Deterministic Hyperdimensional Learning with Rank Refinement (Student Abstract)

Abu Kaisar Mohammad Masum, Sercan Aygun
School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, 70503, USA
c00591145@louisiana.edu, sercan.aygun@louisiana.edu

Abstract

Hyperdimensional Computing (HDC) represents data as high-dimensional hypervectors that are robust and efficient for learning. Existing methods often rely on pseudo-random hypervector generation, which can suffer from poor orthogonality and high variance across runs, ultimately slowing convergence. These approaches typically require numerous iterations (20–100) to achieve acceptable accuracy. We propose a method that utilizes deterministic Sobol-based linear projections and rank-based retraining to construct more stable and discriminative hypervectors, thereby reducing class confusion. Unlike pseudo-random initialization, our projections guarantee reproducibility and better coverage of the feature space. As a result, our approach achieves up to 97% accuracy in only 5 iterations. This makes our model up to 20× faster while simultaneously improving accuracy.

Introduction

HDC encodes data into high-dimensional hypervectors ($\mathcal{HV}$s) that must be nearly orthogonal to ensure robustness and discriminability (Kanerva 1988). Conventional HDC relies on `pseudo-random` $\mathcal{HV}$s, which provide only approximate orthogonality and often require many iterations to converge. Sobol sequence (Joe and Kuo 2008) `quasi-random` methods improve fast-convergent space-filling properties, yet still depend on stochastic initialization.

In this work, we introduce a deterministic Sobol-based linear projection that generates well-structured and reproducible hypervectors, ensuring stable coverage of the feature space. To further reduce class confusion, we propose a rank-based retraining strategy, where class prototypes are iteratively refined by enforcing a margin between the correct and most competitive incorrect classes. This joint design enables our model to achieve higher accuracy with up to 20× faster convergence compared to existing HDC approaches (Aygun and Najafi 2024).

Method

Figure 1 illustrates the overall workflow of our proposed method, showing the transformation from raw input vectors to high-dimensional $\mathcal{HV}$s, class prototype construction, and refinement through rank-based retraining.

Linear Mapping

To project an input vector $\mathbf{x} \in \mathbb{R}^L$ into a high dimensional space $\mathbb{R}^D$, we construct two projection matrices, a position matrix $\mathbf{P} \in \mathbb{R}^{D \times L}$ and a value matrix $\mathbf{V} \in \mathbb{R}^{D \times L}$, both generated using Sobol sequences. Sobol provides low-discrepancy quasi-random vectors with improved space-filling properties, and linear projections are chosen over non-linear mappings to ensure scalability and faster convergence. For each dimension $i \in \{1, \ldots, D\}$, we compute linear projections $p_i = \mathbf{p}_i^\top \mathbf{x},\ v_i = \mathbf{v}_i^\top \mathbf{x}$, where $\mathbf{p}_i$ and $\mathbf{v}_i$ are the ith rows of $\mathbf{P}$ and $\mathbf{V}$.

Feature Binding and Normalization

The projection outputs are fused elementwise to produce a hypervector $\mathbf{h} \in \mathbb{R}^D$. We normalize the input to $[-1, 1]$ via $\tilde{x} = 2\mathbf{x} - 1$ and compute:

$$h_i = \left(\frac{1}{L} \sum_{j=1}^{L} P_{ij} \tilde{x}_j \right) \cdot \left(\frac{1}{L} \sum_{j=1}^{L} V_{ij} \tilde{x}_j \right) \tag{1}$$

Finally, the hypervector is L_2-normalized $\mathbf{h} \leftarrow \frac{\mathbf{h}}{\|\mathbf{h}\|_2}$.

