UCI Samueli
School of Engineering

DOMINO: Domain-Invariant Hyperdimensional Classification for Multi-Sensor Time Series Data

Junyao Wang, Luke Chen, Mohammad Abdullah Al Faruque
University of California, Irvine

IEEE/ACM
2023 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN
ICCAD
42nd Edition

Multi-Sensor Time Series Data

- Rapid evolution of the Internet of Things (IoT)
- Heterogeneously connected sensors capture information over time, constituting muti-sensor time series data
- Sophisticated DNNs, e.g., RNNs, have been proposed to capture spatial and temporal dependencies in these data
- Hyperdimensional Computing (HDC) has been introduced
  - ✓ Fast convergence
  - ✓ High computational efficiency
  - ✓ Ultra-robustness against noise

Too complicated for resource-constrained edge platforms!

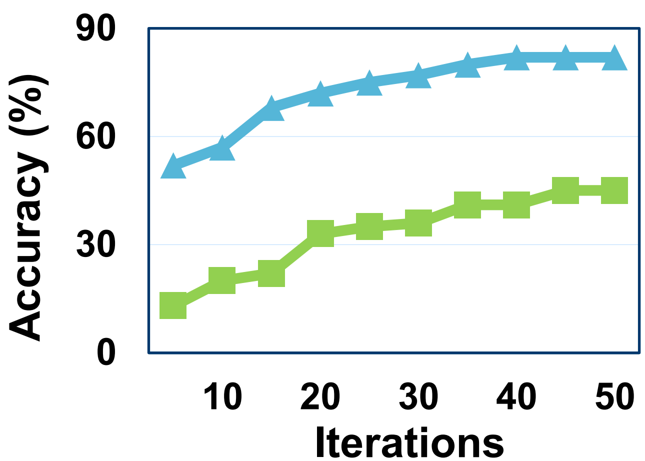
Distribution Shift

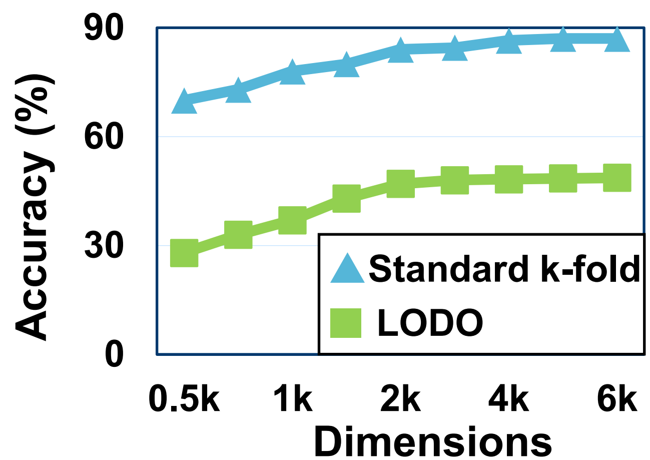
- Here we define the problem:
  - Suppose we have k $(k \geq 1)$ source domains in training data $\mathcal{X}_{\mathcal{S}}$ and an output space $\mathcal{Y}$
  - our goal is to train a classification model f: $\mathcal{X}_{\mathcal{S}} \rightarrow \mathcal{Y}$
  - f should be able to capture latent features and make predictions for data samples from test data/target domain $\mathcal{X}_{\mathcal{T}}$
- Challenge
  - The distributions in source domains and target domains can be different, $\mathcal{P}_{(\mathcal{X}_{\mathcal{S}},\mathcal{Y})} \neq \mathcal{P}_{(\mathcal{X}_{\mathcal{T}},\mathcal{Y})}$
  - A model trained with samples in $\mathcal{X}_{\mathcal{S}}$ may fail to adapt to samples from $\mathcal{X}_{\mathcal{T}}$

Challenges

- Existing HDCs are not immune to the distribution shift issue
  - The accuracy of leave-one-domain-out (LODO) cross-validation (CV) is notably lower than k-fold CV
  - LODO CV: training a model on all the available data except for one domain that is left out for inference
  - k-fold CV: randomly divides all data into k subsets with $k-1$ subsets for training and the remaining one for inference
- ***Can we filter out those domain-variant dimensions?***

