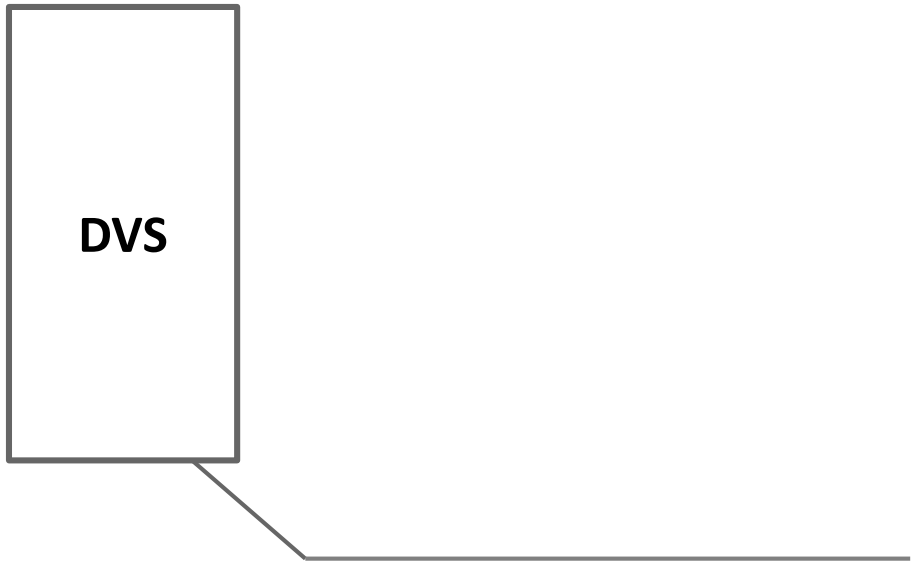
ETH zürich

Integrating Event-based Dynamic Vision Sensors with Sparse Hyperdimensional Computing: A Low-power Accelerator with Online Learning Capability

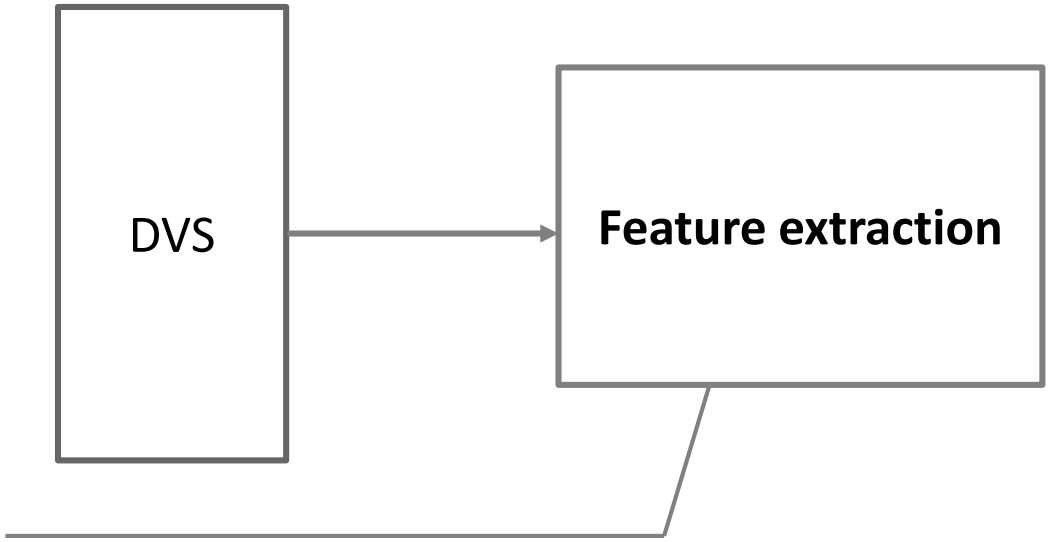
Michael Hersche, **Edoardo Mello Rella**, Alfio Di Mauro, Luca Benini and Abbas Rahimi
ETH Zurich, Dept. EE & IT, Integrated Systems Laboratory (IIS), Switzerland
07/16/2020

Extract motion information from Dynamic Vision Sensors sequences for autonomous driving



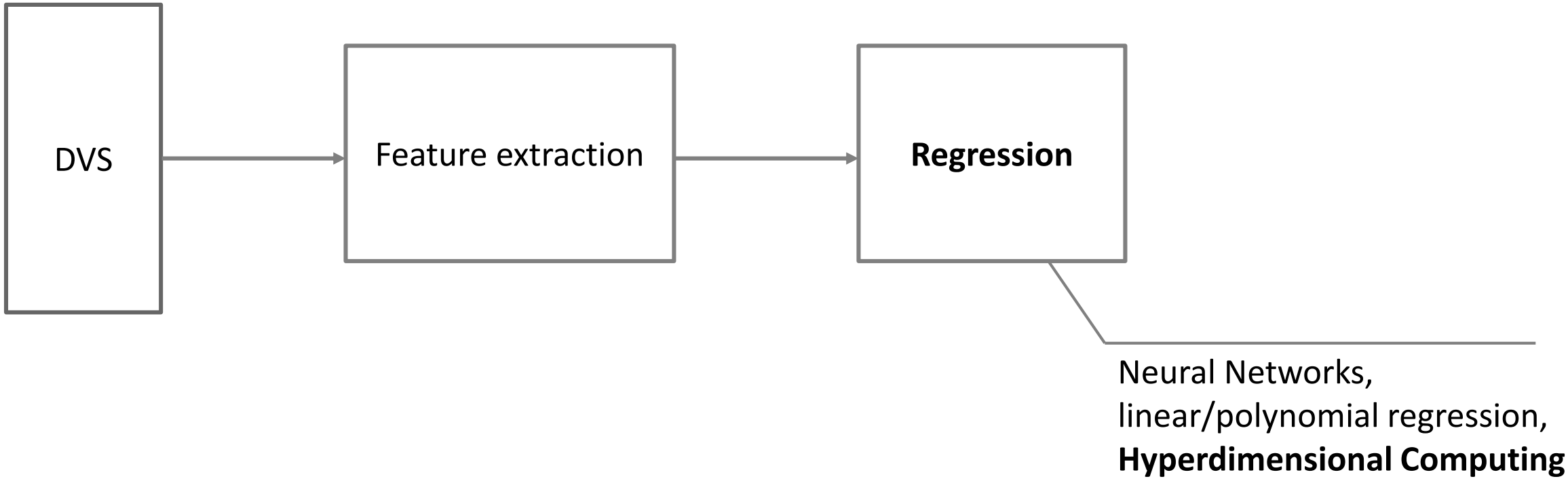
Vision sensor:
Records motion in the visual field.
Outputs sequences of asynchronous events (detected motion)

Extract motion information from Dynamic Vision Sensors sequences for autonomous driving

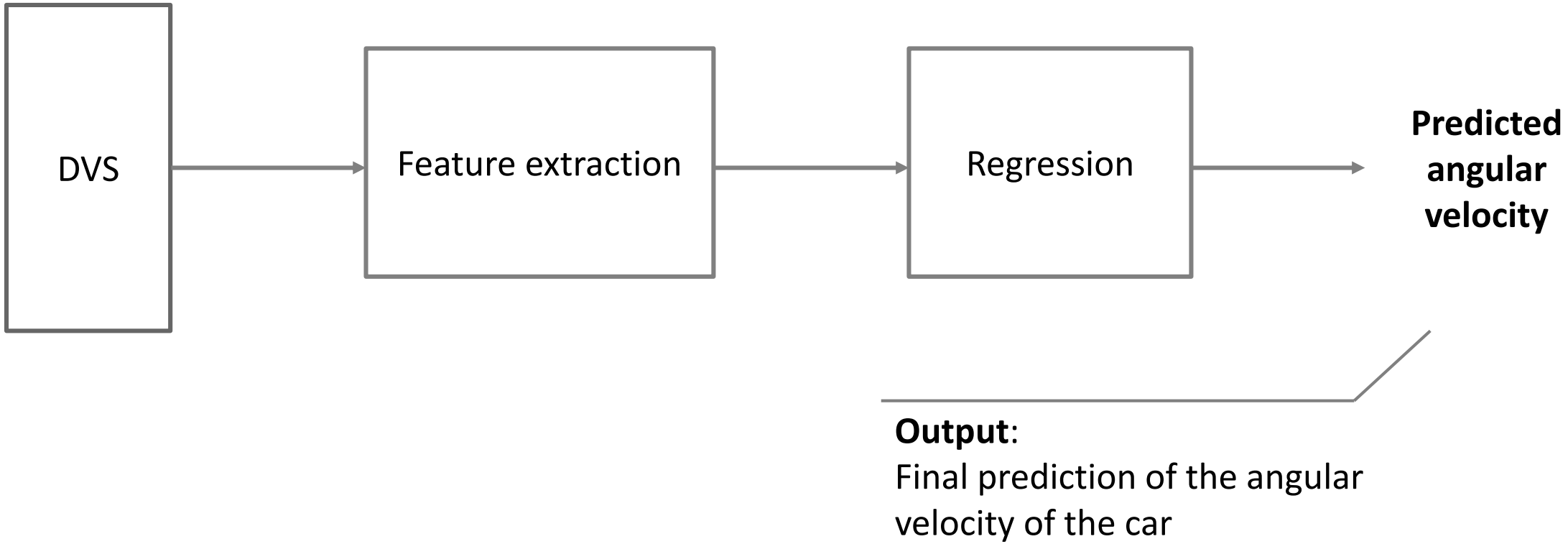


Data preprocessing:
Projects asynchronous events into interpretable features

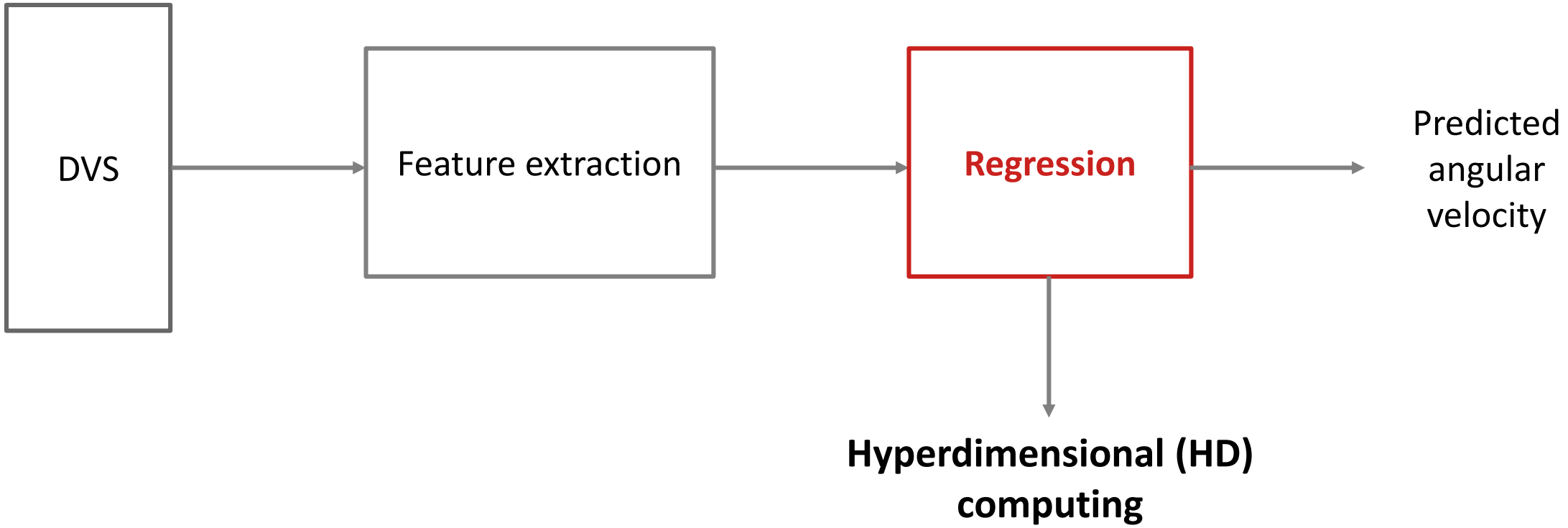
Extract motion information from Dynamic Vision Sensors sequences for autonomous driving



Extract motion information from Dynamic Vision Sensors sequences for autonomous driving



Extract motion information from Dynamic Vision Sensors sequences for autonomous driving



Hyperdimensional (HD) computing with dense vectors

Binary random **vectors** in **high dimensional** space

Hyperdimensional (HD) computing with dense vectors

Binary random **vectors** in **high dimensional** space

Vector distance representative of **similarity of source data**

- Projecting similar data to HD vectors leads to similar vectors
- Hamming distance for computationally simple distance computation
- Two independent vectors **orthogonal with high probability**

Hyperdimensional (HD) computing with dense vectors

Binary random **vectors** in **high dimensional** space

Vector distance representative of **similarity of source data**

- Projecting similar data to HD vectors leads to similar vectors
- Hamming distance for computationally simple distance computation
- Two independent vectors **orthogonal with high probability**

Basic operations

- **Bundling**: combines vectors into one → **majority vote**
- **Binding**: associate vectors to establish correlation → **bitwise XOR**
- **Permutation**: generates an orthogonal vector and is used to add information

Hyperdimensional (HD) computing with dense vectors

Binary random **vectors** in **high dimensional** space

Vector distance representative of **similarity of source data**

- Projecting similar data to HD vectors leads to similar vectors
- Hamming distance for computationally simple distance computation
- Two independent vectors **orthogonal with high probability**

Basic operations

- **Bundling**: combines vectors into one → **majority vote**
- **Binding**: associate vectors to establish correlation → **bitwise XOR**
- **Permutation**: generates an orthogonal vector and is used to add information