Class Prototypes and Similarity Classification

Each class $c \in \{1, \ldots, C\}$ is represented by a prototype hypervector $\mathbf{H}_c \in \mathbb{R}^D$. These are obtained by summing all training hypervectors with label $y_i = c$ and normalizing $\mathbf{H}_c = \frac{1}{N_c} \sum_{i:y_i=c} \mathbf{h}_i$, $\mathbf{H}_c \leftarrow \frac{\mathbf{H}_c}{\|\mathbf{H}_c\|_2}$. For a test hypervector $\mathbf{h}$, classification is performed by nearest-prototype search, $\hat{y} = \arg\max_{c \in \{1,\ldots,C\}} \cos(\mathbf{h}, \mathbf{H}_c)$. Since all vectors are unit-normalized, this reduces to $\hat{y} = \arg\max_c \mathbf{h}^\top \mathbf{H}_c$.

Margin-Based Rank Loss Retraining

To refine class prototypes and reduce misclassification, we adopt a margin-based rank loss. For each sample $\mathbf{h}_i$ with true class y_i, let $\hat{y}_i$ be the most similar incorrect class $\hat{y}_i = \arg\max_{c \neq y_i} \cos(\mathbf{h}_i, \mathbf{H}_c)$. The loss function is defined as:

$$\mathcal{L}_i = \max\Big(0, m - \big(\cos(\mathbf{h}_i, \mathbf{H}_{y_i}) - \cos(\mathbf{h}_i, \mathbf{H}_{\hat{y}_i})\big)\Big) \tag{2}$$

where $m > 0$ is a margin hyperparameter. Gradient updates adjust $\mathbf{H}_{y_i}$ closer and $\mathbf{H}_{\hat{y}_i}$ farther from $\mathbf{h}_i$, followed by normalization.

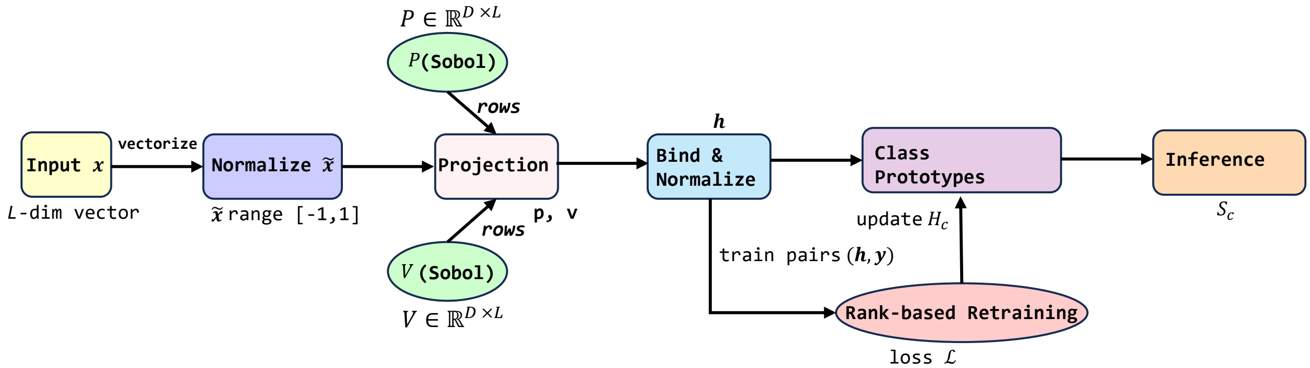

Figure 1: Flow of the proposed method: input $\mathbf{x}$ is normalized, projected via Sobol matrices $\mathbf{P}$, $\mathbf{V}$, bound into hypervectors $\mathbf{h}$, superposed into prototypes $\mathbf{H}_c$, and classified by cosine similarity. Prototypes are refined through rank-based retraining with margin loss $\mathcal{L}$.

