CryHD: Accurate, Real-Time, and Energy-Efficient Infant Cry Classification using Hyperdimensional Learning

Ananthasai Subramanian[§*], Mehran Moghadam[§*], Soheil Khooyooz[†], Emilien J Meyer[‡],
M. Hassan Najafi[§], Sercan Aygun[‡], Nima TaheriNejad[†]
[§]Case Western Reserve University, Cleveland, OH, USA
[†]Heidelberg University, Heidelberg, Germany
[‡]University of Louisiana at Lafayette, Lafayette, LA, USA
{axs2700, moghadam, najafi}@case.edu, {soheil.khooyooz, nima.taherinejad}@ziti.uni-heidelberg.de, {c00439243, sercan.aygun}@louisiana.edu (*Equally contributed)

***Abstract*—Taking care of newborn infants has always been a challenging task for first-time parents, especially when the infants start crying. Infant crying sounds are a newborn's primary means of communicating physical and emotional needs. Experts and parenting studies have identified a set of universal infant sounds, also known as the Dunstan Baby Language (DBL), that help parents decode what their baby is trying to express. Automatic classification of these sounds offers a practical aid for caregivers, yet prevailing approaches rely on deep Convolutional Neural Networks (CNNs) that, while accurate, incur substantial memory, latency, and energy costs and are sensitive to the bit-level faults that arise under the low-power operation of wearable devices. In this work, we present CryHD, a fast and energy-efficient framework for infant cry classification based on Hyperdimensional Computing (HDC), which encodes acoustic features into high-dimensional vectors and classifies them through lightweight, highly parallel, and error-tolerant operations. Evaluated on the DBL dataset across its five cry types, CryHD attains a competitive accuracy of 89.17% while reducing memory usage by 1,958×, runtime execution by 2.5×, and energy consumption by 3× relative to a state-of-the-art CNN baseline. Owing to the intrinsic robustness of HDC, CryHD further retains 82.80% inference accuracy under a 20% bit-error rate from soft errors, where the CNN baseline drops to 63.25%. These results establish hyperdimensional learning as a viable and efficient alternative to deep neural networks for infant cry classification.**

***Index Terms*—Dunstan Baby Language, energy-efficient machine learning, hyperdimensional computing, infant cry classification.**

I. Introduction

Newborn infants convey their needs through crying. Pediatricians, caregivers, nurses, and experienced parents can often interpret an infant's cry based on prior experience. First-time parents, however, frequently struggle with their infant's crying, as the different cries sound nearly identical to them. This leaves many parents frustrated and unable to calm their newborns. One of the major concerns of first-time parents is therefore to recognize what their newborn needs, and the most direct way to do so is to decode the cry sound and respond promptly. Acquiring this perceptual skill, however, is far more demanding for a human than learning it through artificial intelligence (AI) and machine learning (ML) algorithms [1], [2].

Recent progress in audio and speech ML has made automatic infant cry classification a practical supportive resource for caregivers [3]. A widely used reference for this task is the Dunstan Baby Language (DBL), a theory that identifies five major canonical cry types, namely *eair*, *eh*, *heh*, *neh*, and *owh*, corresponding to belly pain, the need to burp, discomfort, hunger, and tiredness, respectively [4]. Owing to its well-defined labels, the DBL corpus has become a standard benchmark for multi-class infant cry classification [3]. For acoustic representation, Mel-Frequency Cepstral Coefficients (MFCCs) and time-frequency images such as spectrograms and Mel-spectrograms are the most common feature choices, the latter typically paired with Convolutional Neural Networks (CNNs) [3], [5], [6]. A substantial body of work has pursued ever-higher classification accuracy on DBL, predominantly through deep and increasingly complex models. Junaidi [7] reports 77.3% accuracy with a Mel-spectrogram and ResNet-152V2 pipeline. Abbaskhah et al. [5] obtain 92.1% with a CNN operating on MFCC and energy features, and Qiu et al. [8] report 92.9% using a hybrid-feature ResLSTM model under cross-validation. More recently, CryHop [4] demonstrated that careful preprocessing can substantially improve reliability on DBL: by computing Mel-spectrograms with a variable hop length, the authors remove the audio-length bias that had inflated earlier results, verify through Gradient-weighted Class Activation Mapping that the model attends to informative spectral patterns rather than clip duration, and combine transfer learning with ensemble averaging to achieve 94.88% top-1 and 99.49% top-2 accuracy.

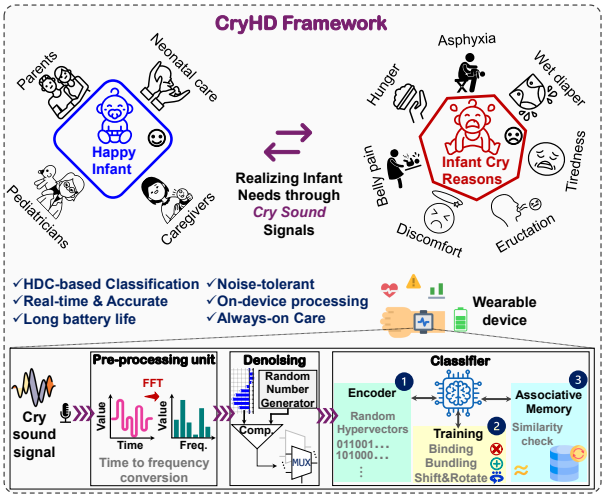


Fig. 1: Overview of CryHD framework.