Build **memories** to learning with HD vectors

- **Bundling vectors** with corresponding labels to build representative model

Hyperdimensional (HD) computing with sparse vectors

Binary vectors in high dimensional space with **low density of 1s**

Simplify cost of operations

	HD sparse vectors	**HD dense vectors**
Bundling	Bitwise OR	Majority vote
Similarity	Overlap	Hamming distance

Contributions

Apply **HD sparse vectors** for angular velocity estimation

Contributions

Apply **HD sparse vectors** for angular velocity estimation

Novel **Randomized Activation Functions Encoding** (**RAFE**) to project DVS sequences to **HD sparse vectors**

Contributions

Apply **HD sparse vectors** for angular velocity estimation

Novel **Randomized Activation Functions Encoding** (**RAFE**) to project DVS sequences to **HD sparse vectors**

Apply **online retraining** on HD vectors algorithms → Introduce online updates of HD models during operation

Contributions

Apply **HD sparse vectors** for angular velocity estimation

Novel **Randomized Activation Functions Encoding** (**RAFE**) to project DVS sequences to **HD sparse vectors**

Apply **online retraining** on HD vectors algorithms → Introduce online updates of HD models during operation

Compare performance of sparse vectors to dense vectors, MLP, and linear&polynomial regression

Contributions

Apply **HD sparse vectors** for angular velocity estimation

Novel **Randomized Activation Functions Encoding** (**RAFE**) to project DVS sequences to **HD sparse vectors**

Apply **online retraining** on HD vectors algorithms → Introduce online updates of HD models during operation

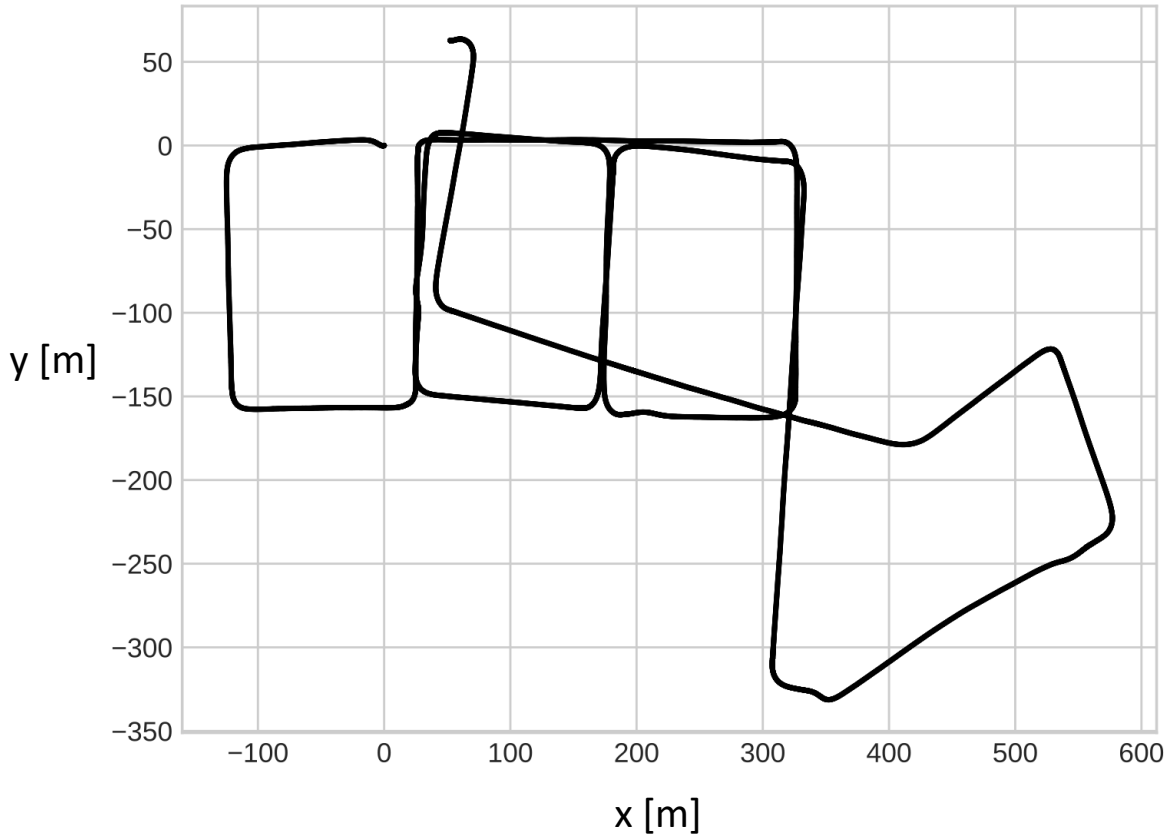
Compare performance of sparse vectors to dense vectors, MLP, and linear&polynomial regression

Measure **power consumption and run-time** of best methods on GAP8 [1], a low power MCU

[1] E. Flamand, et al.2018. GAP-8: A RISC-V SoC for AI at the Edge of the IoT. In *2018 IEEE 29th ASAP. 1–4*

MVSEC Dataset

Sample **camera trajectory**



Five sequences for 30’ recordings on **two days** and **three nights**

Urban sequences

DVS mounted on a **car**

Angular velocity range:
[-35, 35] °/s

Training-testing scenario and error metric ARPE

50% of data as **training** set and **50%** for **testing**

Average relative pose error (**ARPE**)

Cosine similarity of velocity vectors

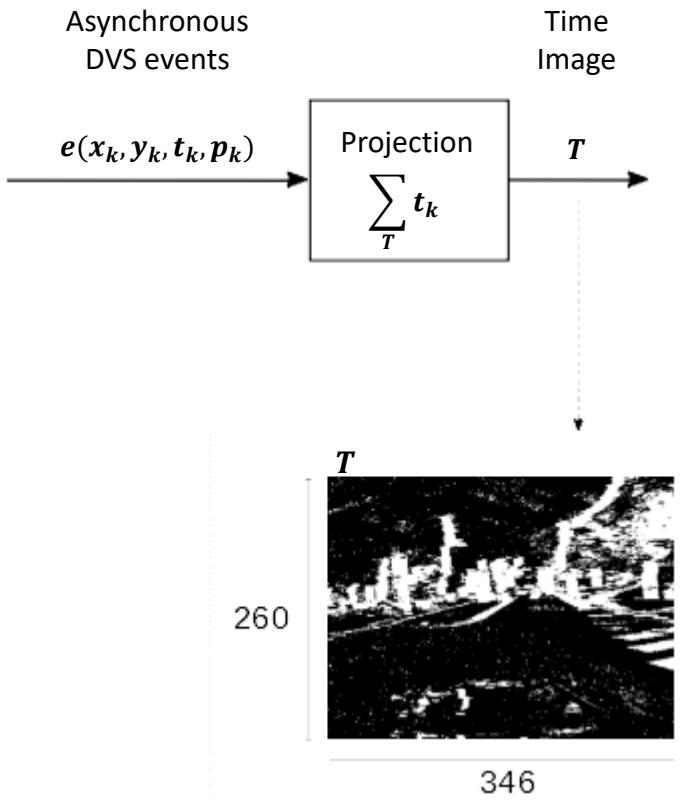
Difference in angular velocity

True velocity

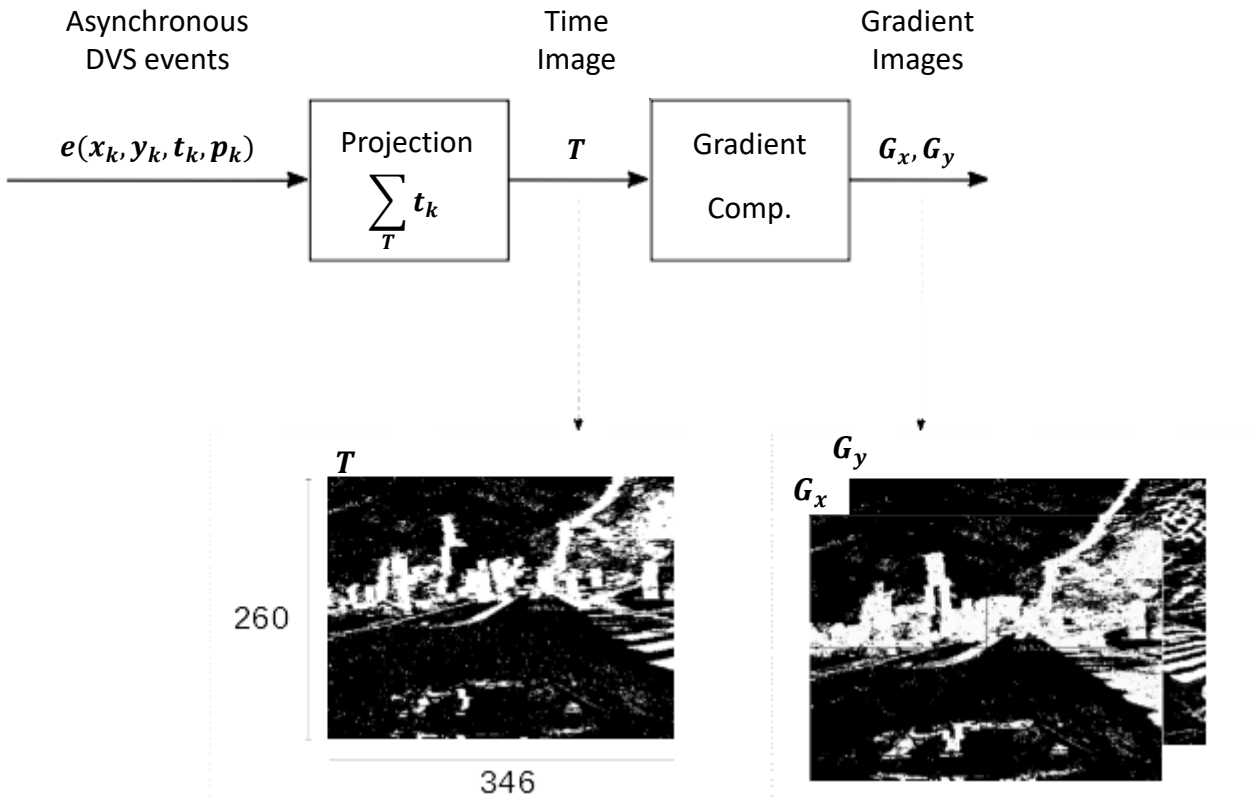
Predicted velocity

$$\text{ARPE} = \frac{1}{n}\sum_{i=1}^{n} \arccos\left(\frac{< \hat{\mathbf{v}}_i, \mathbf{v}_i >}{||\hat{\mathbf{v}}_i||_2 \cdot ||\mathbf{v}_i||_2}\right)$$

