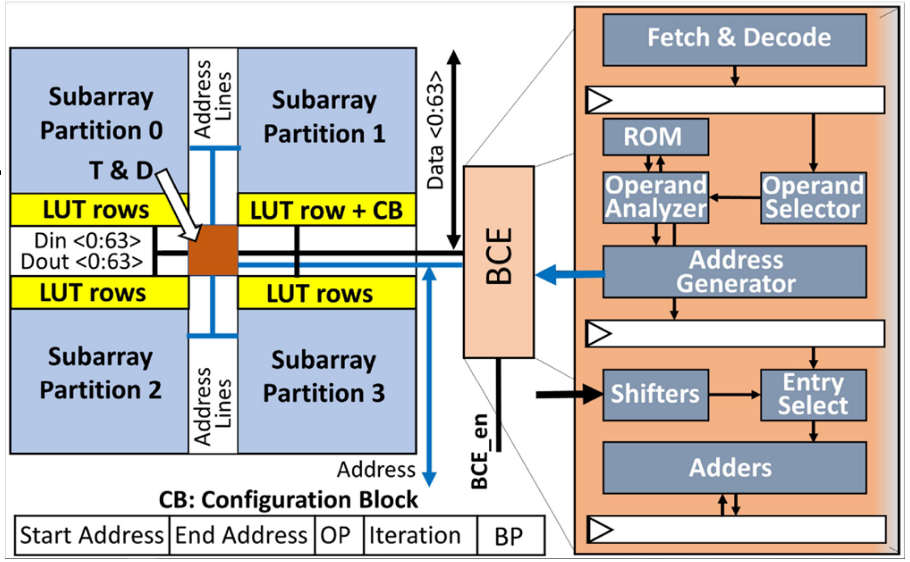
Look-Up Table based Energy Efficient Processing in Cache Support for Neural Network Acceleration

SRAM Organization



With LUT-based compute engine

Bitline free (BFree) Compute engine (BCE) is attached to each Sub-array



Current in-memory solutions requires frequent accesses to the highly parasitic bitlines which incurs high energy penalty. Our solution using reduced access rows within the sub-array in conjunction with compute engine eliminates the energy costs.

Collaboration with Intel Labs

LUT Functions: Multiplication

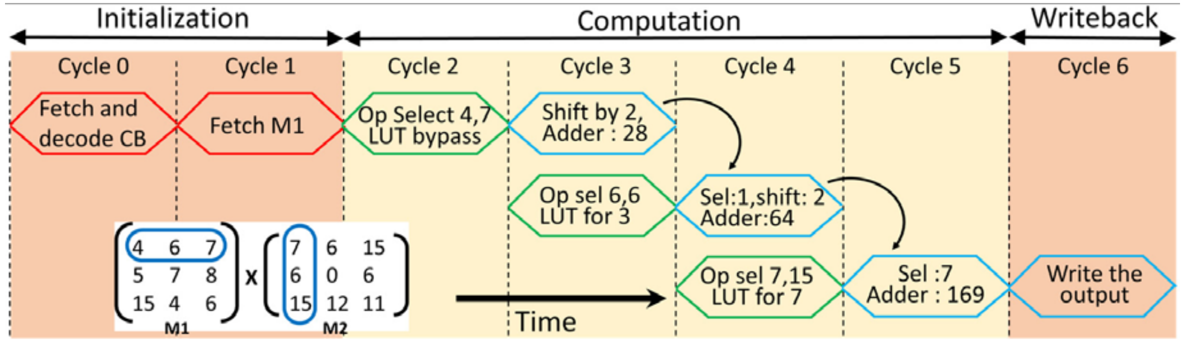
Operand	3_d	5_d	7_d	9_d	11_d	13_d	15_d
3_d	9_d	15_d	21_d	27_d	33_d	39_d	45_d
5_d	15_d	25_d	35_d	45_d	55_d	65_d	75_d
7_d	21_d	35_d	49_d	63_d	77_d	91_d	105_d
9_d	27_d	45_d	63_d	81_d	99_d	117_d	135_d
11_d	33_d	55_d	77_d	99_d	121_d	143_d	165_d
13_d	39_d	65_d	91_d	117_d	143_d	169_d	195_d
15_d	45_d	75_d	105_d	135_d	165_d	195_d	225_d

7_d	21_d	35_d	49_d	63_d	77_d	91_d	105_d

105_d

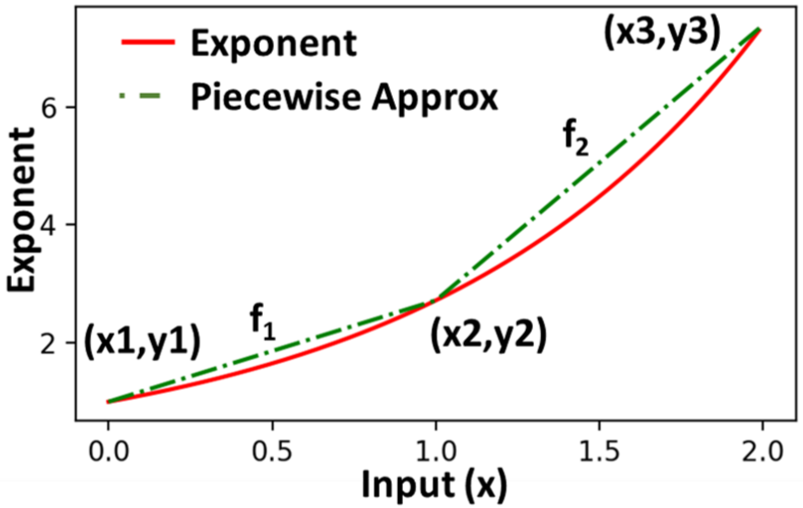
1 Use operand A to generate the address

2 Operand B as select signal



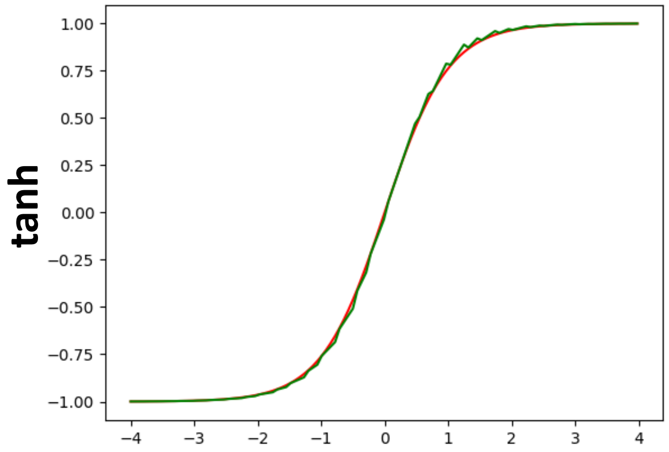
Naïve multiply LUT requires 256B of entries for 4-bit operands. With simple data shifting optimizations[6], the even number operands can be computed, thereby reducing the LUT size to 49Bytes.