A practical realization of such an aid is an application that runs continuously on a wearable device, such as a smartwatch, alerting the caregiver and indicating the likely cause of the cry in real time. Wearables, however, operate under tight energy, memory, and compute budgets and must remain responsive while preserving battery life, which places the efficiency of the classifier at the center of the design. Viewed from this perspective, the dominant CNN-based solutions share two limitations. First, deep networks carry large parameter counts and rely on computationally expensive arithmetic, resulting in considerable memory footprints, inference latency, and energy consumption, all of which are difficult to accommodate within the budget of a battery-powered wearable. Second, the techniques used to extend battery life, such as voltage scaling and low-power memories, increase the incidence of soft errors, that is, transient bit-level faults in stored parameters and computation. Because the predictions of conventional neural networks depend on precise weight values, they are sensitive to such corruption, making fault tolerance a desirable property for dependable on-device operation. Recent surveys identify computational efficiency and low latency as prerequisites for moving infant cry analysis from the laboratory into practical systems [3]. The reliability of such systems under the hardware faults induced by low-power operation, however, has received little attention, as has overall efficiency relative to accuracy.

In this work, we address this gap with Hyperdimensional Computing (HDC), a brain-inspired learning paradigm in which information

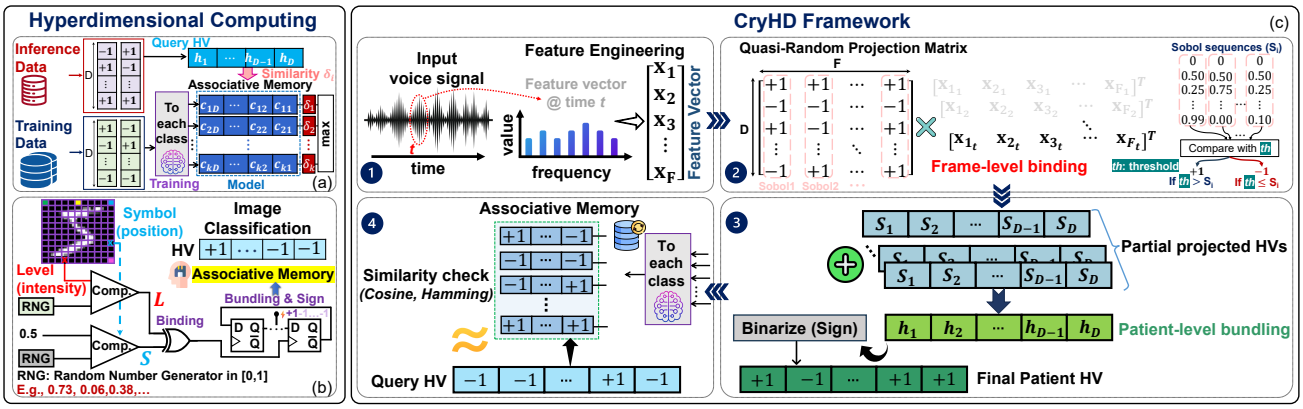


Fig. 2: (a) Overview of an HDC framework, (b) Record-based encoding [9] in image classification tasks, and (c) Underlying pipeline flow in **CryHD**.

is represented by high-dimensional vectors, termed hypervectors (HVs), and manipulated through a small set of simple and highly parallel operations such as binding, bundling, and permutation [9]–[11]. HDC offers single-pass training, an inexpensive inference path dominated by elementary vector operations, and, owing to its distributed and redundant representation, strong intrinsic tolerance to noise and bit-level errors [12], [13]. These properties make HDC a natural candidate for fast, accurate, energy-efficient, and noise-robust classification, yet, to the best of our knowledge, it has not previously been applied to infant cry classification. We therefore introduce **CryHD**, an HDC-based framework for infant cry classification that brings these advantages to this task. Fig. 1 illustrates the overall **CryHD** pipeline, which replaces computationally expensive deep learning layers with lightweight, parallelizable hyperdimensional operations that fit within the strict power, memory, and compute budgets of a wearable device. Fig. 2(a) shows an overview of a standard HDC framework, which maps input data into the hyperdimensional space during training to build a model and performs inference by comparing query HVs against the stored class HVs via associative memory and similarity checks. To represent continuous values in this domain, such as image pixel intensities or continuous acoustic features, HDC commonly utilizes a *record-based* encoding scheme (Fig. 2(b)) that binds structural positions to their corresponding intensity levels. The main contributions of this paper are as follows:

➢ Proposing, to the best of our knowledge, the first HDC framework for infant cry classification, combining a Sobol-based projection record encoding with an N=1 bag-of-frames aggregation that captures cry dynamics through MFCC derivatives.