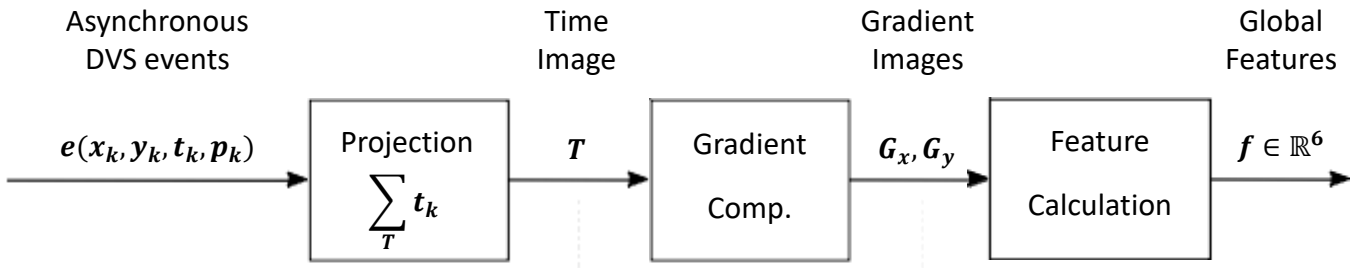
Prediction pipeline

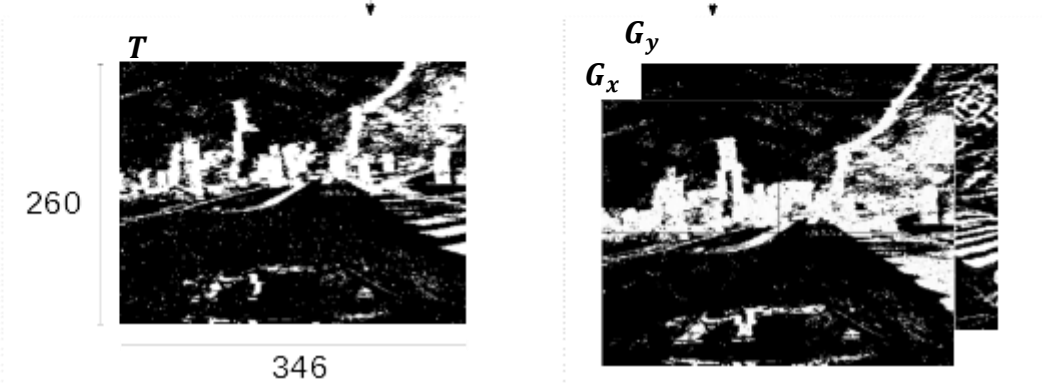


Prediction pipeline

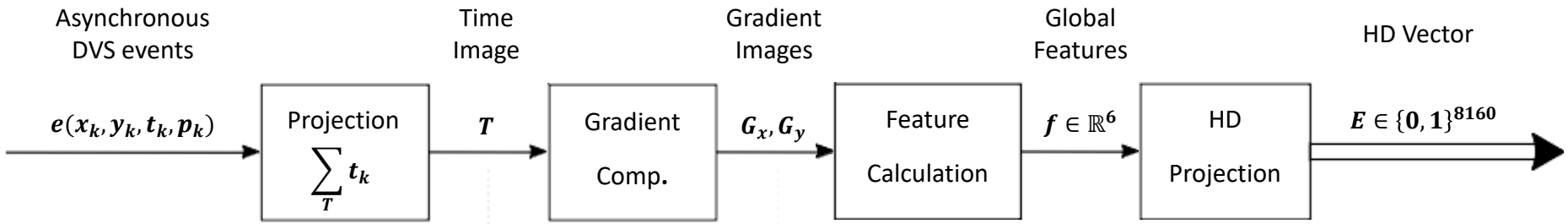


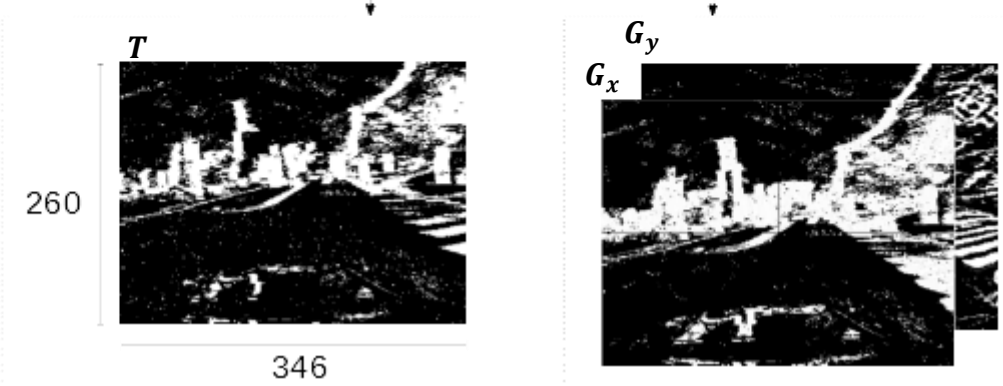
Prediction pipeline



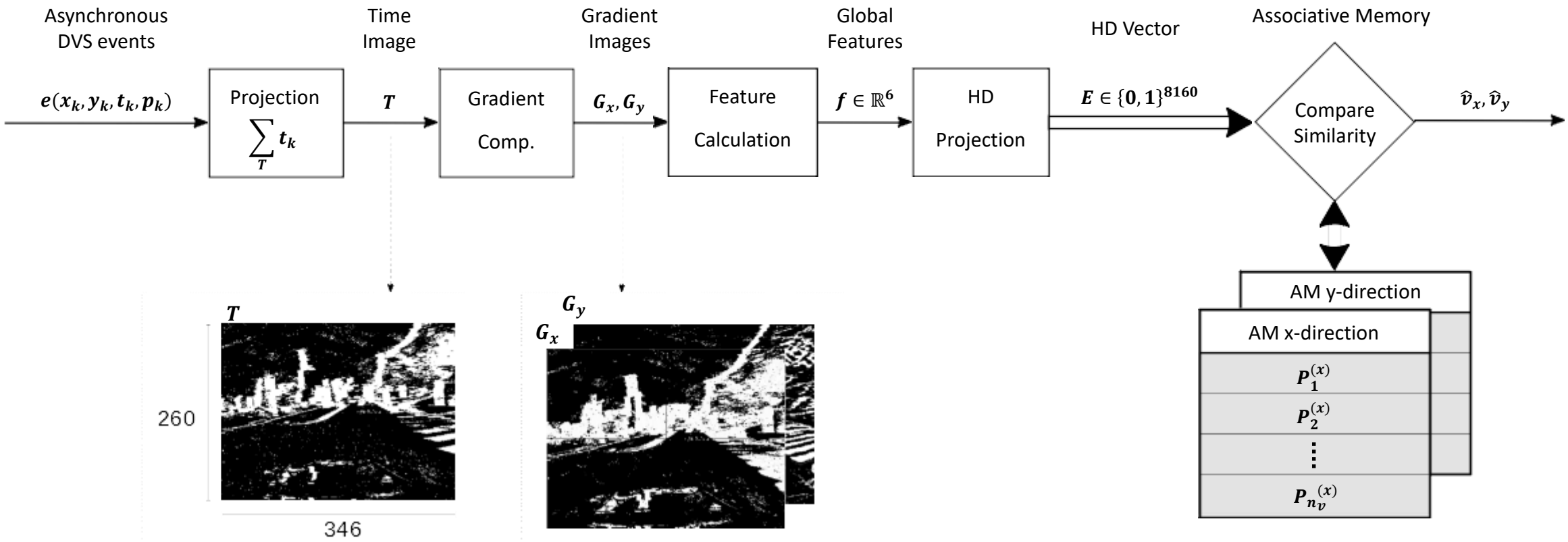


Prediction pipeline

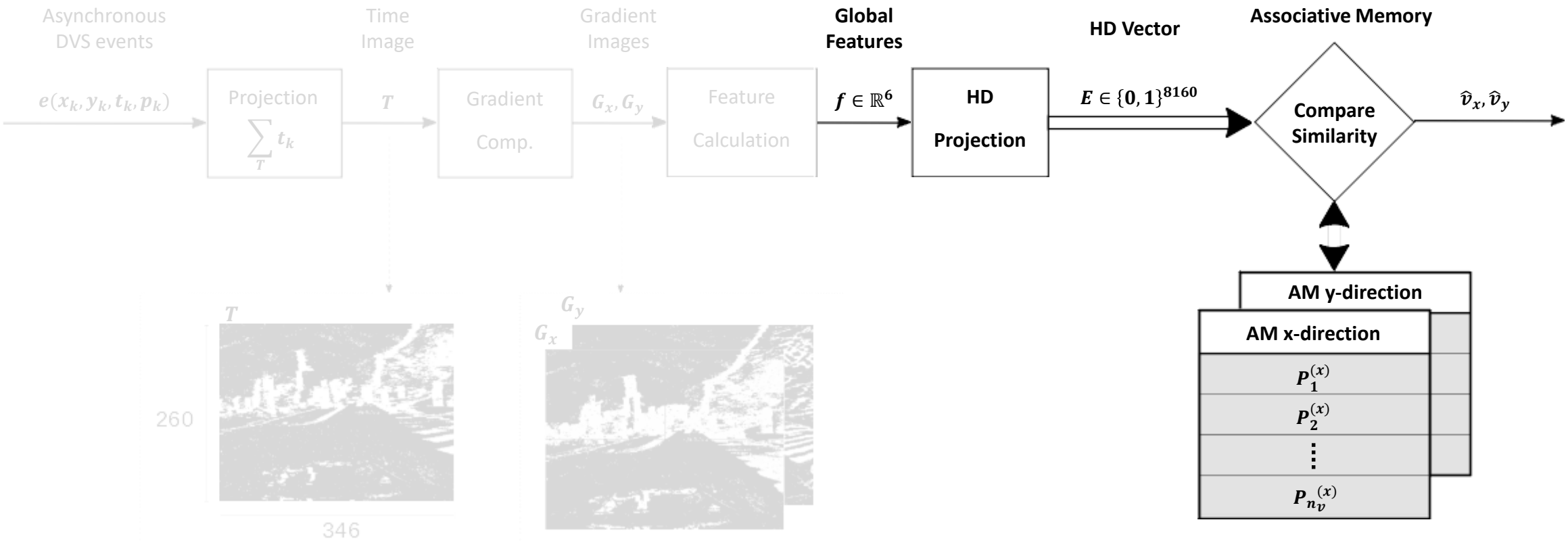




Prediction pipeline



Prediction pipeline



Prior work: Represent global features with binary dense vectors [2]

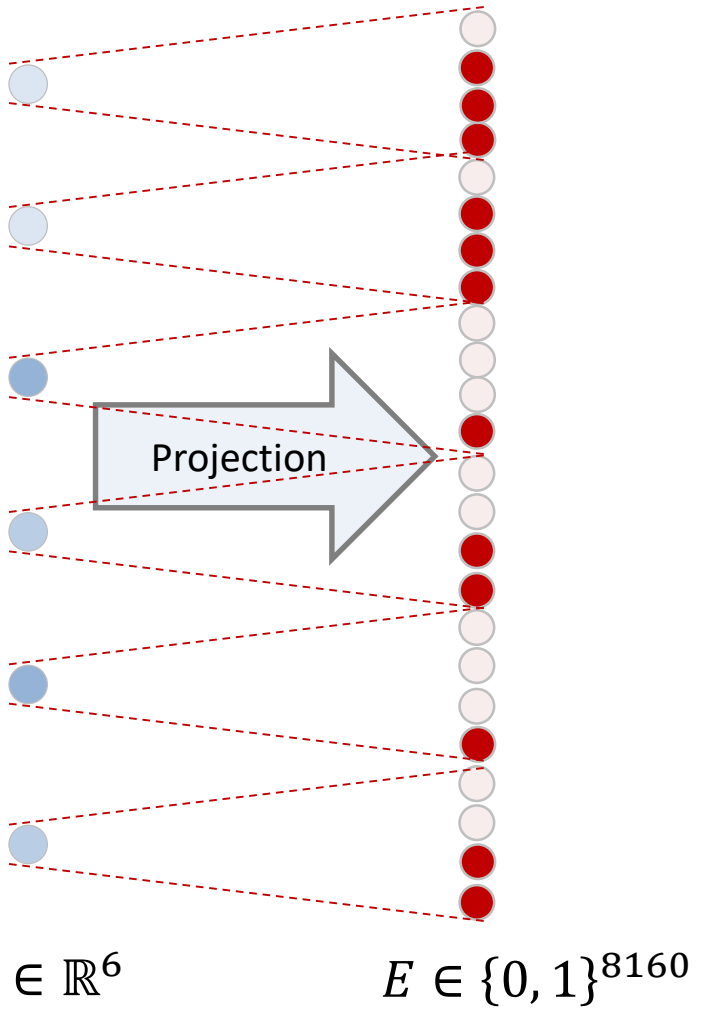
Thermometer coding

Each feature is mapped to a different section of the vector

Bundling with majority vote

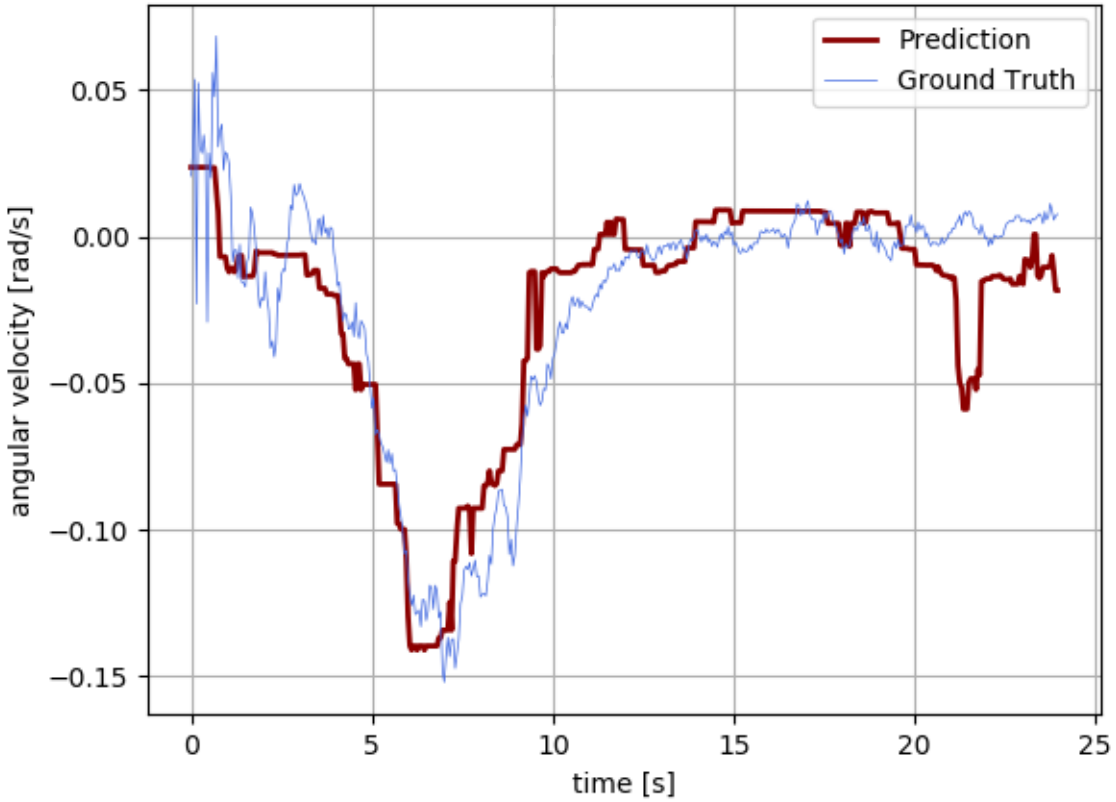
Majority vote between all vectors with same label