LUT Functions: Activation Functions



Piecewise LUT

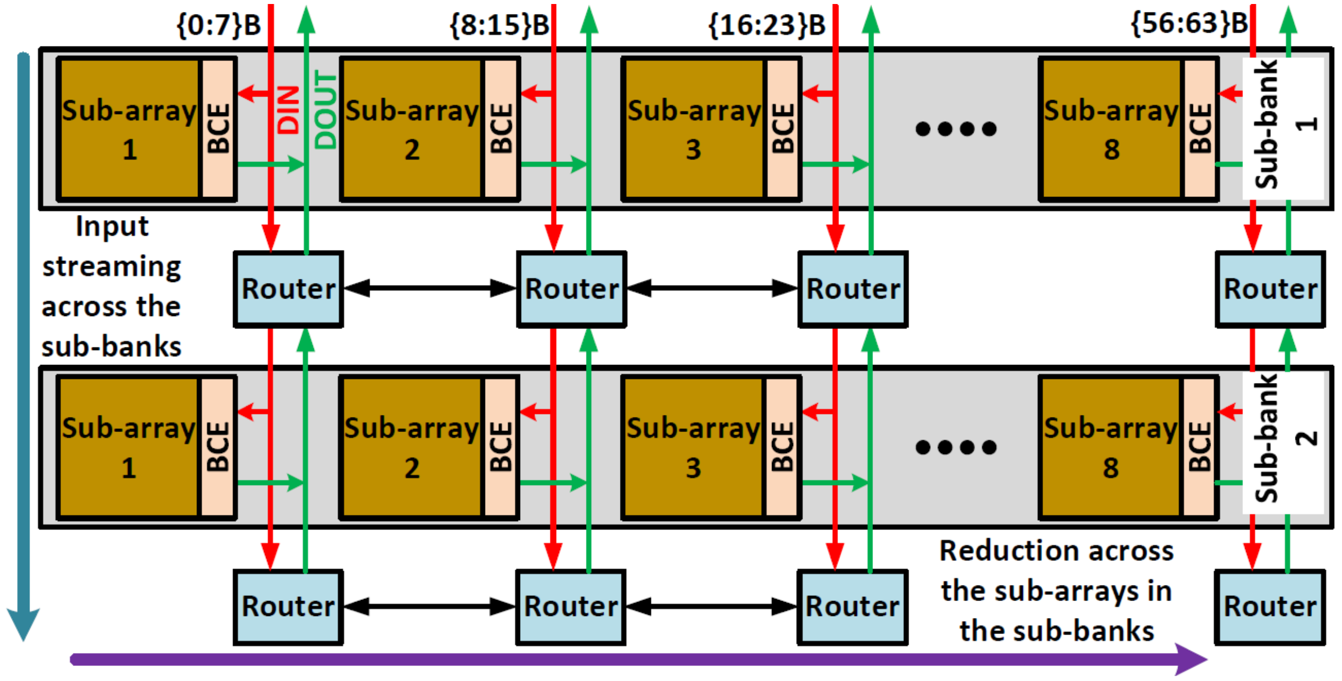
Function	Input Range	α^s	k
f_1	x1,x2	α^1	k^1
f_2	x2,x3	α^2	k^2



The activation functions like exponent, tanh, sigmoid are supported with the piecewise approximation method[7].

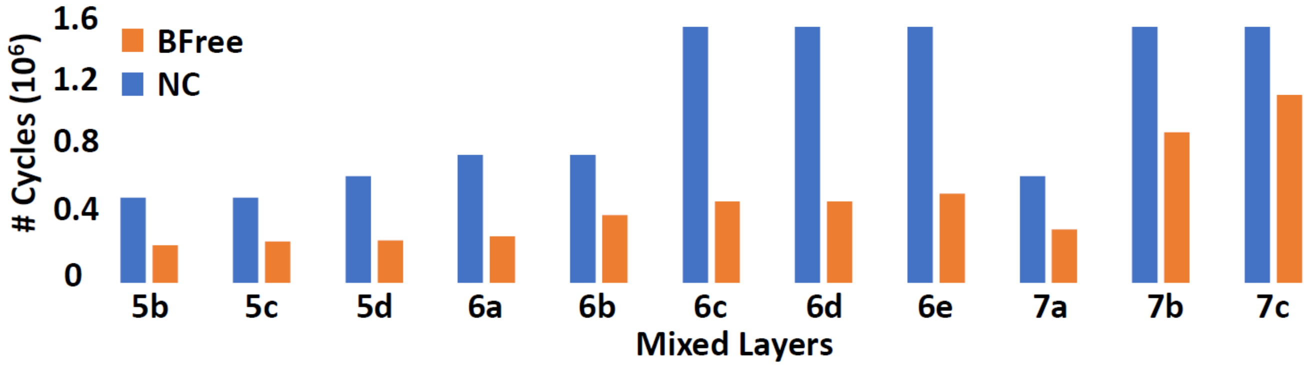
LUT size: 34 entries for 2-bit fractional part

Systolic Dataflow within the Banks



To enable systolic dataflow within the banks, simple switch routers are sandwiched between the sub-arrays. The control signals to these routers are controlled by the BCEs.

Performance Evaluation



Main benefits of BFree over state-of-the-art Neural Cache for Inception V3:

- Minimal perturbation to the sub-array, thereby running at higher frequency.
- Less data movement overheads due to systolic flow.

Our Bitline-Free architecture performs 1.72x faster and 3.14x energy efficient than the state-of-the-art Bitline based computing – Neural Cache while running Inception-V3.

LSTM

Matrix-vector multiplication, tanh and sigmoid

BFree performs 2065x, 224x faster and 3100x, 443x energy efficient than CPU and GPU, respectively.