Method	MNIST	F-MNIST	UCIHAR	ISOLET	Iters	Speedup
HDC	89.0	N/A	95.7	91.5	10	2×
RFF HDC-1bit	95.4	84.0	95.7	94.4	10	2×
RFF G(2^3)-VSA	95.7	86.7	95.6	94.4	10	2×
RFF G(2^4)-VSA	96.6	86.5	96.6	96.0	10	2×
LeHDC	94.7	87.1	95.2	94.9	100	20×
OnlineHD	96.0	N/A	95.0	96.0	8	1.6×
QuantHD (Binary)	N/A	N/A	96.5	94.6	30	6×
HyDREA	N/A	N/A	96.5	93.0	100	20×
OpenHD	N/A	N/A	96.4	95.2	20	4×
Proposed (Ours)	96.0	86.0	97.0	94.2	**5**	—

Table 1: State-of-the-art comparison on MNIST, Fashion-MNIST, UCIHAR, and ISOLET datasets. *HDC* (Imani et al. 2019b), *RFF-HDC-1bit* and *RFF-VSA* (Yu et al. 2022), *LeHDC* (Duan et al. 2022), *OnlineHD* (Hernández-Cano et al. 2021), *QuantHD* (Imani et al. 2019a), *HyDREA* (Morris et al. 2021), *OpenHD* (Kang et al. 2022).

Results

We evaluate our HDC framework on several benchmark datasets. The $\mathcal{HV}$ dimension is set to $D = 10{,}000$. Retraining is performed for up to 5 iterations. All experiments run in `PyTorch` on an `NVIDIA RTX 3050` GPU.

The evaluation covered four benchmark datasets: MNIST and Fashion-MNIST for image classification, and UCIHAR and ISOLET for sensor-based activity recognition and speech recognition, respectively. For MNIST and Fashion-MNIST, grayscale images were flattened to 784-dimensional feature vectors. UCIHAR included 561 features extracted from smartphone sensors, while ISOLET consisted of 617 acoustic features per sample. Table 1 summarizes the results across datasets, including accuracy and the number of training iterations. Our proposed method achieves competitive accuracy with only 5 iterations, compared to 8–100 iterations required by state-of-the-art baselines. On UCIHAR, it reaches 97.0% accuracy, surpassing all existing methods while offering up to 20× iteration efficiency improvement. On MNIST and ISOLET, it achieves 96% and 94.2% accuracy, remaining within 1–2% of the baseline accuracies of other architectures but at only a fraction of their iteration cost.

Acknowledgments

This work is supported by National Science Foundation Grants No. 2019511 & 2339701, NASA grant 80NSSC25C0335, and gifts from NVIDIA and NASA.

References

Aygun, S.; and Najafi, M. H. 2024. Sobol sequence optimization for hardware-efficient vector symbolic architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Duan, S.; Liu, Y.; Ren, S.; and Xu, X. 2022. Lehdc: Learning-based hyperdimensional computing classifier. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 1111–1116.

Hernández-Cano, A.; Matsumoto, N.; Ping, E.; and Imani, M. 2021. Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 56–61. IEEE.

Imani, M.; Bosch, S.; Datta, S.; Ramakrishna, S.; Salamat, S.; Rabaey, J. M.; and Rosing, T. 2019a. Quanthd: A quantization framework for hyperdimensional computing. *IEEE*

Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(10): 2268–2278.

Imani, M.; Messerly, J.; Wu, F.; Pi, W.; and Rosing, T. 2019b. A binary learning framework for hyperdimensional computing. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 126–131. IEEE.

Joe, S.; and Kuo, F. Y. 2008. Constructing Sobol sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 30(5): 2635–2654.

Kanerva, P. 1988. *Sparse distributed memory*. MIT press.

Kang, J.; Khaleghi, B.; Rosing, T.; and Kim, Y. 2022. Openhd: A gpu-powered framework for hyperdimensional computing. *IEEE Transactions on Computers*, 71(11): 2753–2765.

Morris, J.; Ergun, K.; Khaleghi, B.; Imani, M.; Aksanli, B.; and Rosing, T. 2021. Hydrea: Towards more robust and efficient machine learning systems with hyperdimensional computing. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 723–728. IEEE.

Yu, T.; Zhang, Y.; Zhang, Z.; and De Sa, C. M. 2022. Understanding hyperdimensional computing for parallel single-pass learning. *Advances in neural information processing systems*, 35: 1157–1169.