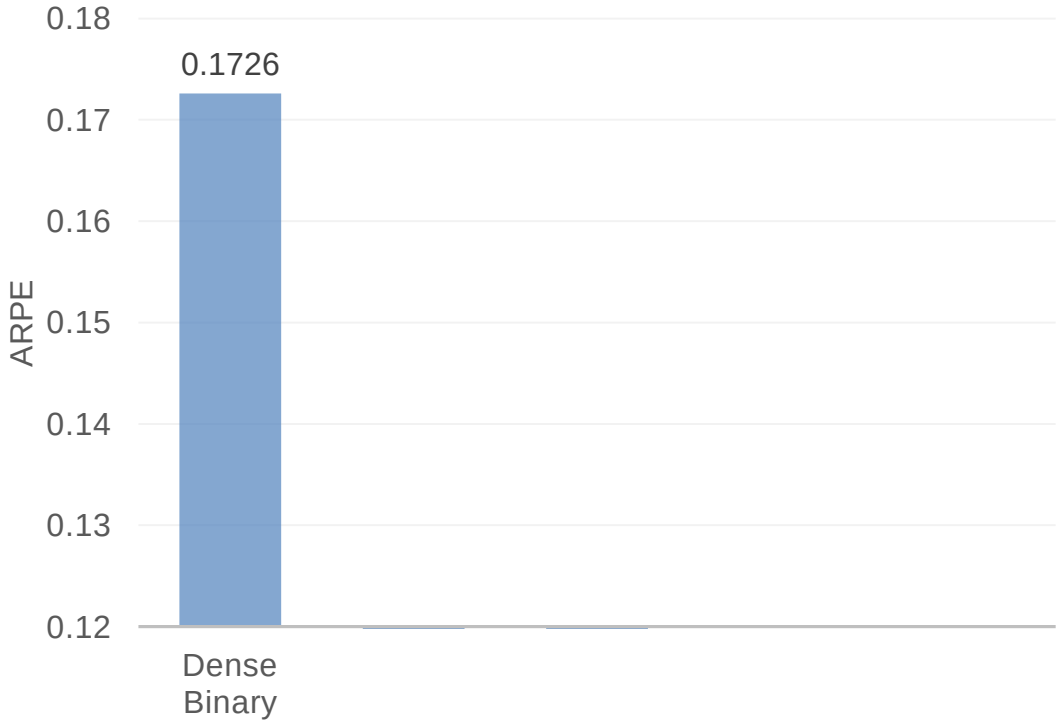
Hamming similarity for regression



[2] A. Mitrokhin, et al. 2019. Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception. *Science Robotics 4,* 30 (5 2019)

Prior work: Represent global features with binary dense vectors [2]





[2] A. Mitrokhin, et al. 2019. Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception. *Science Robotics 4,* 30 (5 2019)

HD sparse vectors for regression

Build **Associative Memories (AM)** using bundling operation

Bitwise OR

Infer **angular velocity** with similarity score

Overlap between AM and inference vector

HD sparse vectors for regression

Build **Associative Memories (AM)** using bundling operation

- **Bitwise OR**

Infer **angular velocity** with similarity score

- **Overlap** between AM and inference vector

Drawbacks

- **Sparsity loss in AM** increasing training set dimension

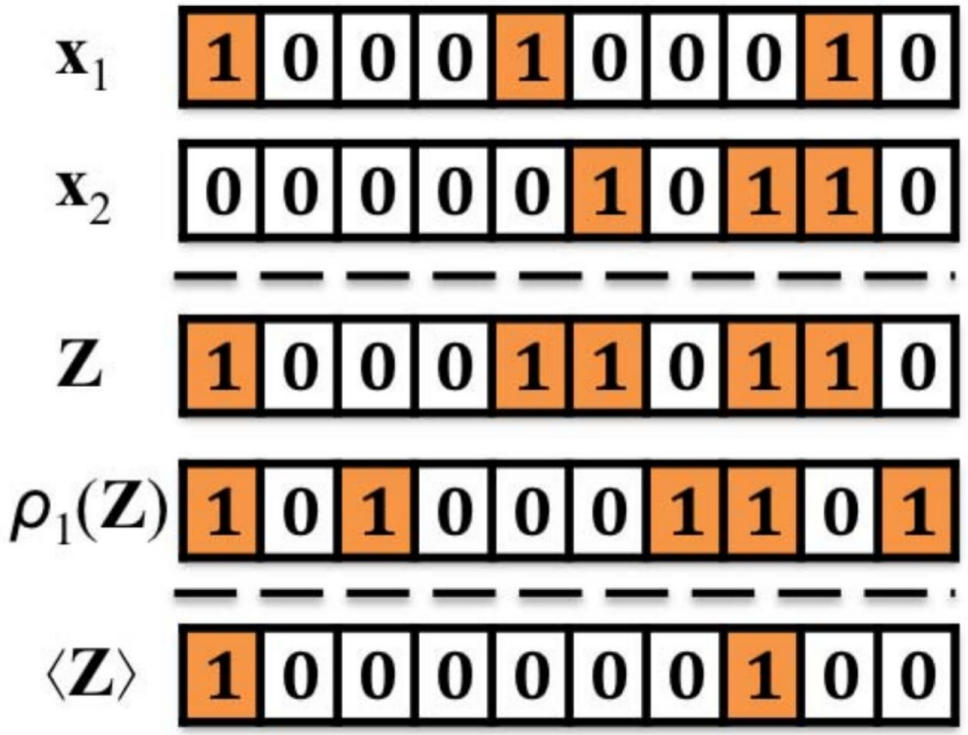
Context Dependent Thinning (CDT) [3] preserves sparsity

CDT operation

- **Permute** vector to be thinned
- Apply **AND** operation on it and its permutation
- **OR together** arbitrary number of vectors obtained with this operation

Resulting vector has reduced density

It has to be applied after bundling significant number of vectors



[3] D. A. Rachkovskij et al. Representation and processing of structures with binary sparse distributed code, *IEEE Trans. Knowl. Data Eng. 2001*

HD sparse vectors for regression

Build **Associative Memories (AM)** using bundling operation

- **Bitwise OR**

Infer **angular velocity** with similarity score

- **Overlap** between AM and inference vector

Drawbacks

- **Sparsity loss** increasing training set dimension

Sparse vectors encoding

- Selective Kanerva Coding (**SKC**)

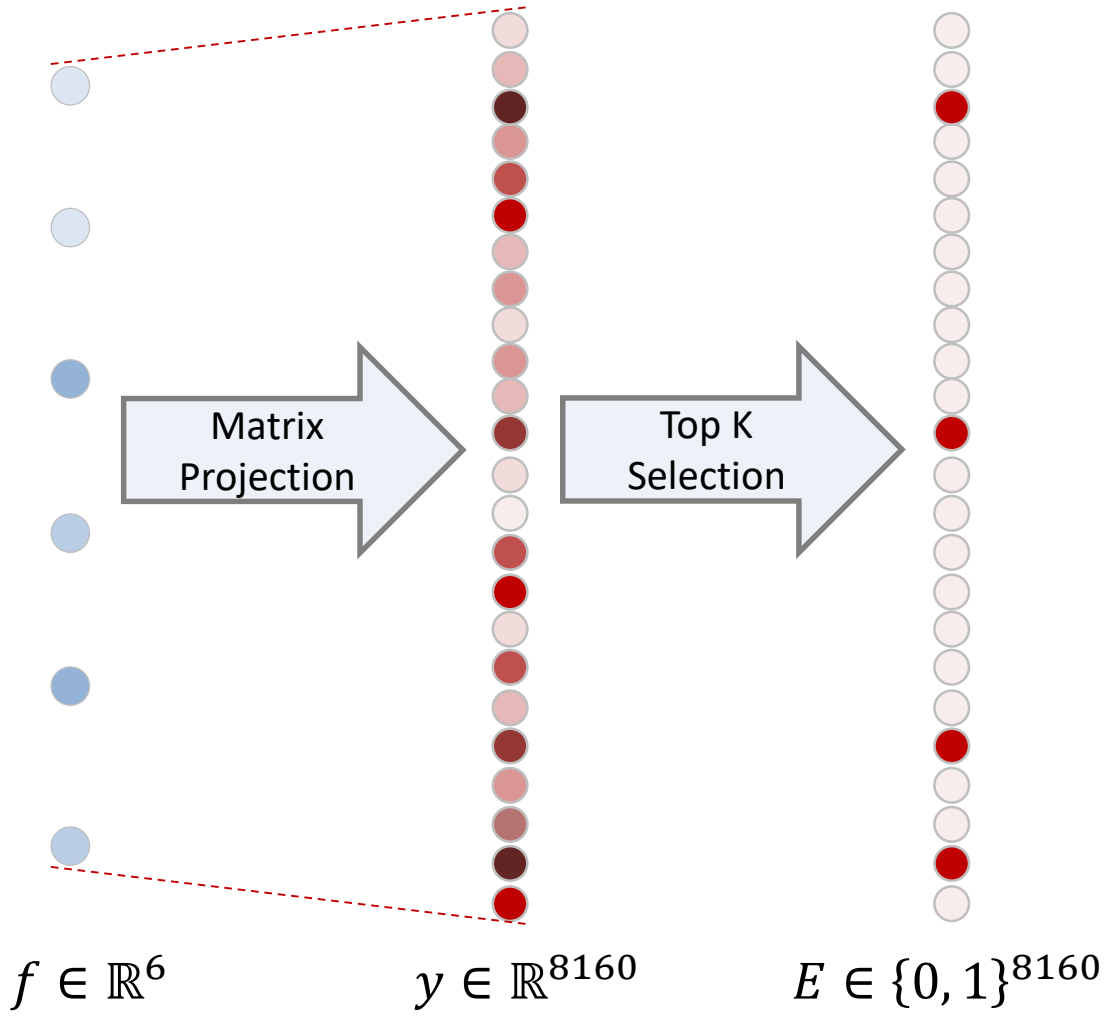
Selective Kanerva Coding

Global projection to HD vector

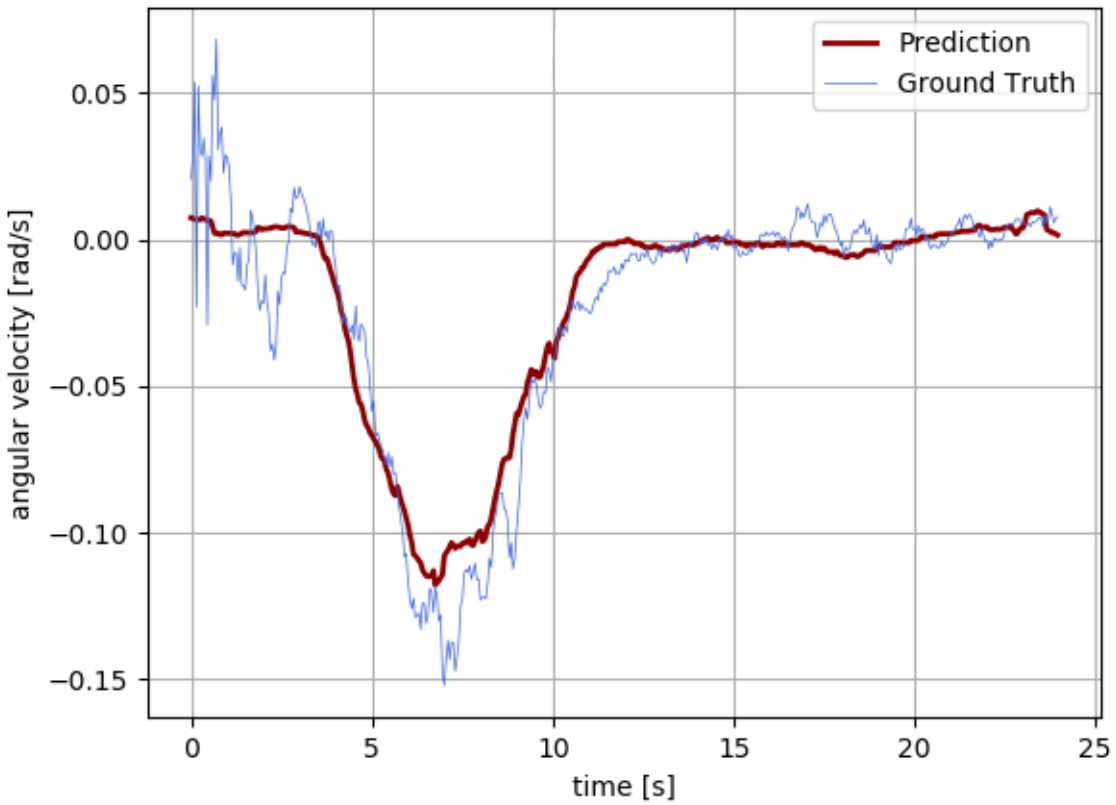
Projection matrix – feature vector product to give score vector

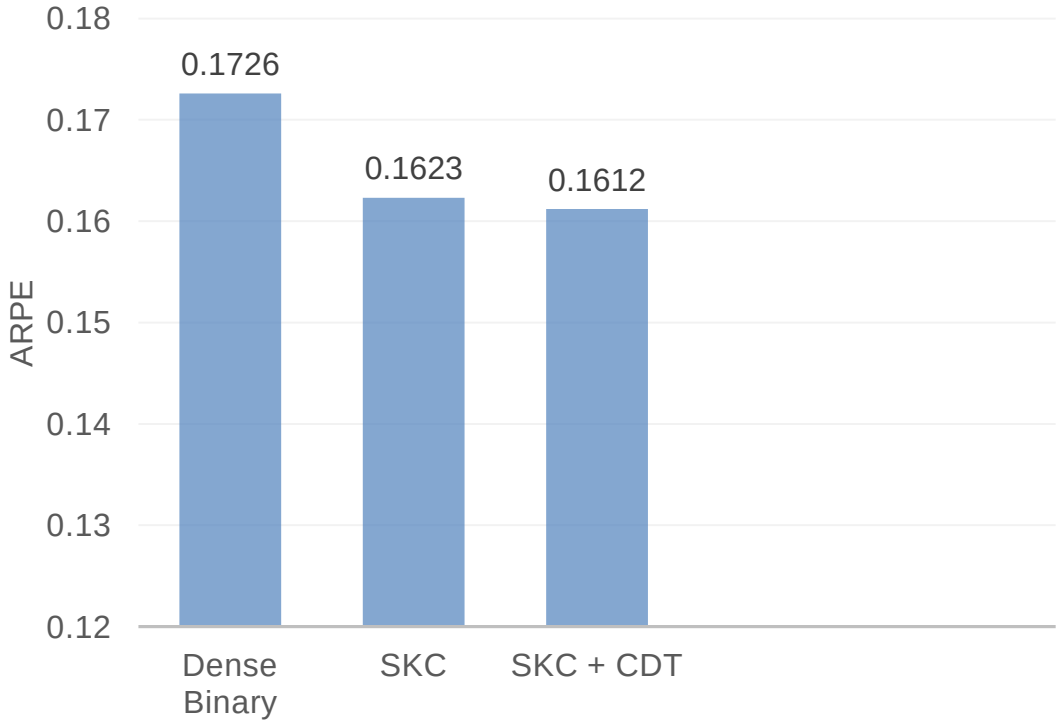
$y_i = < f, n_i >$
n_i: normally distributed random dictionary