Transformer Network

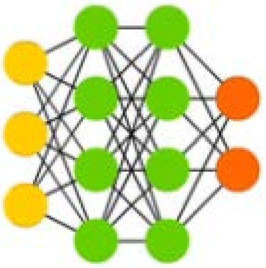
Matrix-matrix multiplication, matrix addition, normalisation, tanh, sigmoid, softmax.

BFree shows 101x, 3x speed up and 91x, 11x energy efficiency than CPU and GPU, respectively for BERT-Base model.

Visual Analytics – Compressed Domain Processing

Skeleton-based Human Action Recognition

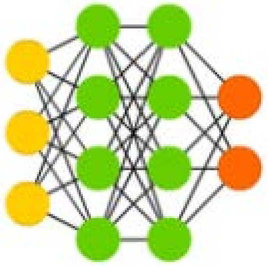
Compressed Video → Decode →



300 secs

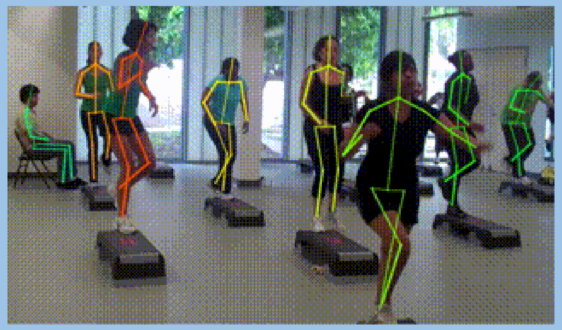
CNNs

Compressed Video → I-Frame / P-Frame / B-Frame / MVs →

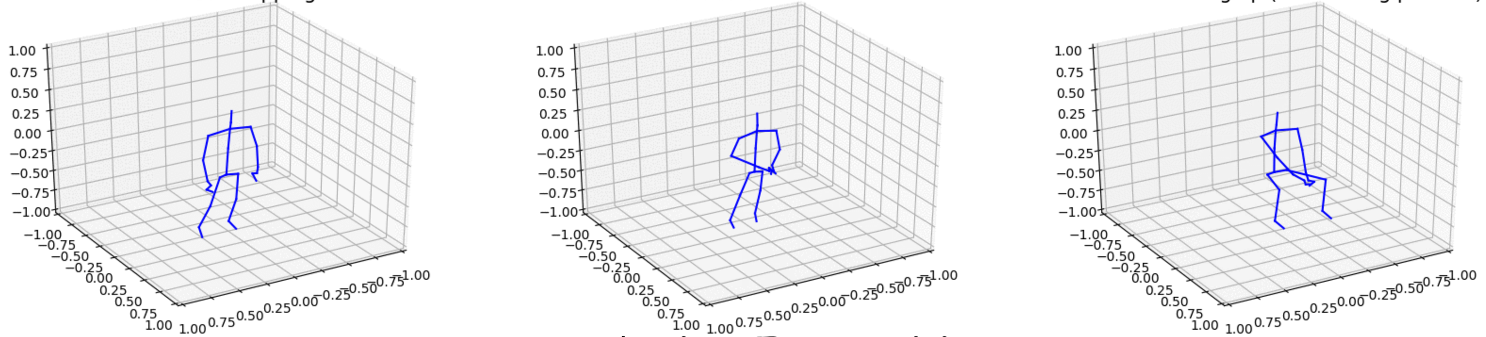


CNNs

50 secs

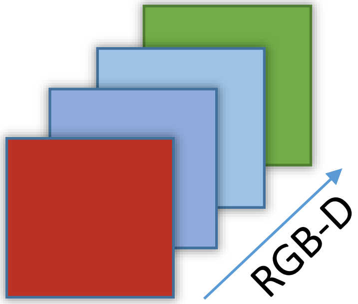


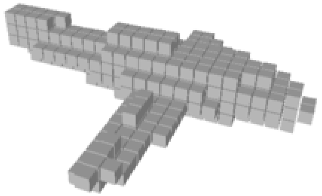
Pose Estimation
(CNN on compressed domain)



Action Recognition
(Compact Transformer)

Understanding the 3D World from 2D





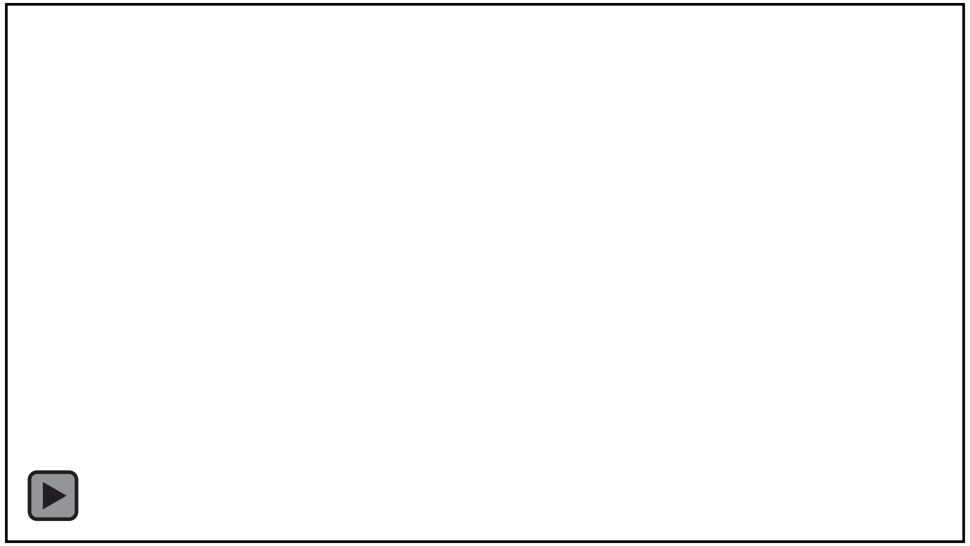
3D-R2N2

PointSetGen

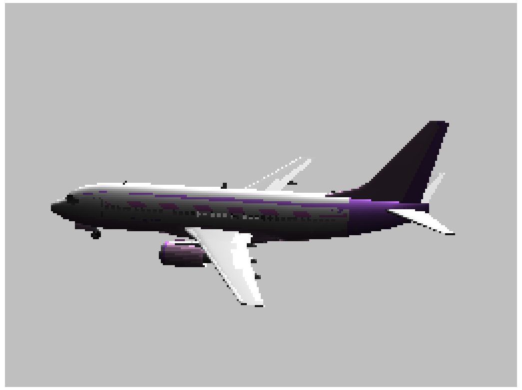
AtlasNet

- ❖ Understanding the 3D world from monocular vision has always been an area of great interest.
- ❖ Standard RGB 3 channel images do not possess the depth of field information
- ❖ RGB data in presence of adequate depth information can generate accurate 3D models