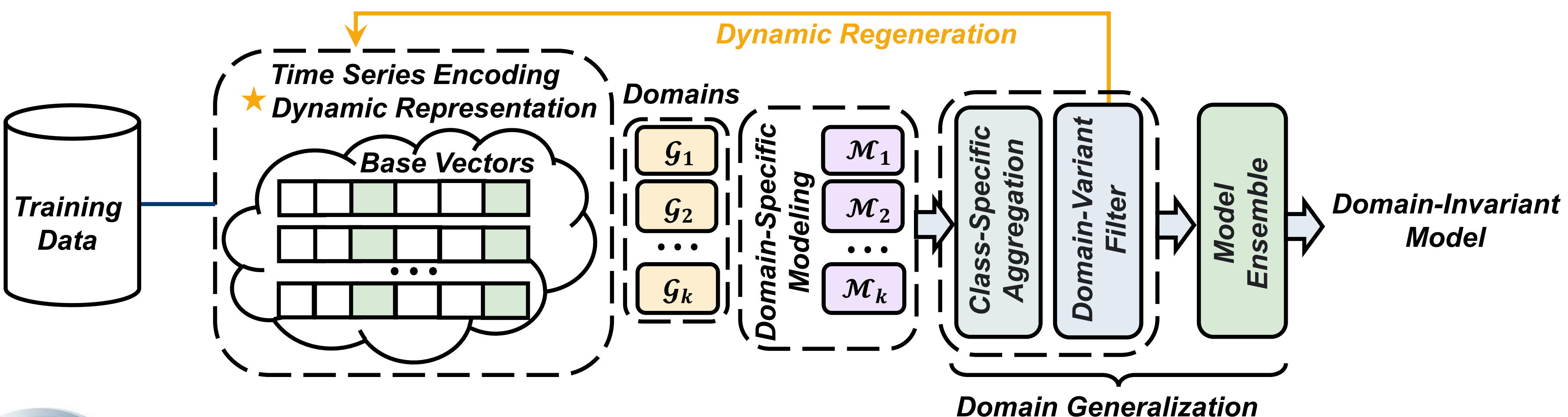
a very limited domain generalization capability





Our Work

- ***DOMINO***, the first HDC-based domain generalization algorithm that dynamically identifying and regenerating domain-variant dimensions.
  - Time Series Encoding
  - Domain Specific Modeling
  - Domain Generalization

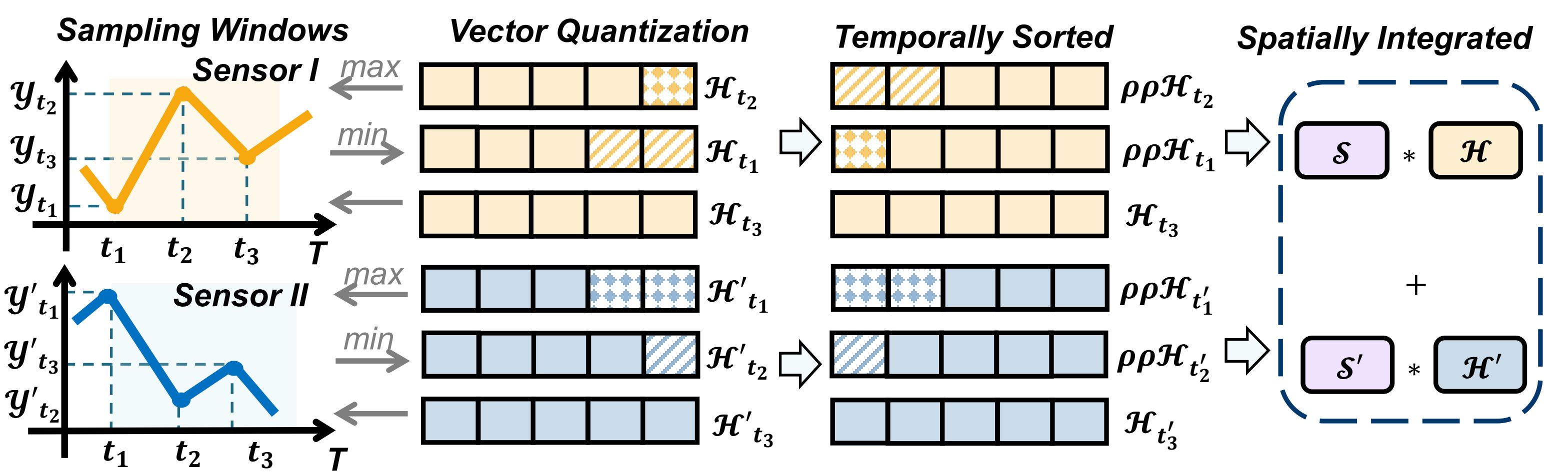


Time Series Encoding

- HDC starts with encoding low-dimensional inputs to hypervectors, each containing thousands of elements.
- Assign random hypervectors $\mathcal{H}_{max}$ and $\mathcal{H}_{min}$ to represent the maximum and minimum signal values, i.e., $\mathcal{Y}_{max}$ and $\mathcal{Y}_{min}$.
- Vector quantization to values between $\mathcal{Y}_{max}$ and $\mathcal{Y}_{min}$. For instance,