Highest k scores set elements to 1 in the final sparse binary vector



Selective Kanerva Coding



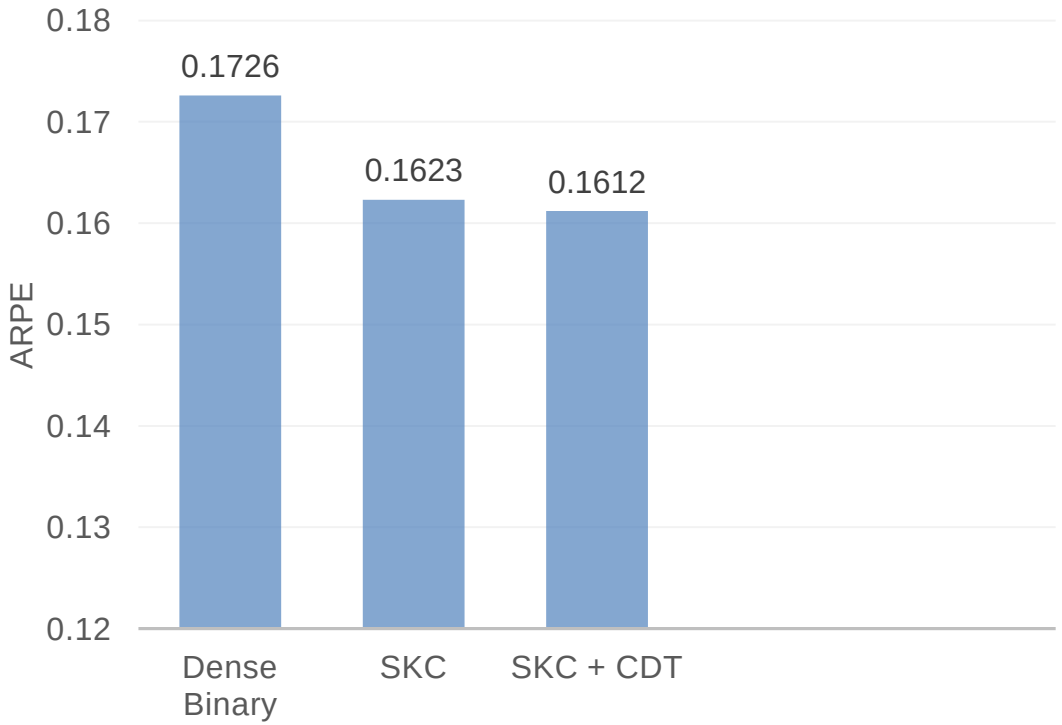


Selective Kanerva Coding comes with drawbacks

Drawbacks → High complexity

High dimensional non-binary matrix product

Position of 1s determined based on every score
Highest scores selection



Our Solution: Randomized Activation Functions Encoding

Local sparse vector projection

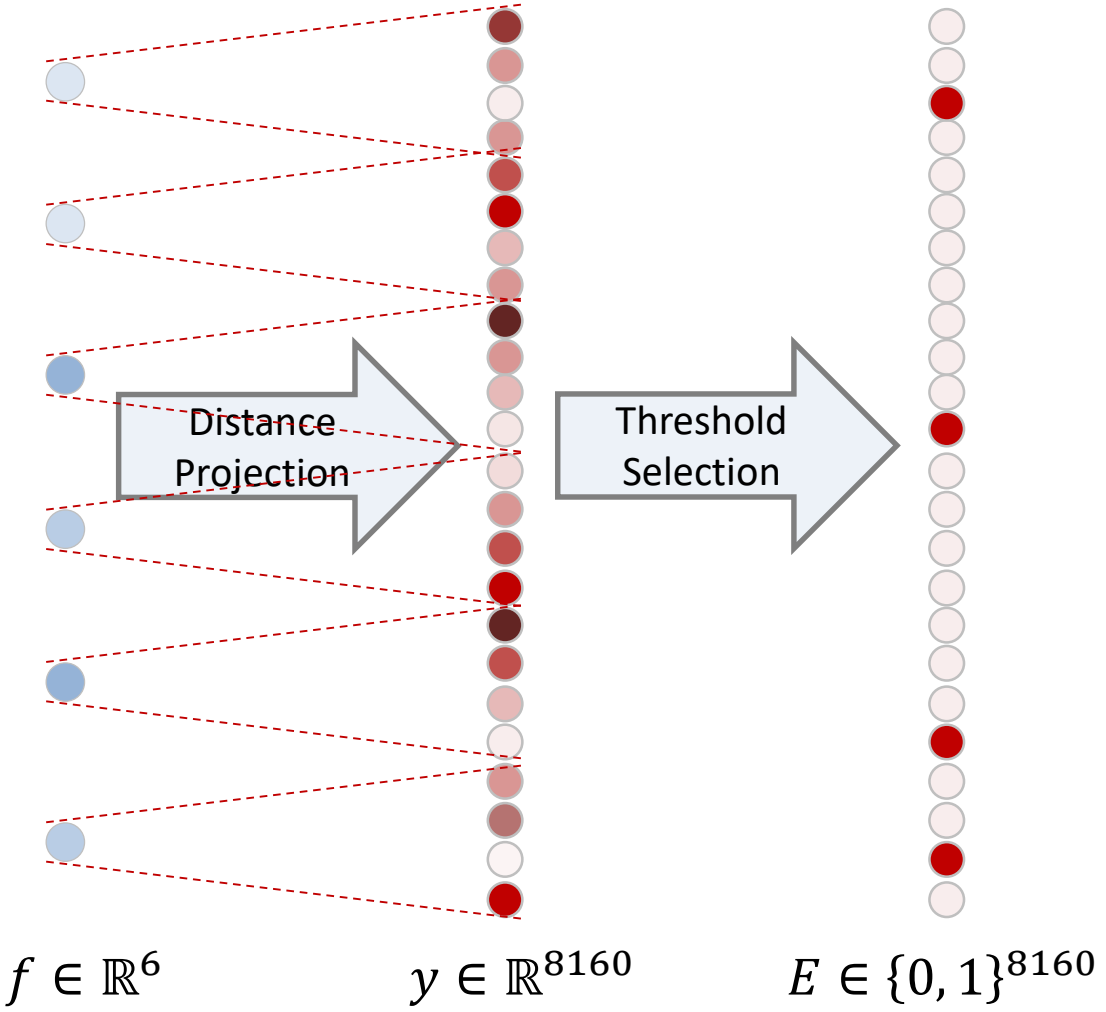
Randomized embedding vector to create vector sections for each feature

Threshold distance to determine 1s in vectors subsections

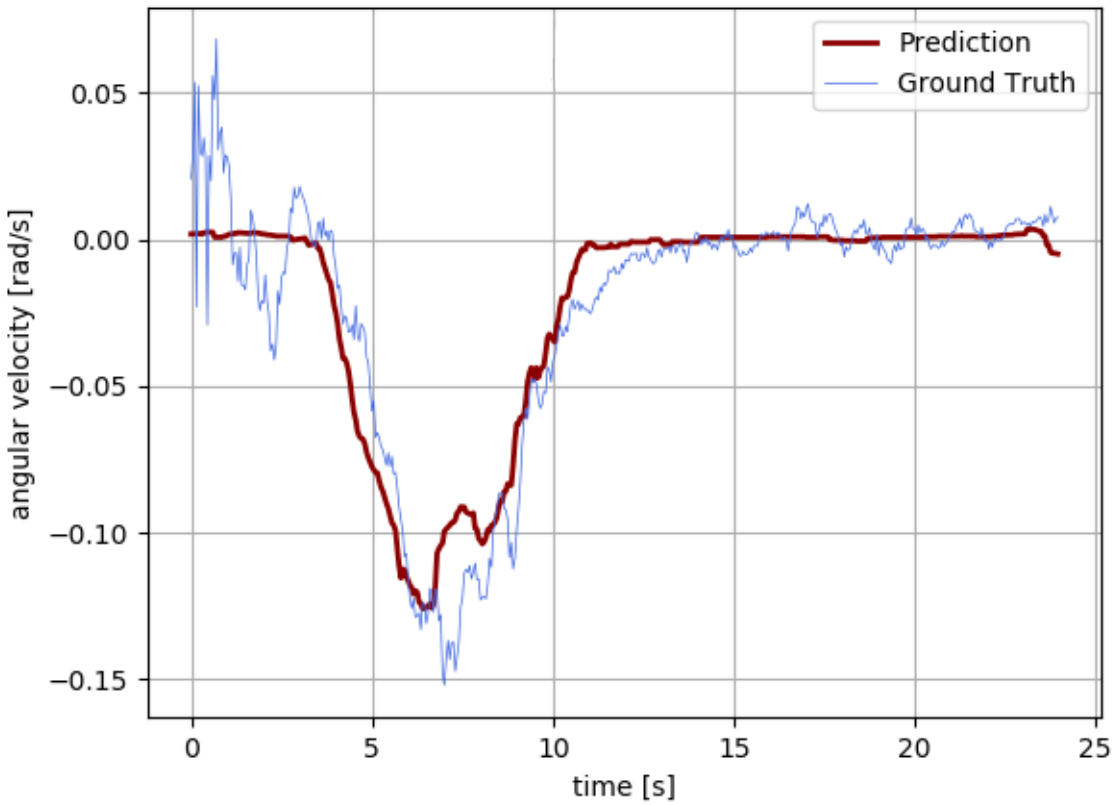
Features determine each element value independently → lightweight approach