1. Choy, Christopher B., et al. "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction." *European conference on computer vision*. Springer, Cham, 2016.
2. Fan, Haoqiang, et al. "A point set generation network for 3d object reconstruction from a single image." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
3. Groueix, Thibault, et al. "A papier-mâché approach to learning 3d surface generation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

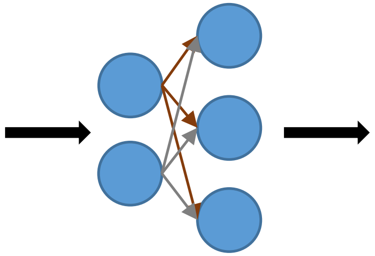


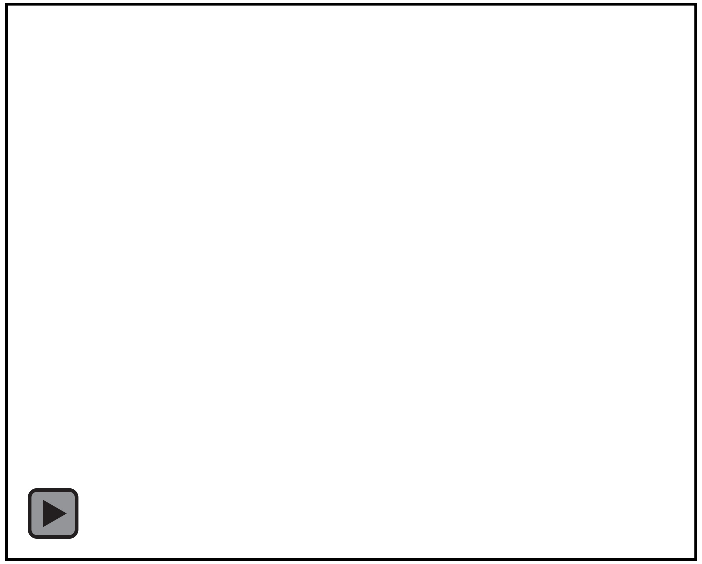
Point Cloud Generation from RGB Image and Dense Depth