$$\mathcal{H}_{t_3} = \mathcal{H}_{t_1} + \frac{y_{t_3} - y_{t_1}}{y_{t_2} - y_{t_1}} \cdot (\mathcal{H}_{t_2} - \mathcal{H}_{t_1});\ \mathcal{H}'_{t_3} = \mathcal{H}'_{t_2} + \frac{y'_{t_3} - y'_{t_2}}{y'_{t_1} - y'_{t_2}} \cdot (\mathcal{H}'_{t_1} - \mathcal{H}'_{t_2})$$

- Temporally sorting by rotation shifts (ρ), e.g., $\mathcal{H} = \rho\rho\mathcal{H}_{t_1} * \rho\mathcal{H}_{t_2} * \mathcal{H}_{t_3}$
- Spatially integrating by binding, e.g., $\mathcal{H} = \mathcal{S}_1 * \mathcal{H}_1 + \cdots + \mathcal{S}_n * \mathcal{H}_n$



Domain-Specific Modeling

- Construct one HDC model for each subset
  - Based on cosine similarity between $\mathcal{H}$ and class hypervectors:

$$\delta(\mathcal{H}, \mathcal{C}_i) = \frac{\mathcal{H} \cdot \mathcal{C}_i}{\|\mathcal{H}\| \cdot \|\mathcal{C}_i\|} = \frac{\mathcal{H}}{\|\mathcal{H}\|} \cdot \frac{\mathcal{C}_i}{\|\mathcal{C}_i\|} \propto \mathcal{H} \cdot Normalize(\mathcal{C}_i)$$

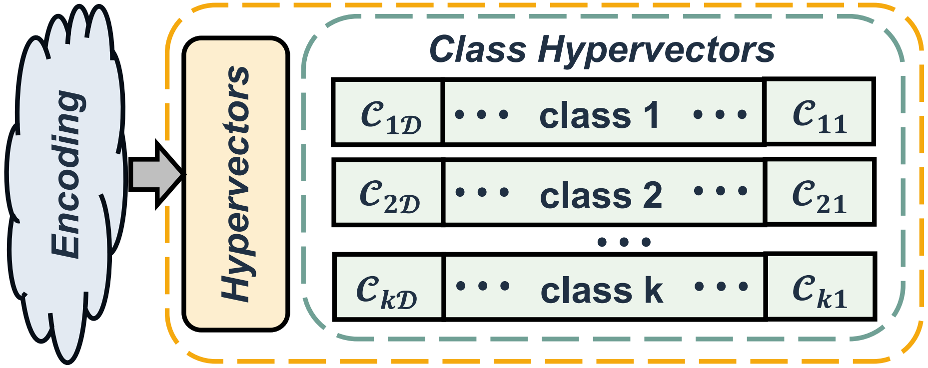
  - Scaling a weight to each sample depending on how much new information is added to class hypervectors. If $\mathcal{H}$ has highest cosine-similarities to $\mathcal{C}_i$ while its true label is $\mathcal{C}_j$:

$$\mathcal{C}_i \leftarrow \mathcal{C}_i - \eta \cdot [1 - \delta(\mathcal{H}, \mathcal{C}_i)] \times \mathcal{H}$$

$$\mathcal{C}_j \leftarrow \mathcal{C}_j + \eta \cdot \left[1 - \delta\left(\mathcal{H}, \mathcal{C}_j\right)\right] \times \mathcal{H}$$

A large $\delta(\mathcal{H}, \cdot)$ indicates the input data points is marginally mismatches, while a small $\delta(\mathcal{H}, \cdot)$ indicates a noticeably new pattern.

Our algorithm provides a higher chance for non-common patterns to be considered.