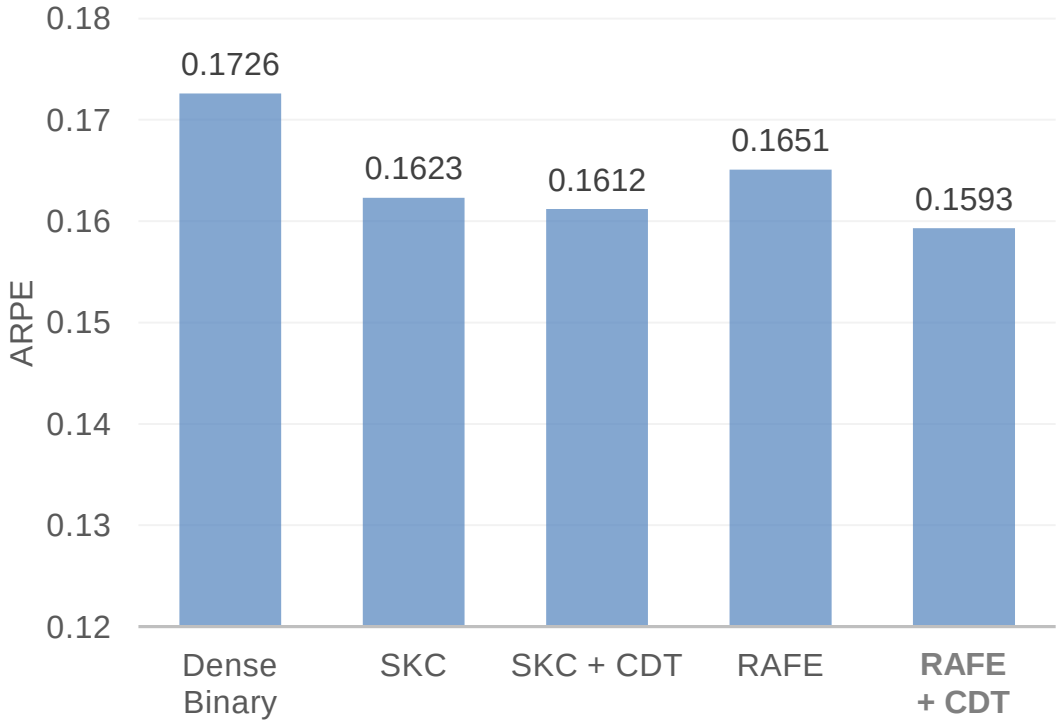
$y_i = |f_j - n_i|$

n_i: normally distributed random number

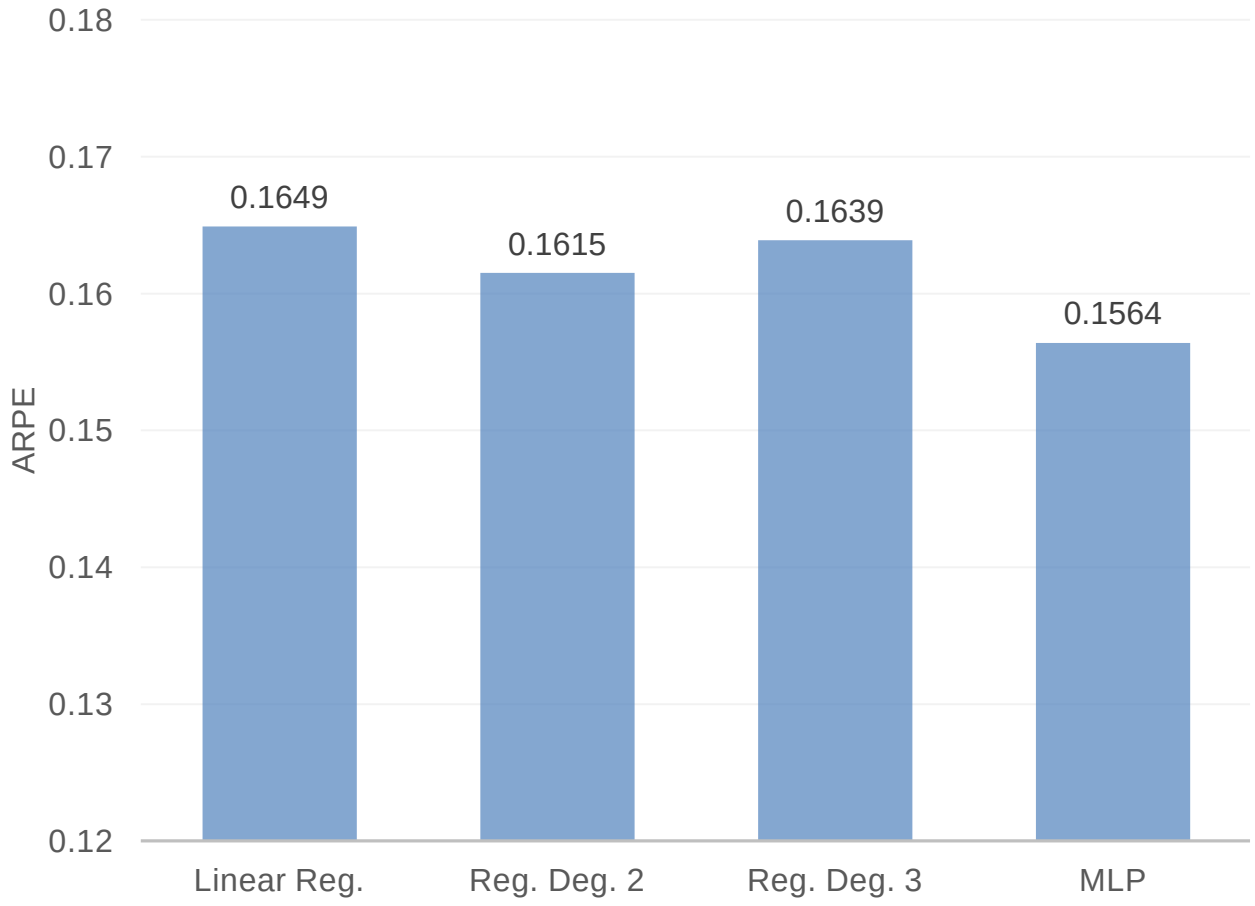


Randomized Activation Functions Encoding

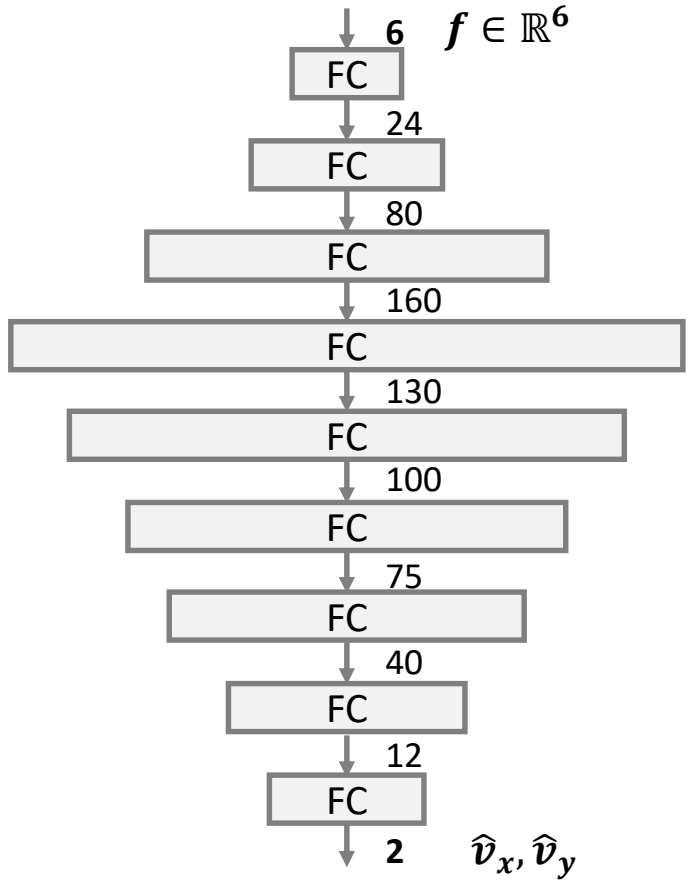




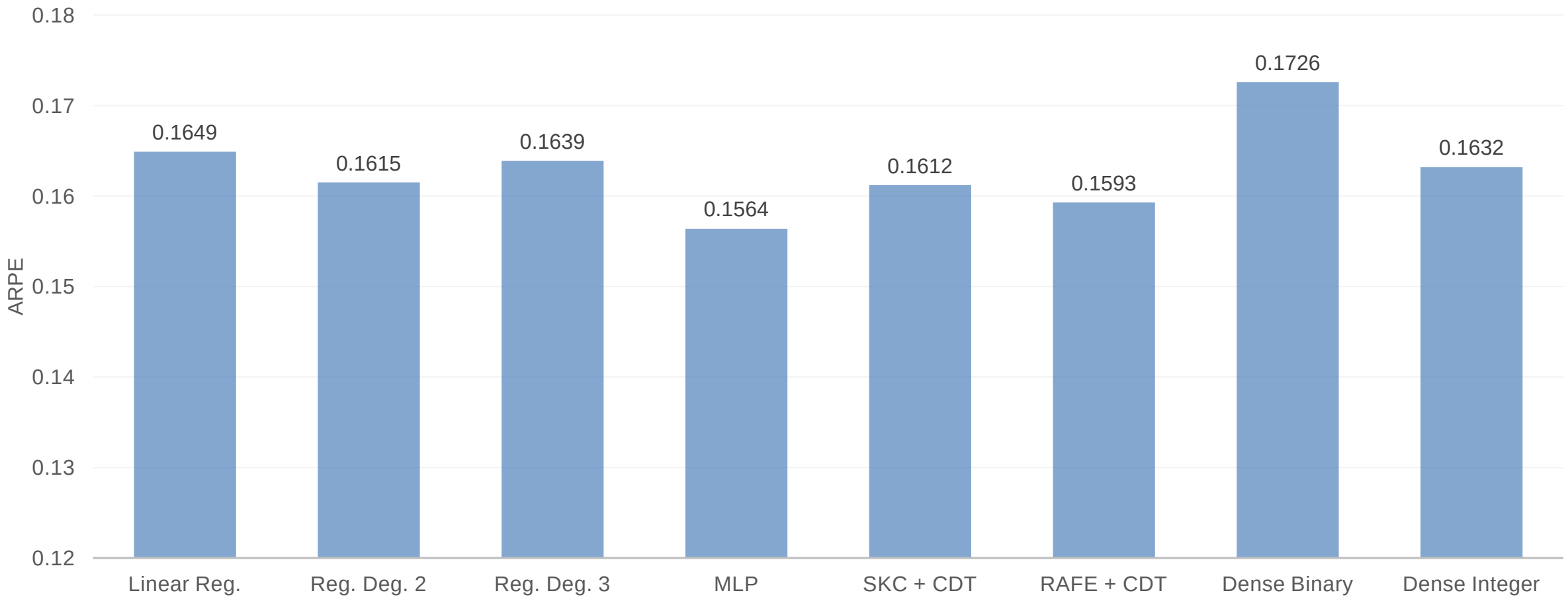
MLP and linear&polynomial regression



MLP Architecture



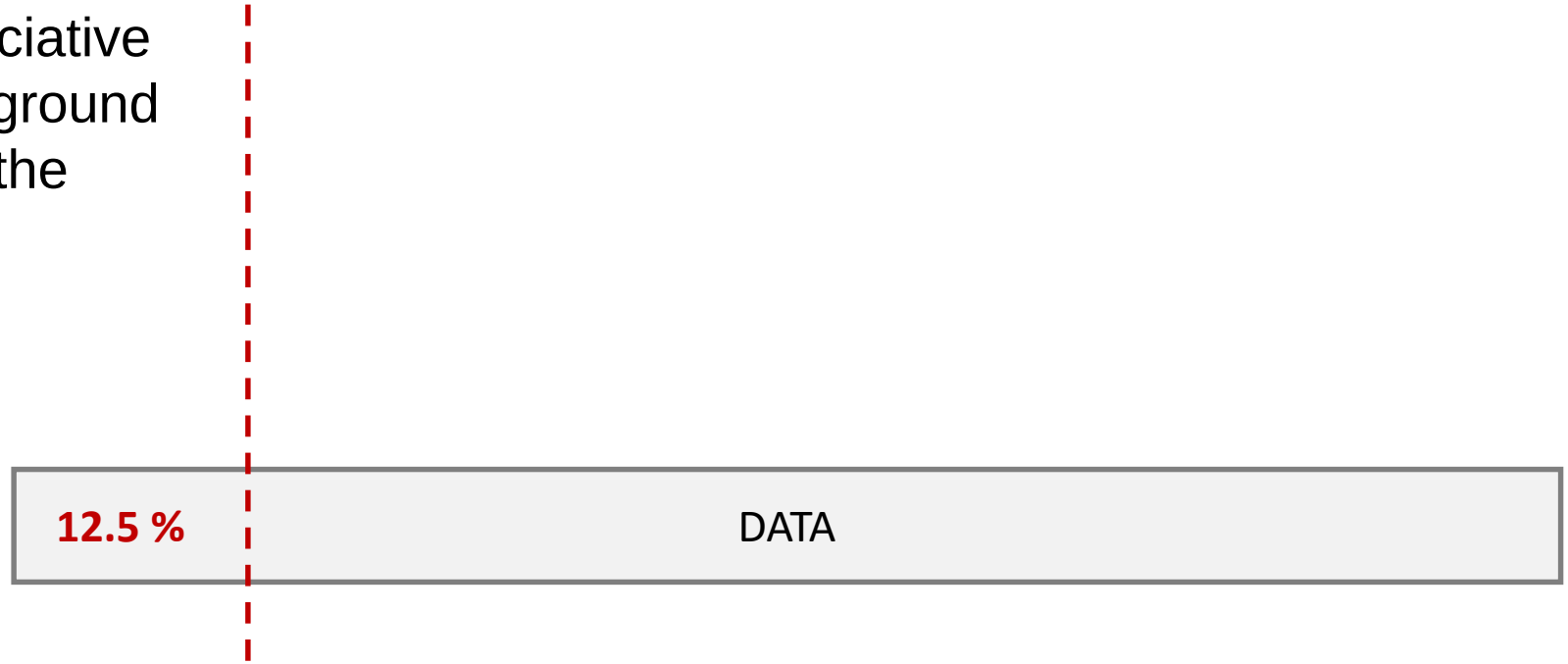
Regression performance comparison



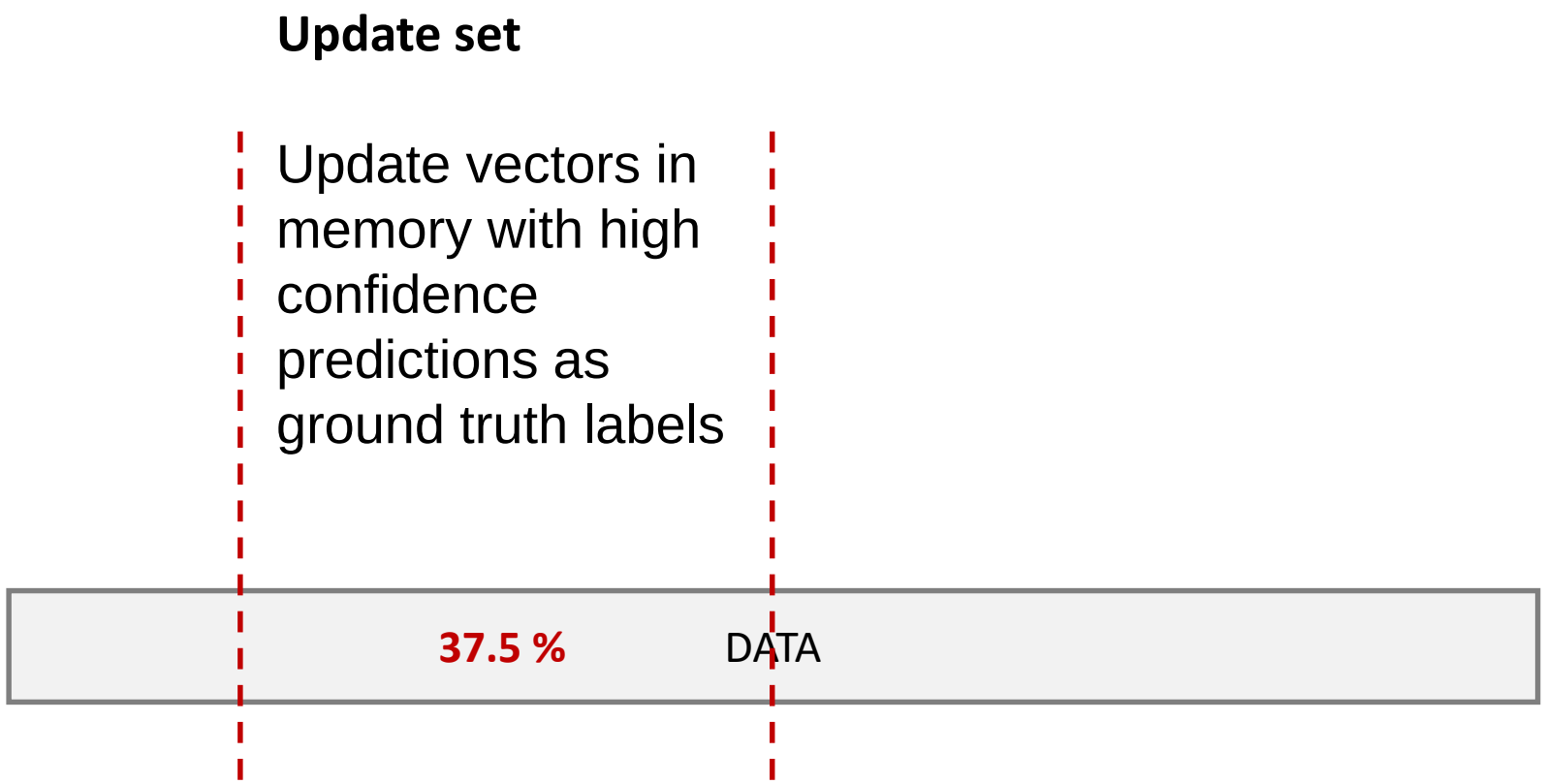
HD vectors allow online update to improve prediction