RGB Image

Depth

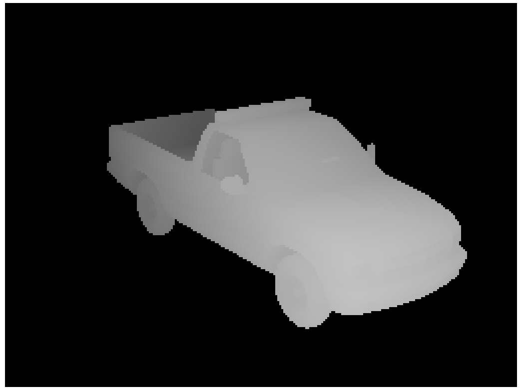




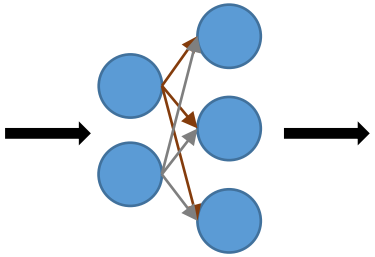
3D Pointcloud

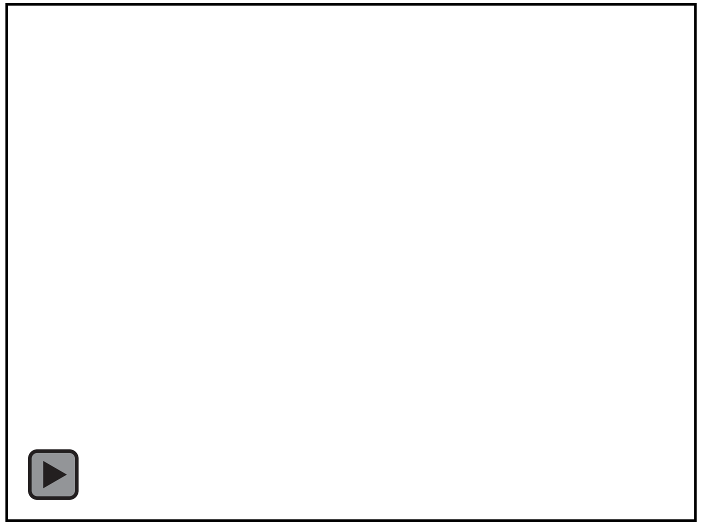
Artificial Neural Nets

RGB Image

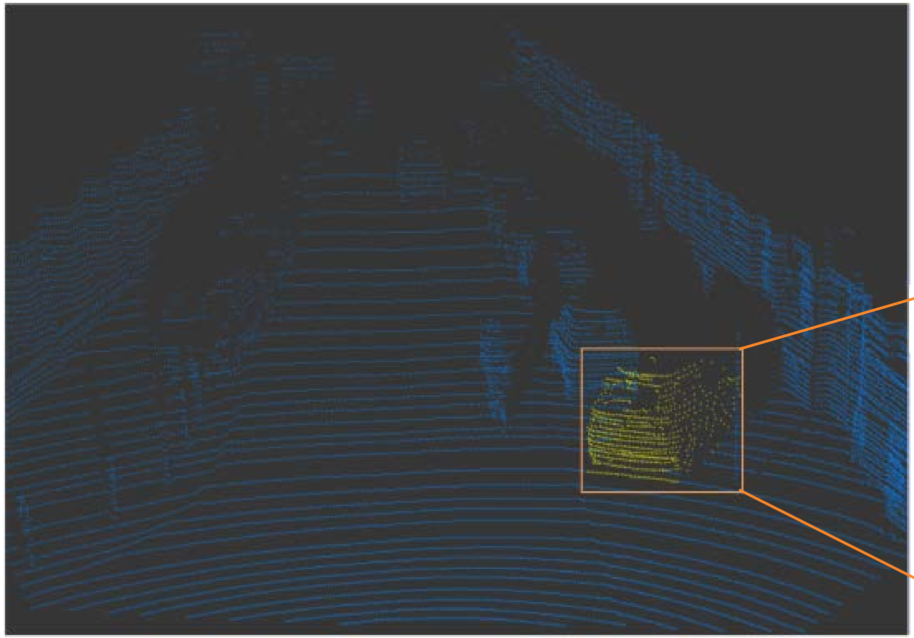


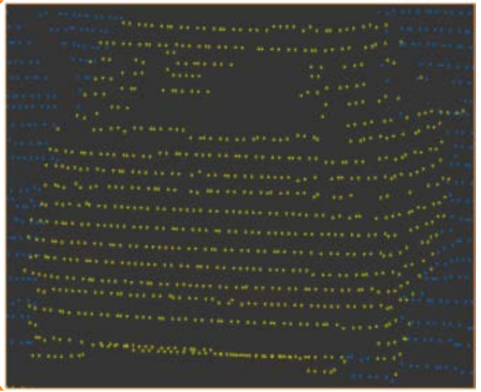
Depth





Existing Depth Sensors Provide Sparse Depth Data



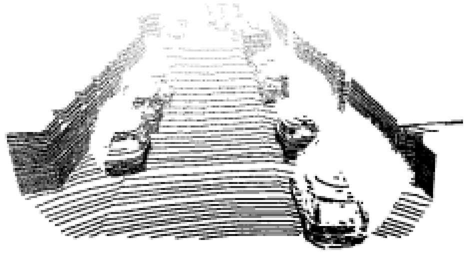


Sparse Depth Map in the Night

Depth Sensors

1. LIDAR
2. Time of Flight
3. RGBD Camera

RGB Image



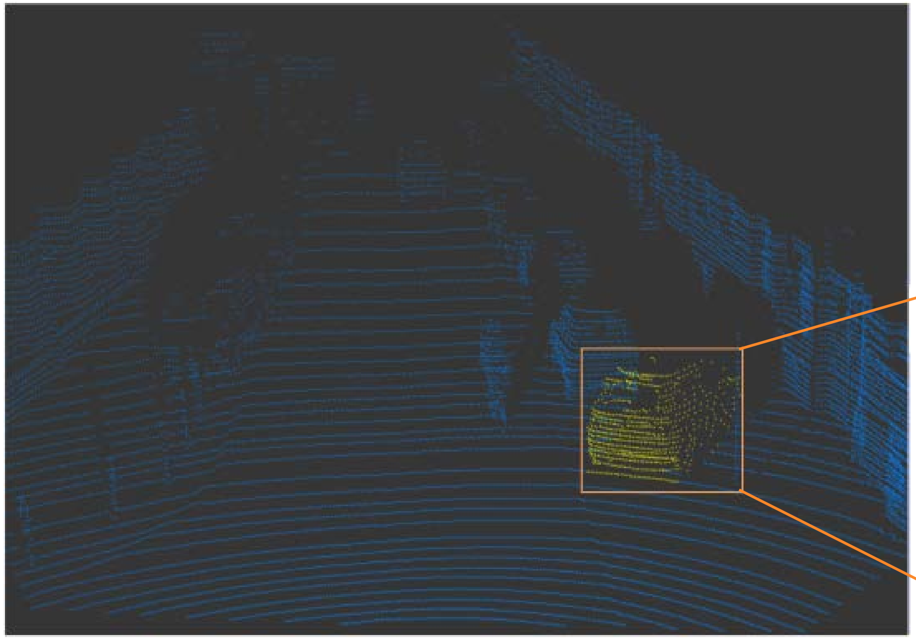
Sparse Depth Map

Problem:

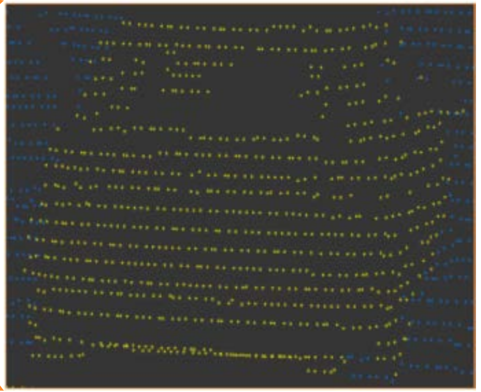
1. These sensors provides sparse depth data both temporally and spatially
2. The LIDAR sensor provides the 3D spatial information at a low frequency ~ 20Hz [2]
3. Moreover, the obtained depth information is sparse e.g., 64 vertical lines in the vertical direction [1]

1. Liu, Haojie, et al. "Pseudo-LiDAR Point Cloud Interpolation Based on 3D Motion Representation and Spatial Supervision." *arXiv preprint arXiv:2006.11481* (2020).
2. Tang, Jie, et al. "Learning guided convolutional network for depth completion." *IEEE Transactions on Image Processing* 30 (2020): 1116-1129.

Existing Depth Sensors Provide Sparse Depth Data



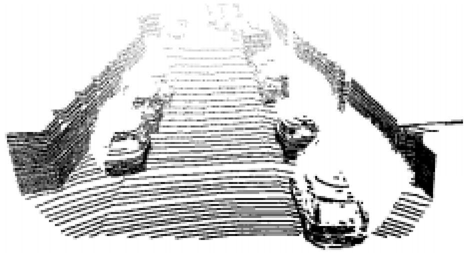
Sparse Depth Map in the Night



Depth Sensors

1. LIDAR
2. Time of Flight
3. RGBD Camera

RGB Image



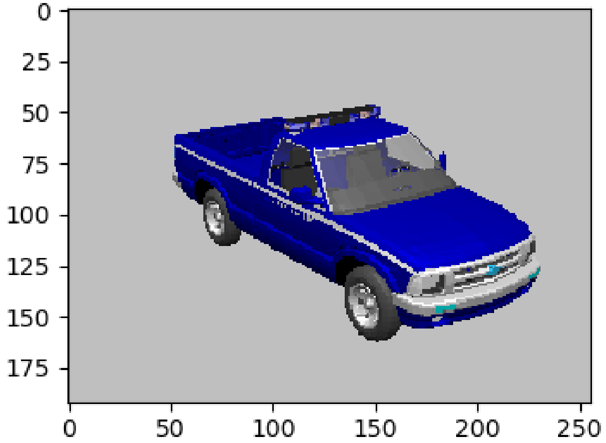
Sparse Depth Map

Problem:

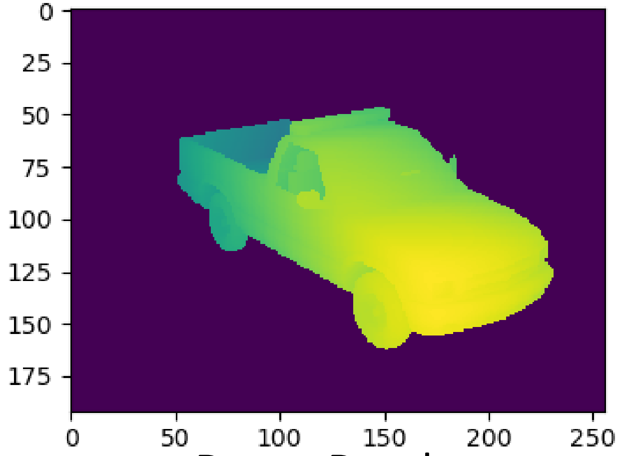
1. These sensors provides sparse depth data both temporally and spatially
2. The LIDAR sensor provides the 3D spatial information at a low frequency ~ 20Hz [2]
3. Moreover, the obtained depth information is sparse e.g., 64 vertical lines in the vertical direction [1]

1. Liu, Haojie, et al. "Pseudo-LiDAR Point Cloud Interpolation Based on 3D Motion Representation and Spatial Supervision." *arXiv preprint arXiv:2006.11481* (2020).
2. Tang, Jie, et al. "Learning guided convolutional network for depth completion." *IEEE Transactions on Image Processing* 30 (2020): 1116-1129.

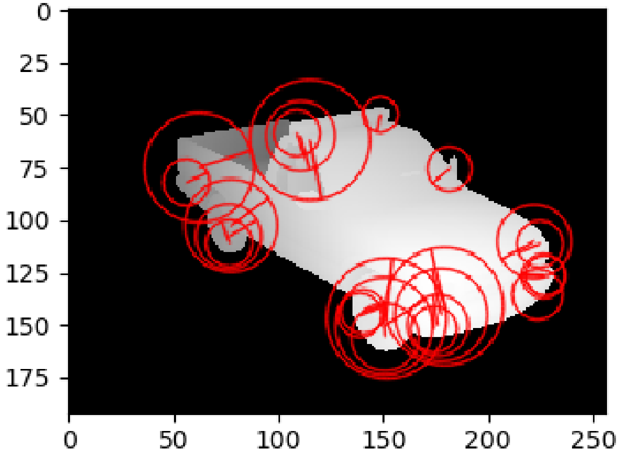
Feature Guided Directed Sampling improves Dense Depth Prediction