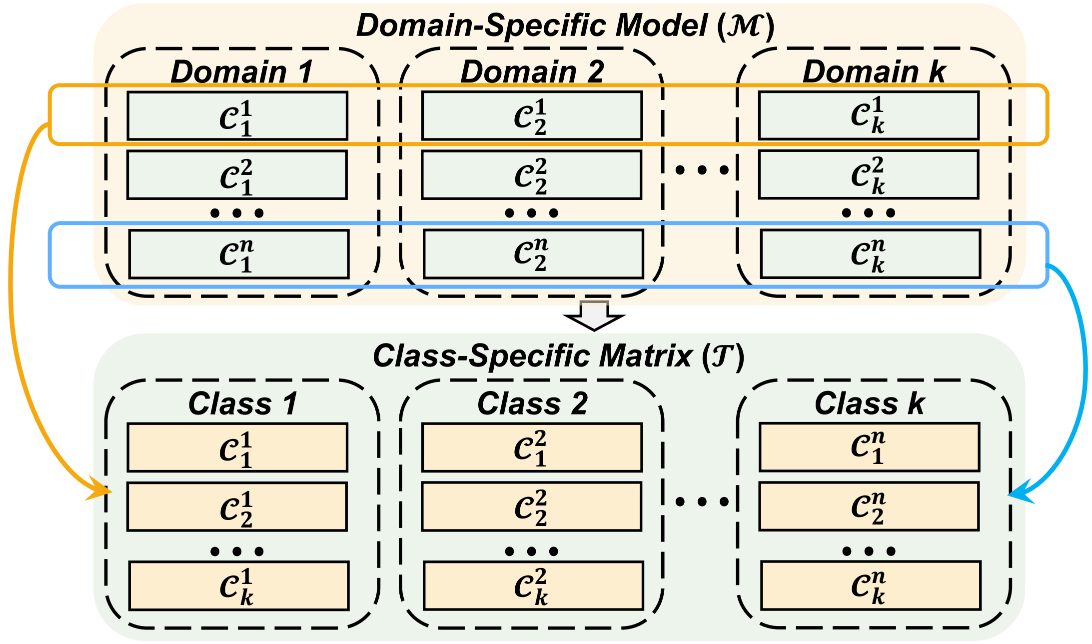


Domain Generalization (DG)

- ***Class-Specific Aggregation:*** extracting class hypervectors from each domain-specific model
- ***Domain-Invariant Filter:*** identifying and regenerating domain-invariant dimensions
- ***Model Ensemble:*** combining multiple domain-specific models into a single domain-invariant model

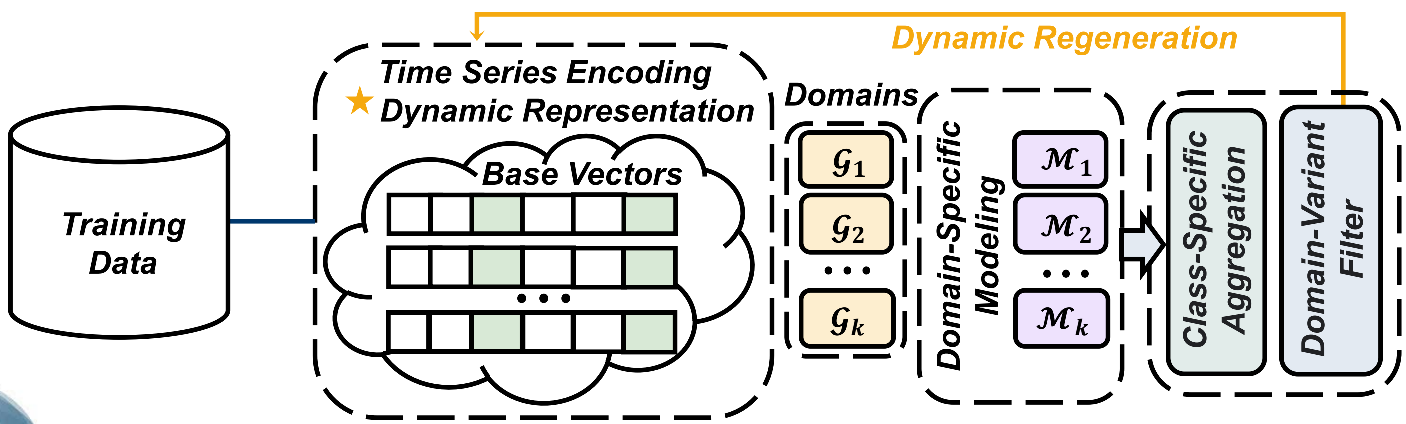
DG: Class-Specific Aggregation

- For each class i: extract the class hypervector from every domain-specific model representing class i to form a class-specific matrix.