Train set

Construct associative memory using ground truth labels for the observations



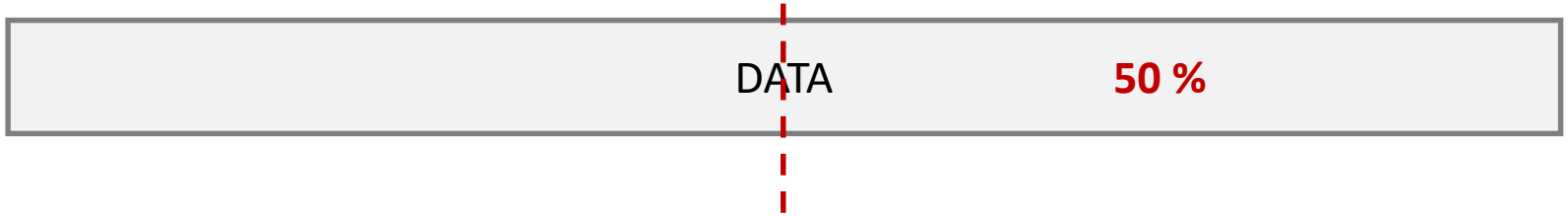
HD vectors allow online update to improve prediction



HD vectors allow online update to improve prediction

Test set

The complete model is tested on the complete test set



HD vectors allow online update to improve prediction

Dense binary vectors

Bundling → majority sum

Store number of 1s added in each position for each memory vector

Loose small binary memory requirements

High dimensional non-binary operations for each online update

HD vectors allow online update to improve prediction

Dense binary vectors

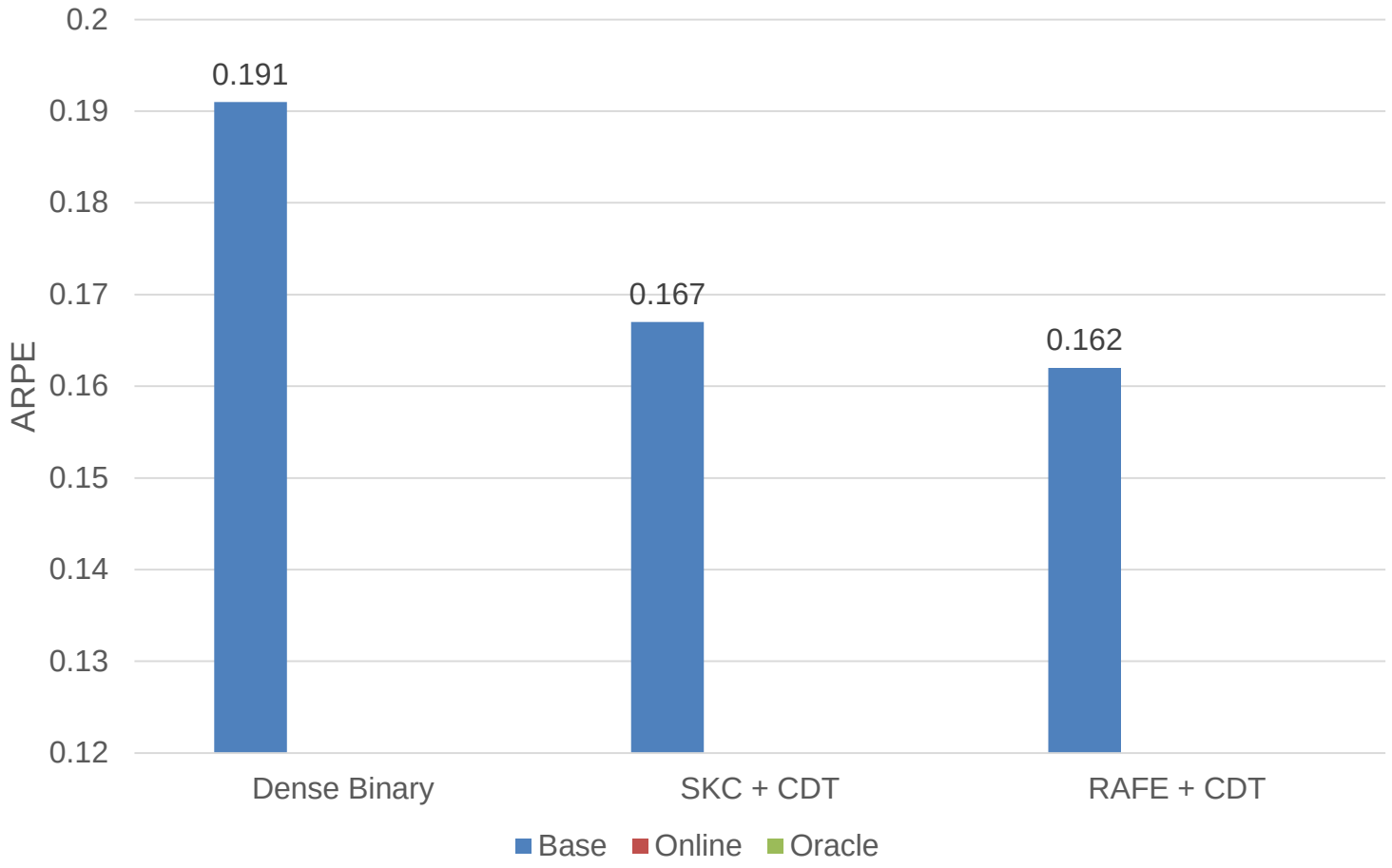
Bundling → majority sum
Store number of 1s added in each position for each memory vector
Loose small binary memory requirements
High dimensional non-binary operations for each online update

Sparse binary vectors

Bundling → bitwise OR
No memory overhead and small additional cost
Perform CDT after many online updates → sporadic added complexity

HD vectors allow online update to improve prediction

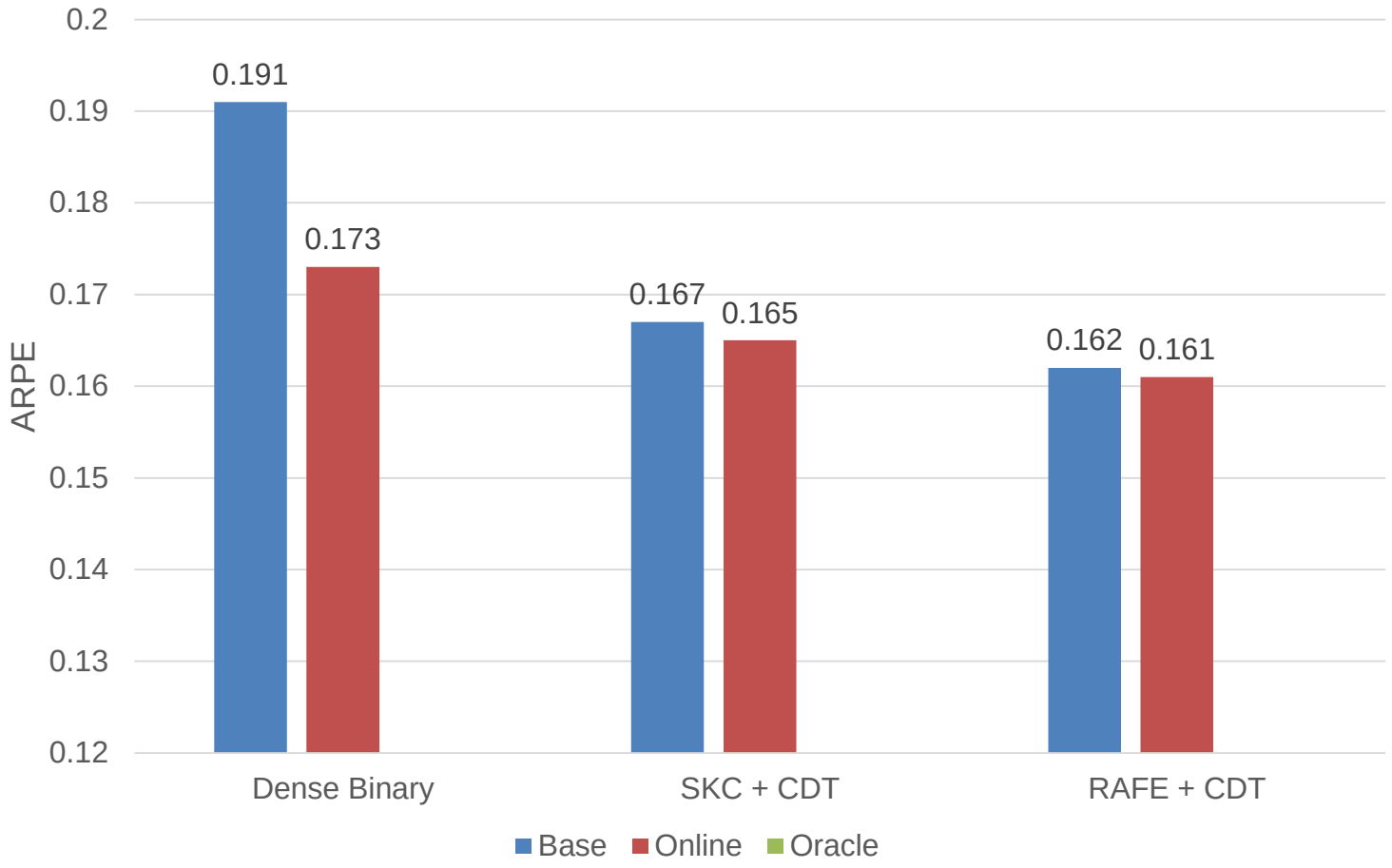
Model trained only on small **train set**
→ **Base**



HD vectors allow online update to improve prediction

Model trained only on small **train set**
→ **Base**

Model updated using online high confidence prediction on **update set**
→ **Online**

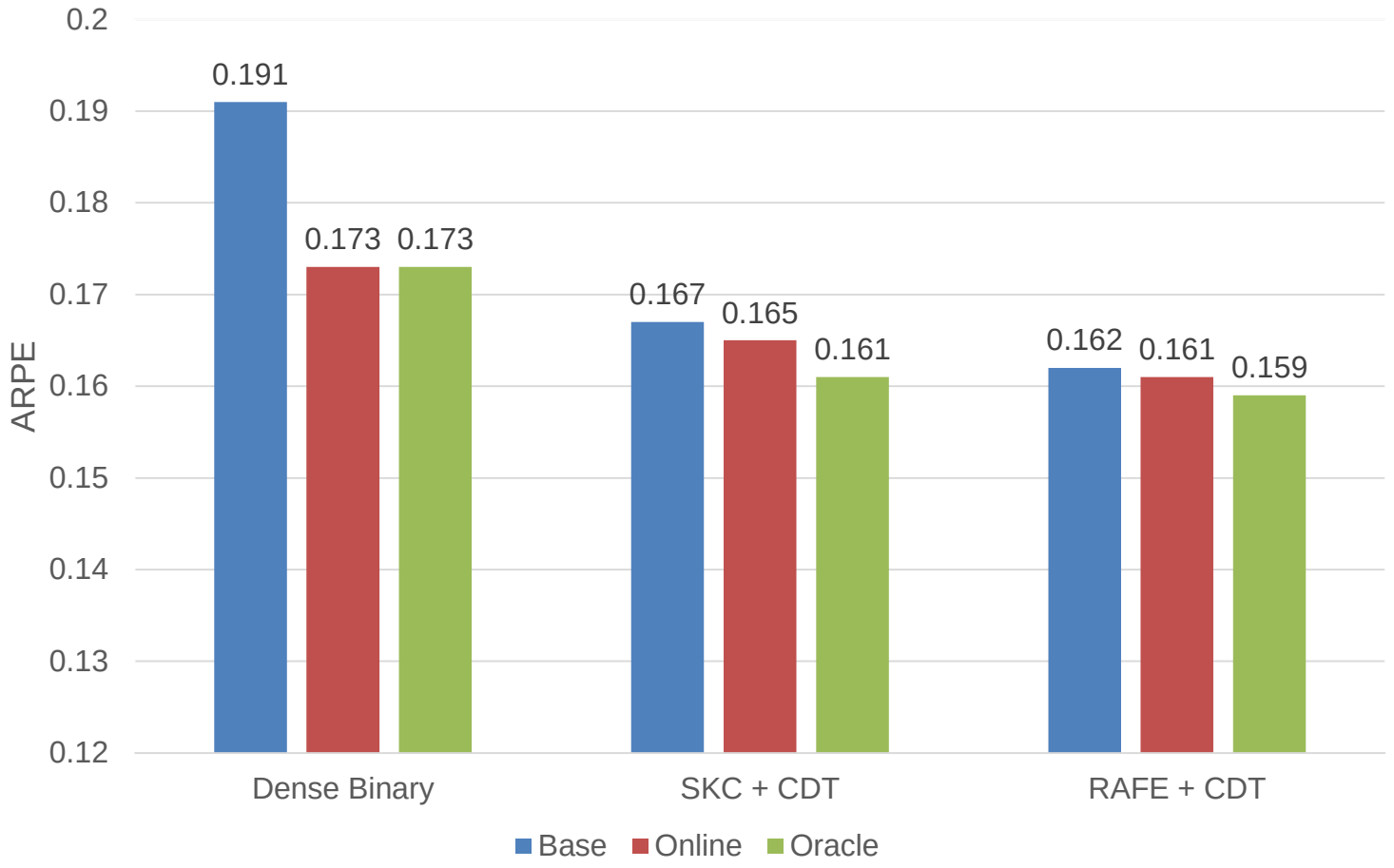


HD vectors allow online update to improve prediction

Model trained only on small **train set** → **Base**

Model updated using online high confidence prediction on **update set** → **Online**