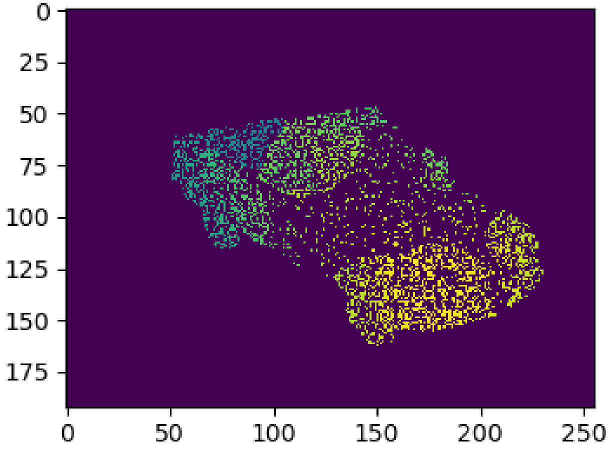
RGB Image



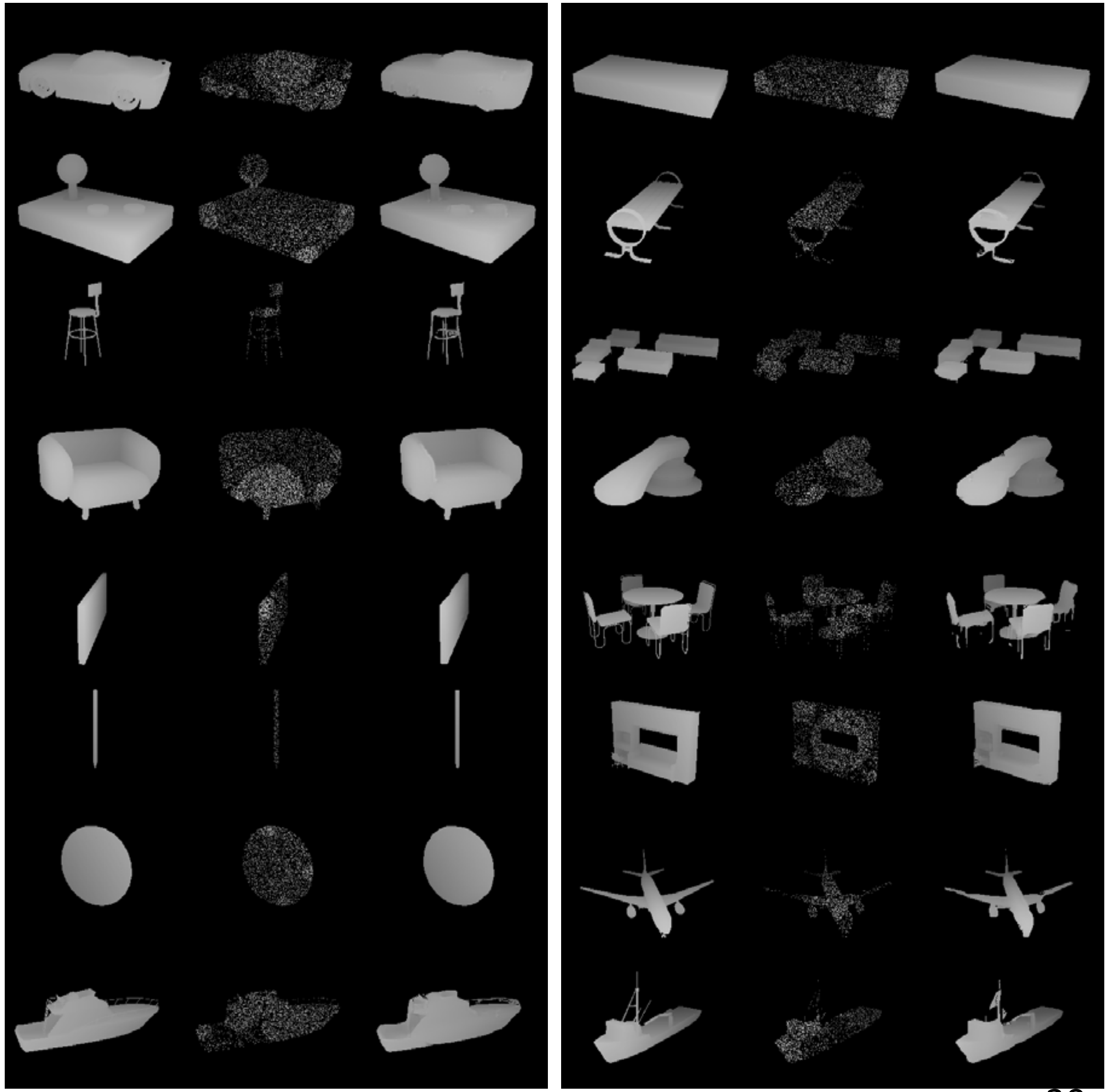
Dense Depth



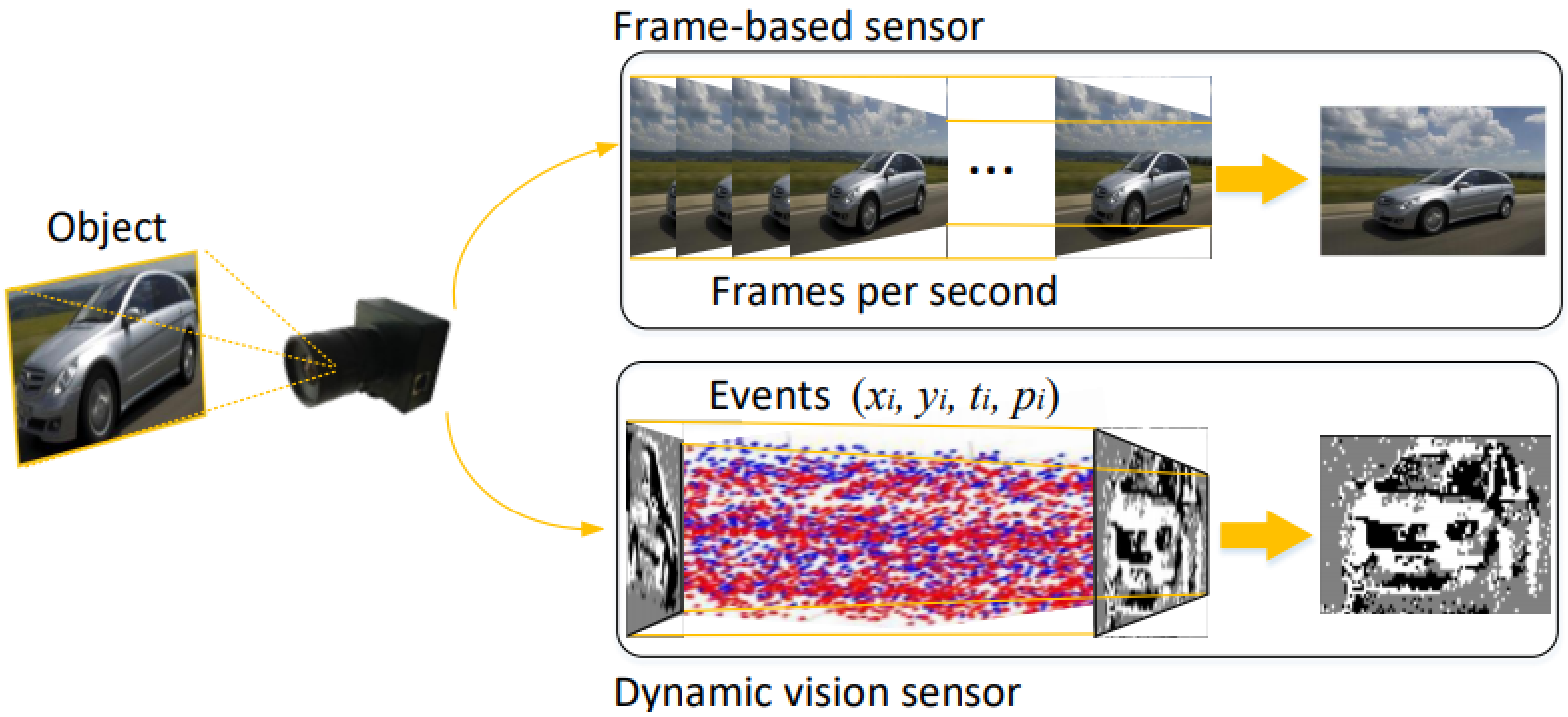
Extracted SURF Features

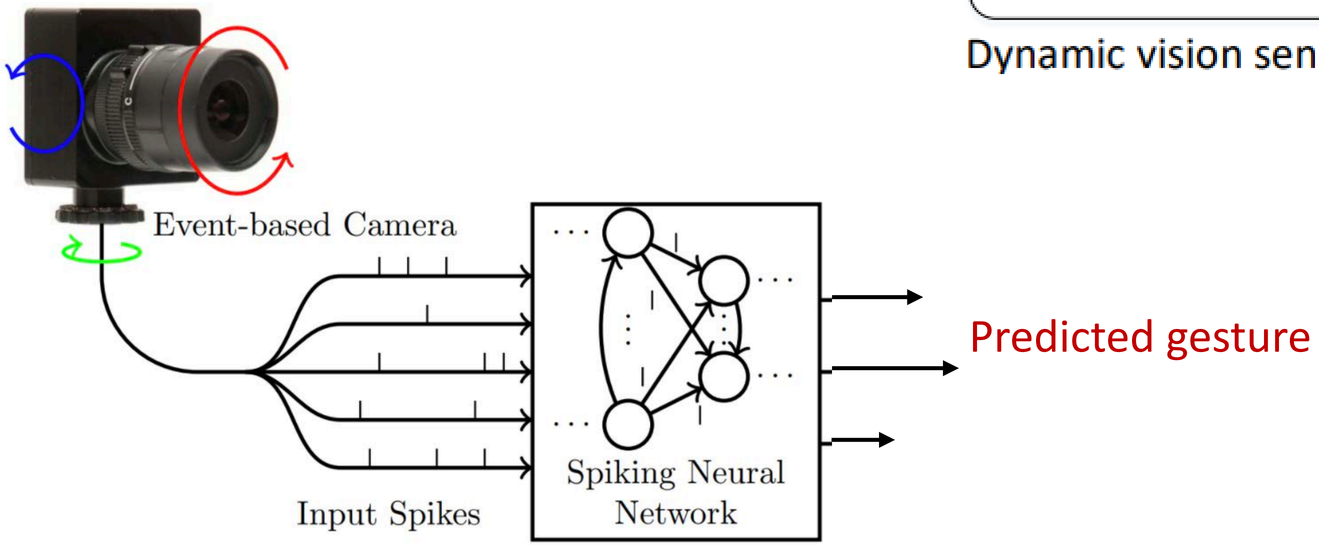


Sampled Sparse Depth

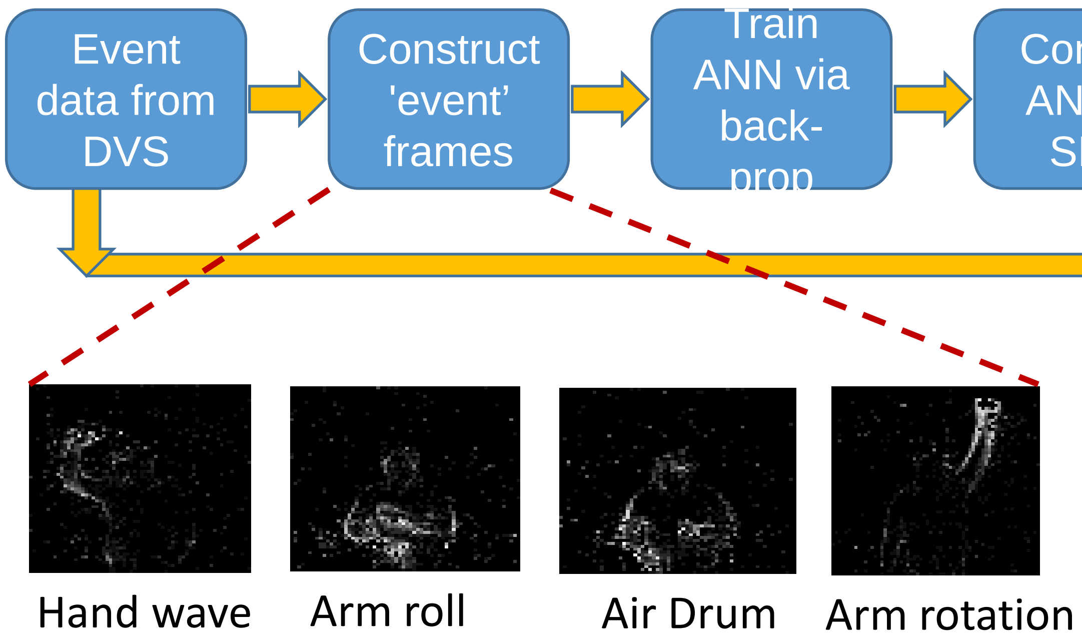

Examples of Recovered Depth

Event-based Sensors and Data

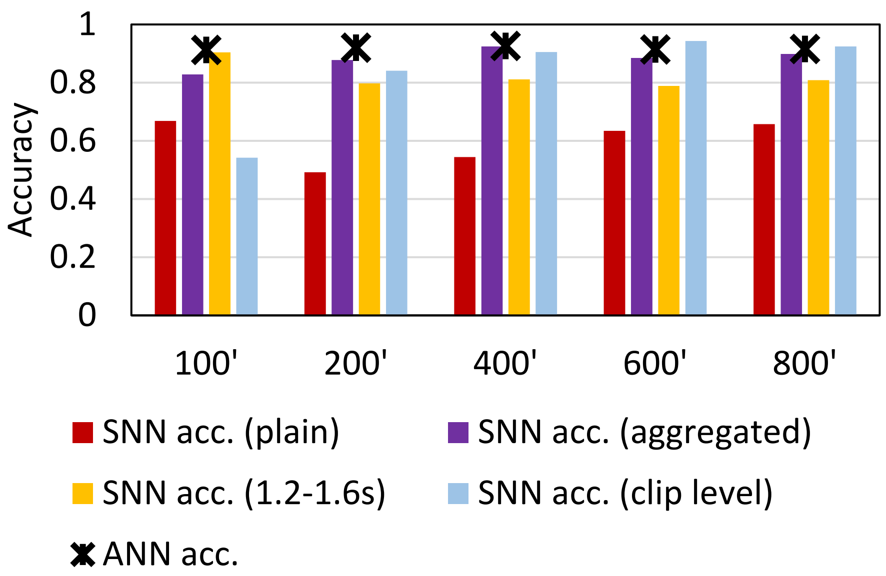