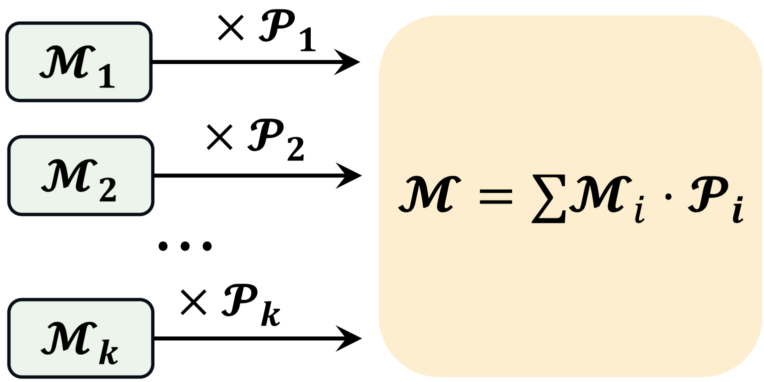
DG: Domain-Variant Filter

- For each class-specific matrix, we calculate the variance of each dimension to measure its dispersion.
- Dimensions with very different values for the same class indicate that they store highly differentiated patterns, and therefore are considered domain-variant.
- We select dimensions with the highest variance and their corresponding projection matrix with a new randomly generated vector.



DG: Model Ensemble

- We assemble all the domain-specific models to construct a general domain-invariant model based on the weight of each domain.
- We then ensemble domain specific models $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_k$ by $\mathcal{M} = \mathcal{P}_1 \cdot \mathcal{M}_1 + \mathcal{P}_2 \cdot \mathcal{M}_2 + \cdots + \mathcal{P}_k \cdot \mathcal{M}_k$.
- For each source domain $\lambda \in [0, k]$, we calculate the proportion of the data from domain λ as $\mathcal{P}_\lambda = \frac{\mathcal{N}_\lambda}{\mathcal{N}_{total}}$, where $\mathcal{N}_\lambda$ denotes the number of sample from domain λ and $\mathcal{N}_{total}$ denotes the total number of samples.



Evaluations

[1] Huang, Zeyi, et al. "Self-challenging improves cross-domain generalization.", ECCV 2020.
[2] Parascandolo, Giambattista, et al. "Learning explanations that are hard to vary.", ICLR, 2020.
[3] Hernández-Cano, Alejandro, et al. "Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system.", DATE, 2021.

- Experimental Setup
  - Server CPU
  - Embedded CPU
  - Embedded GPU
- Baselines
  - CNN-based domain-generalization algorithms: Representation Self-Challenging (RSC) [1] and AND-mask [2]
  - SOTA HDC with static encoder not considering domain generalization, denoted as BaselineHD [3]
    i. Physical Dimensionality ($\mathcal{D} = 0.5k$): a compressed dimensionality designed for resource-constrained platforms
    ii. Effective Dimensionality ($\mathcal{D}^* = 4k$): physical dimensionality + all the regenerated dimensions during the retraining iterations

Evaluations (Cont.)

- Taking human activity recognition as the application use case, we evaluate DOMINO on widely-used multi-sensor time series datasets:
  - DSADS
  - USC-HAD
  - PAMAP2

** Please refer to our paper for detailed data preprocessing steps.*

[1] Barshan, Billur, and Murat Cihan Yüksek. "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units.", The Computer Journal, 2014.
[2] Zhang, Mi, and Alexander A. Sawchuk. "USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors.", ACM conference on ubiquitous computing. 2012.
[3] Reiss, Attila, and Didier Stricker. "Introducing a new benchmarked dataset for activity monitoring." 2012 16th international symposium on wearable computers. IEEE, 2012.