Model updated using ground truth labels on **update set** → **Oracle**



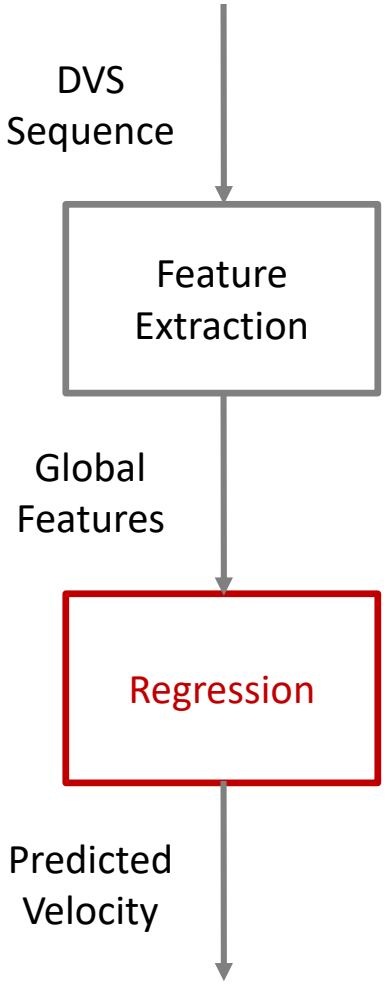
Embedded accelerator and comparison

C implementation of regression inference

- **Dense Binary**
- **SKC**
- **RAFE**
- **MLP**

Execution on **GAP8** [1] board

- RISC-V based ultra-low-power SoC
- 8 core computational cluster
- 100 MHz clock for highest efficiency



[1] E. Flamand, et al.2018. GAP-8: A RISC-V SoC for AI at the Edge of the IoT. In *2018 IEEE 29th ASAP. 1–4*

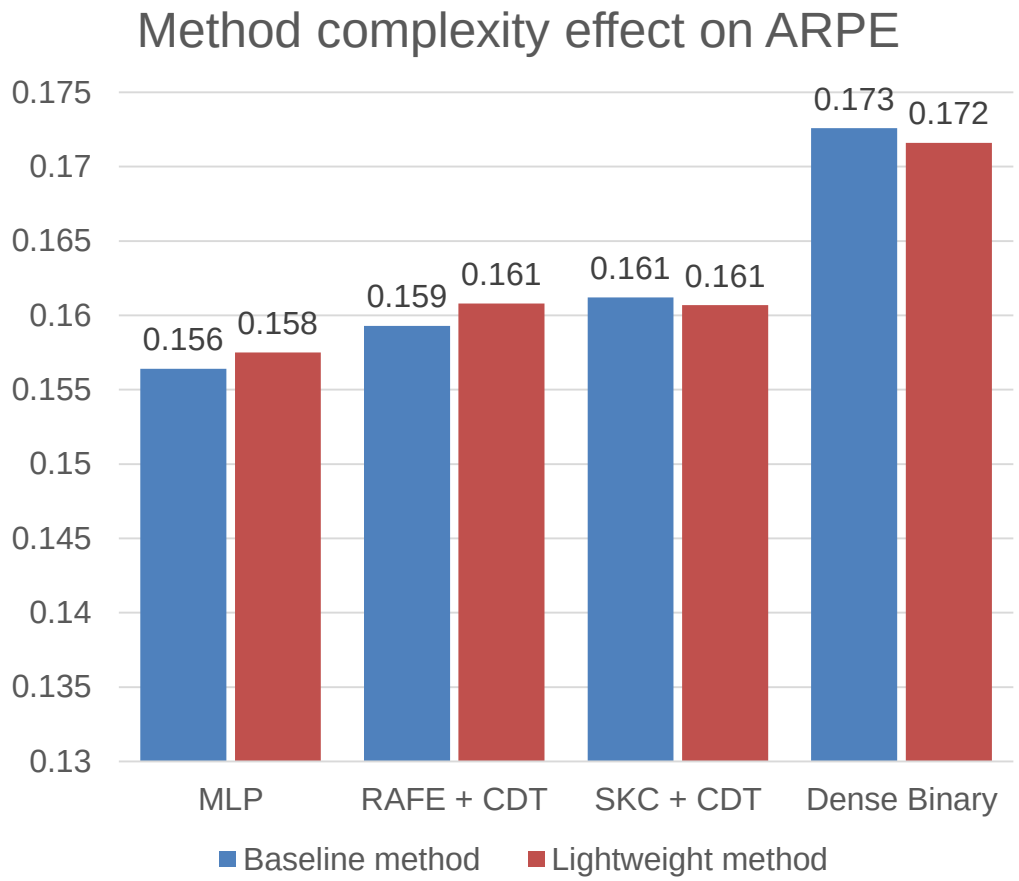
Embedded accelerator and comparison

HD vector methods implementation

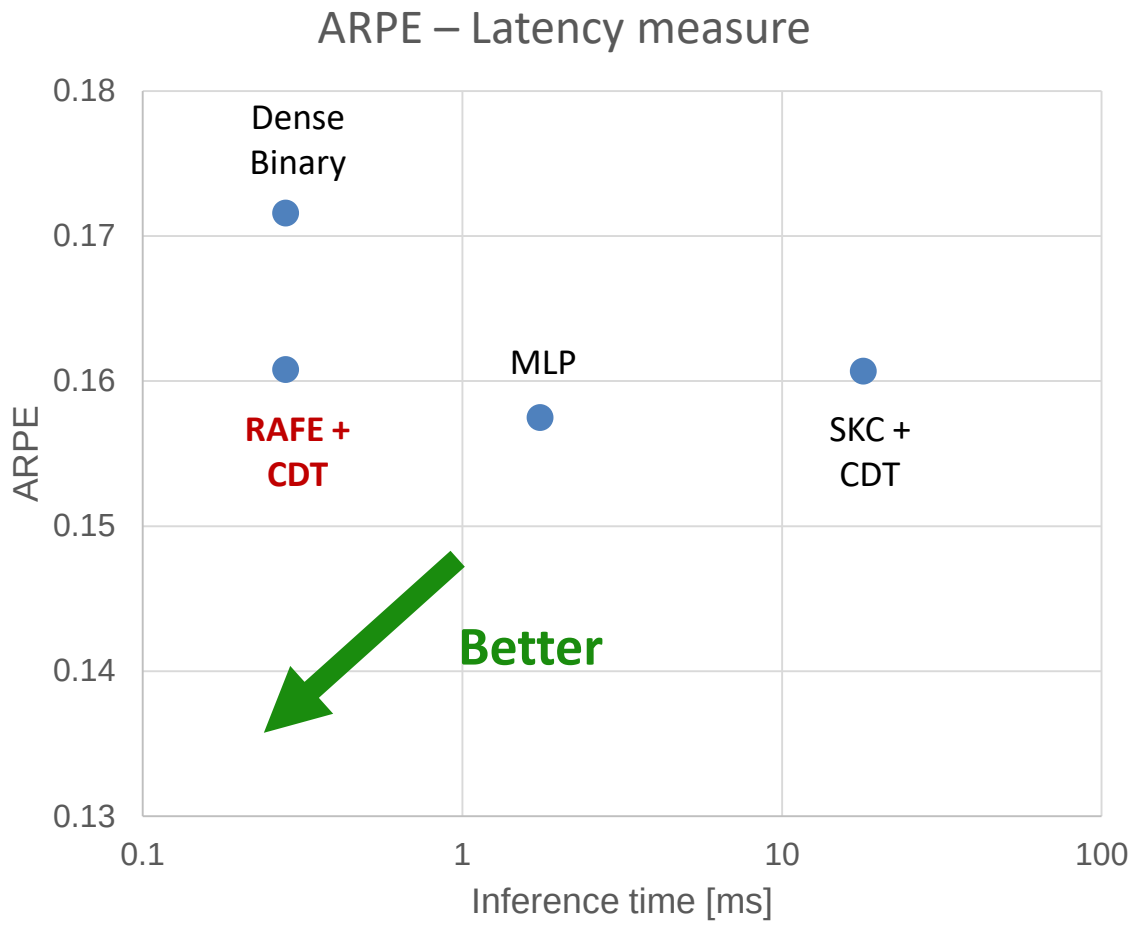
- Dimension reduction for higher performance
- Marginal accuracy loss
- Parallelized features encoding and AM search

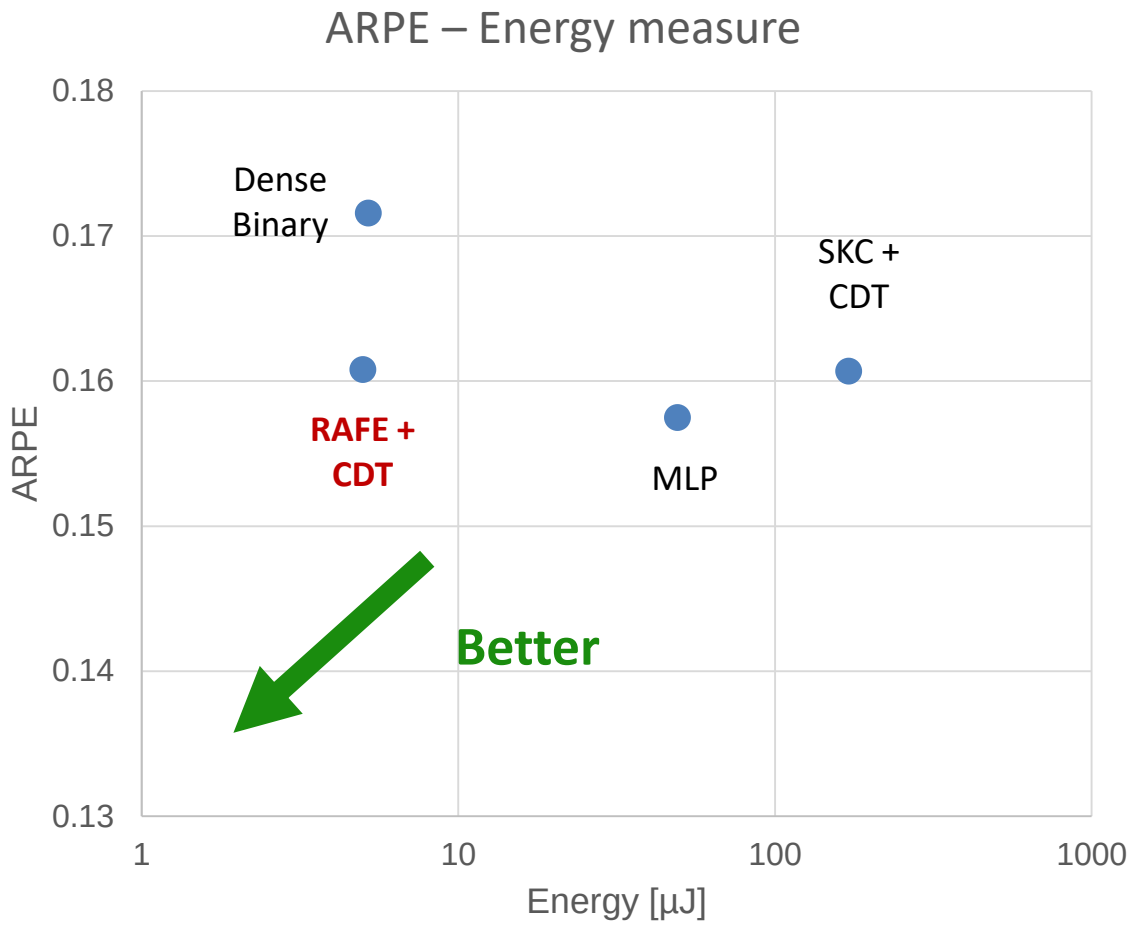
MLP implementation

- FANN based implementation
- Fixed point precision to reduce complexity
- Marginal accuracy loss



Embedded accelerator and comparison





Conclusion

Effective use of **HD sparse vectors** for regression on **event-based DVS** data

Conclusion

Effective use of **HD sparse vectors** for regression on **event-based DVS** data

Improvement of **7.7%** over **Dense Binary** vectors with **RAFE** on **HD sparse vectors** applied with **CDT**, with only **MLP** having **1.8% better ARPE**

Conclusion

Effective use of **HD sparse vectors** for regression on **event-based DVS** data

Improvement of **7.7%** over **Dense Binary** vectors with **RAFE** on **HD sparse vectors** applied with **CDT**, with only **MLP** having **1.8% better ARPE**

Application of **online update** to improve prediction by up to **9% with Dense Binary vectors** and to cope with small datasets

Conclusion

Effective use of **HD sparse vectors** for regression on **event-based DVS** data

Improvement of **7.7%** over **Dense Binary** vectors with **RAFE** on **HD sparse vectors** applied with **CDT**, with only **MLP** having **1.8% better ARPE**

Application of **online update** to improve prediction by up to **9% with Dense Binary vectors** and to cope with small datasets

Embedded accelerator achieving **5.0 µJ** of **energy** per inference with **RAFE**, being **9.84× more efficient** than **MLP** at comparable ARPE (0.1608 vs. 0.1575)

Thank you

Our code is available!

https://github.com/iis-eth-zurich/hd_dvs