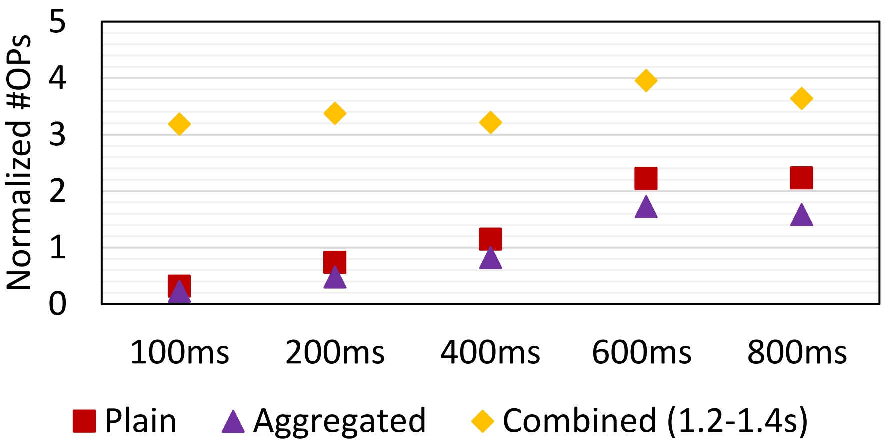
HAR using Event Data + SNNs



SNN acc. vs frame duration (ms)



#OPs (SNN/ANN) vs dataset



- Normal ANN-SNN conversion is noisy, with non-uniform spike rate → accuracy losses
- Near lossless conversion can be achieved by stream-lining the spikes.
- We propose a delayed firing strategy to achieve better accuracy with fewer Ops (denoted in purple) in both figs.

Thank you