Accuracy: LODO Accuracy

- ***Compared to SOTA CNNs:***

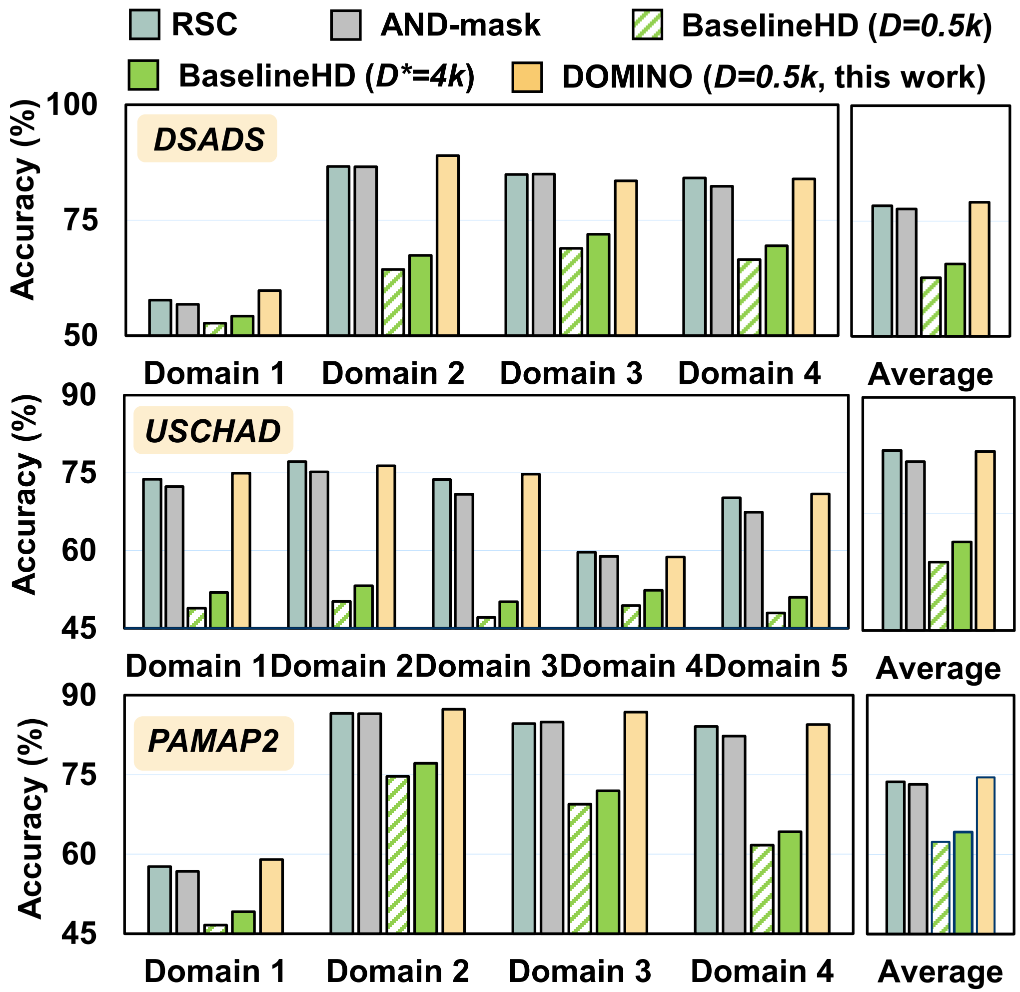
✓ 0.96% higher than RSC

✓ 2.04% higher than AND-mask

- ***Compared to BaselineHD:***

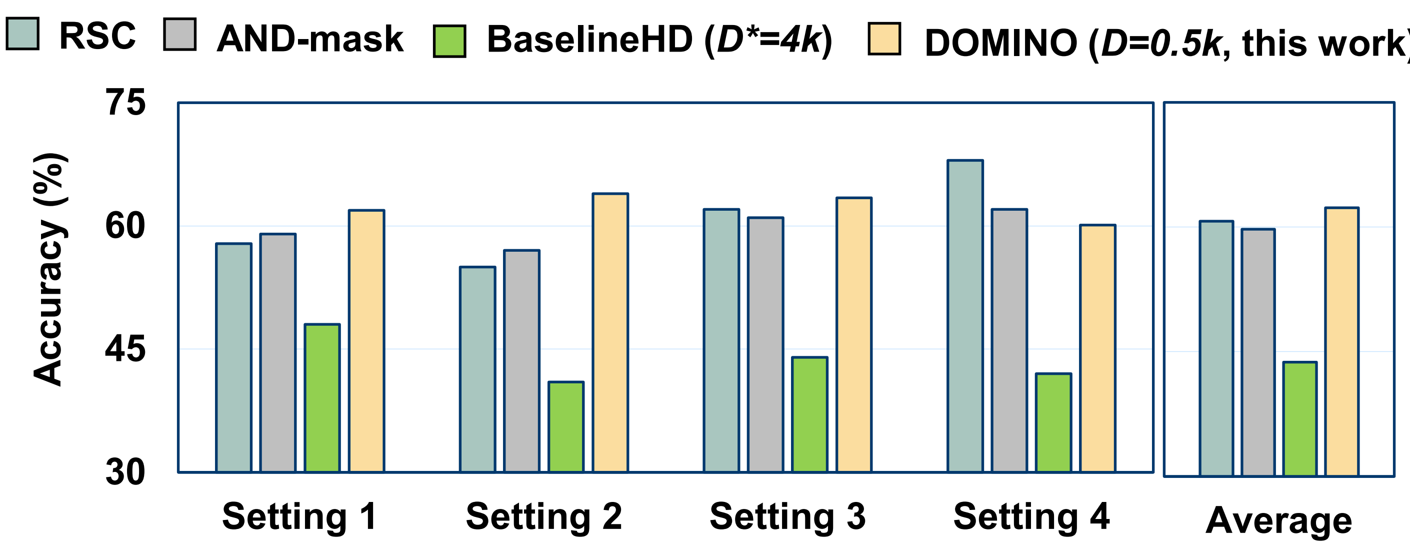
✓ 11.70% higher than BaselineHD ($\mathcal{D}^* = 4k$)

✓ 15.93% higher than BaselineHD ($\mathcal{D} = 0.5k$)

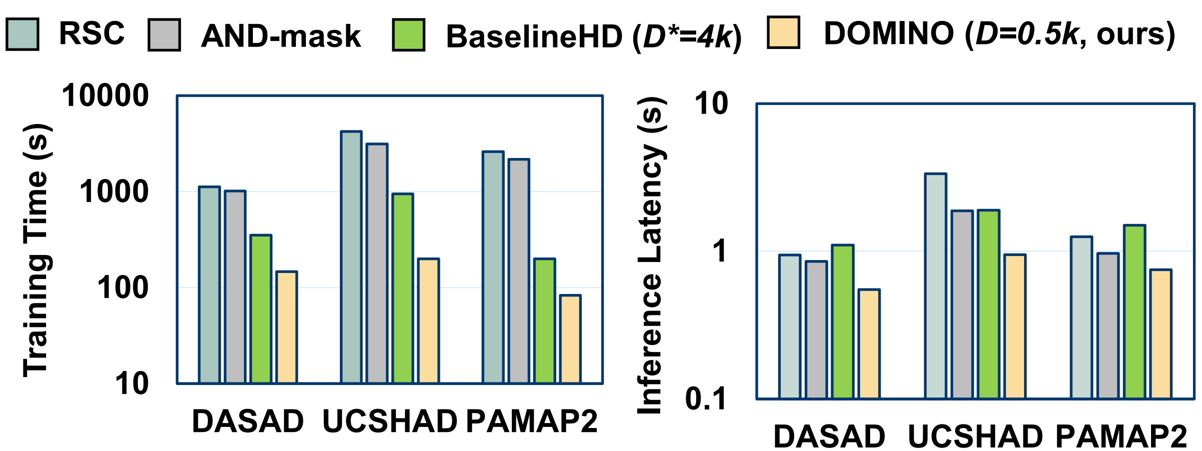


Accuracy: Imbalanced Training Data

- Imbalanced data: certain domains contribute a disproportionately larger portion of the data while other domains are represented by considerably smaller amounts
- DOMINO outperforms SOTA CNN-based DG approaches by exhibiting on average 1.18% and 2.58% higher accuracy than RSC and AND-mask, respectively



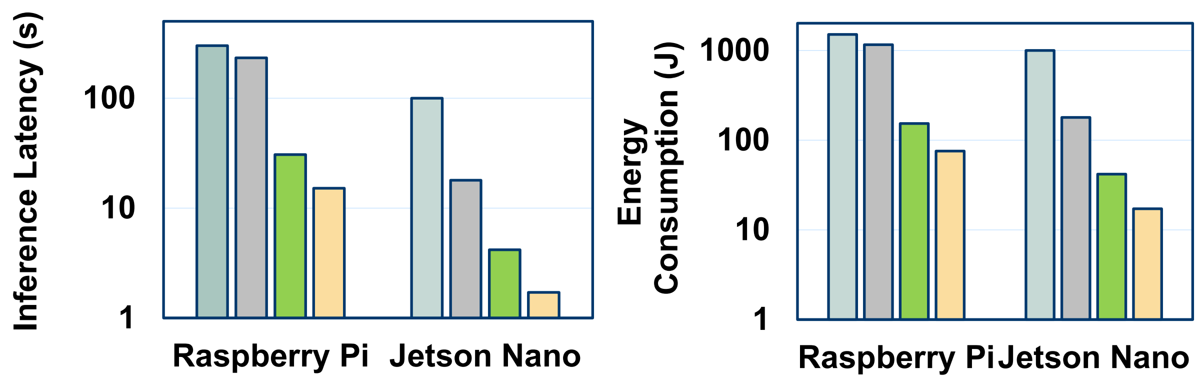
Efficiency



(a) Comparing Efficiency on Server CPU

On Server CPU, DOMINO exhibits:

- 16.34× faster training than RSC
- 14.17× faster training than AND-mask
- 2.89 × faster inference than RSC
- 1.97 × faster inference than AND-mask

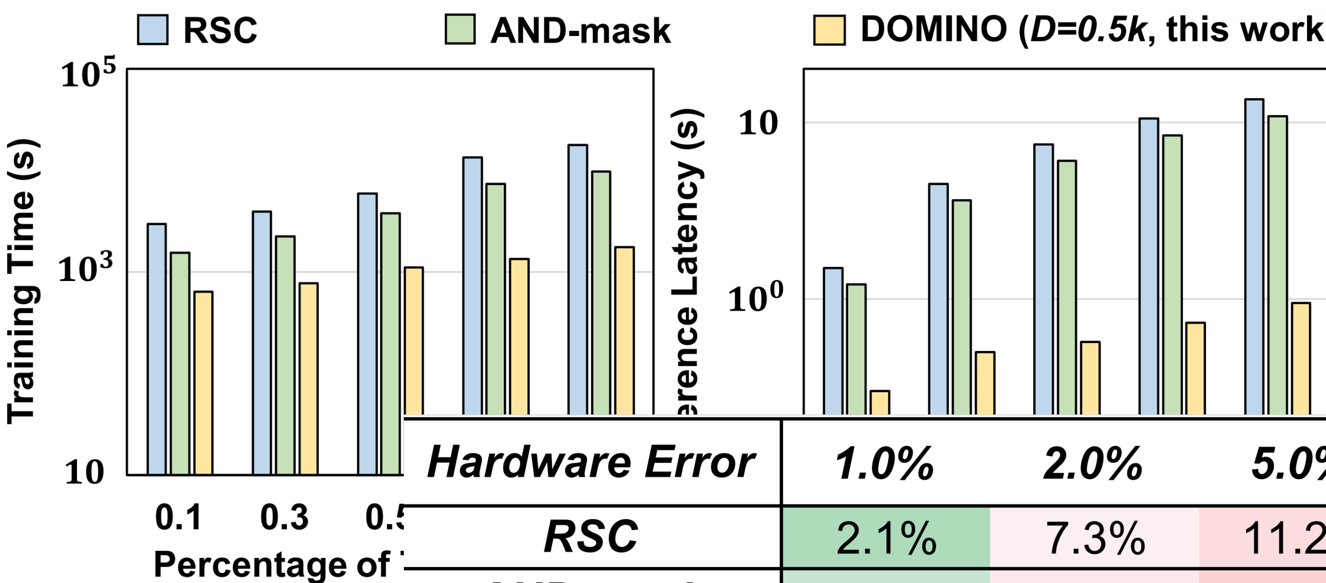


(b) Comparing Efficiency on Embedded Platforms

On embedded devices:

- Raspberry Pi:
  - ✓ 19.79 × faster than RSC
  - ✓ 15.31× faster than AND-mask
- Jetson Nano
  - ✓ 58.44 × faster than RSC
  - ✓ 10.49 × faster than AND-mask

Other Results



Hardware Error		1.0%	2.0%	5.0%	10.0%	15.0%
RSC		2.1%	7.3%	11.2%	20.4%	29.7%
AND-mask		2.9%	8.7%	13.8%	28.1%	35.2%
DOMINO	1 bit	0.0%	0.0%	1.0%	3.1%	4.1%
	2 bits	0.0%	0.5%	1.6%	4.8%	8.0%
	4 bits	0.2%	1.0%	2.9%	7.4%	11.7%
	8 bits	1.4%	3.6%	5.1%	12.8%	17.6%

Thank You For Your Attention.

DOMINO: Domain-Invariant Hyperdimensional Classification for Multi-Sensor Time Series Data

Junyao Wang, Luke Chen, Mohammad Abdullah Al Faruque
University of California, Irvine

IEEE/ACM ICCAD
2023 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN
42nd Edition