➢ Validating that **CryHD** reaches accuracy competitive with a CNN baseline on the five DBL classes, while remaining far more stable than the CNN under low-precision quantization.

➢ Demonstrating that, owing to the distributed and redundant nature of its HV representation, **CryHD** sustains high classification accuracy under soft errors, modeled as random bit flips in the stored model, conditions under which the CNN baseline degrades markedly.

➢ Providing a direct, quantitative comparison between the HDC-based and CNN-based approaches on a common dataset and feature pipeline, offering a fair assessment of the accuracy-cost trade-off.

➢ Showing, through 45 nm synthesis and software profiling, that **CryHD** reduces energy by up to 12×, inference latency by 637×, and model memory by up to 1,958× relative to the CNN baselines, which positions it for next-generation wearable platforms.

II. Proposed Approach

Fig. 2(c) details the **CryHD** pipeline, which builds on the HDC principles described above. The system first converts the raw audio signal into a dense, frequency-domain time series matrix. A sliding window is applied, and 40 MFCCs are extracted per frame to capture the short-term power spectrum [14]. To account for local temporal dynamics without resource-intensive sequence tracking on the device, the framework also computes the first- and second-order temporal derivatives of the MFCCs and concatenates them with the original coefficients. This yields a composite feature vector $\mathbf{x}_t \in \mathbb{R}^F$ at each time step t, so that an audio input spanning T frames produces the sequence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$ (Fig. 2(c) ❶).

To map these continuous acoustic features into hyperdimensional space, **CryHD** uses a highly efficient projection-based record encoding scheme. Let $\mathbf{P}_i \in \{-1, 1\}^D$ be a static, quasi-orthogonal ID HV assigned to the i-th spectral feature (position HV) [15]. The projection matrix $\mathbf{P}$ is sampled using Sobol sequences to guarantee a high-quality, low-discrepancy distribution of these base vectors [16], [17]. The instantaneous spatial HV $\mathbf{S}_t \in \{-1, 1\}^D$ is then constructed by scaling the continuous feature values $x_{t,i}$ with their corresponding ID HVs via scalar multiplication (Fig. 2(c) ❷). These are subsequently bundled (and binarized) across all $\mathcal{F}$ features to yield $\mathbf{S}_t = \text{sign}(\sum_{i=1}^{\mathcal{F}} x_{t,i}\mathbf{P}_i)$ (Fig. 2(c) ❸). By leveraging dense vector-matrix multiplication, this approach executes the ID binding and bundling in one step. This formulation effectively bypasses the heavy computational burden associated with generating dynamic level-HVs for continuous data, making it exceptionally well-suited for low-latency edge inference.

Traditional HDC architectures typically utilize *N-gram* binding to encode temporal dependencies and preserve the exact sequence of events. In this approach, consecutive spatial vectors are cyclically shifted and bound together (e.g., using circular convolution or permuted Hadamard products) to represent strict temporal order [18]. In **CryHD**, however, the acoustic front-end already captures temporal kinematics via the first- and second-order derivative coefficients, so applying an explicit N-gram binding would introduce redundant computation and risk saturating the capacity of the D-dimensional vector space. Consequently, we adopt an N=1 "Bag-of-Frames" temporal aggregation strategy. The continuous sequence of spatial vectors is linearly accumulated and thresholded to produce a single, global HV for the audio clip as $H_P = sign(\sum_{t=1}^{T} S_t)$. During the training phase, a canonical class model $M_c \in \{-1, 1\}^D$ is constructed for each infant cry reason by bundling the global HVs of all samples j belonging to class c, demonstrated as $M_c = sign(\sum_{j \in \mathcal{C}_c} H_P^{(j)})$. At inference, a query vector H_Q is produced for an unlabeled cry through the same pipeline. Classification is executed via an associative memory search that returns the most similar class model as $\hat{y} = arg\ max_c(H_Q \cdot M_c)$ (Fig. 2(c) ❹).

III. Evaluation Results

To evaluate the performance and cost-efficiency of the proposed **CryHD** framework, we compared its classification accuracy, hardware robustness, and implementation costs against a CNN baseline. The

TABLE I: ACCURACY (%) COMPARISON OF DIFFERENT APPROACHES

Method	Configuration	Dimension (D)	Dataset1		Dataset2	
			Top-1	Top-2	Top-1	Top-2
CryHD	Sobol w/ th=0.3 (Fig. 2(c) ❷)	128	80.25	87.90	50.85	77.97
		256	80.25	88.54	52.54	83.05
		512	82.17	88.54	52.54	83.05
		1024	82.80	88.54	54.24	83.05
		2048	82.80	88.54	52.54	83.05
		4096	83.44	91.08	50.85	84.75
		8192	83.44	91.72	52.54	84.75
	Sobol w/ th=0.5 (Fig. 2(c) ❷)	128	82.80	90.45	57.62	81.36
		256	84.08	90.45	59.32	84.75
		512	86.62	91.72	62.71	86.44
		1024	86.62	92.99	64.41	89.83
		2048	86.62	93.63	66.10	91.53
		4096	87.90	93.63	71.19	93.22
		8192	88.54	94.27	72.89	94.92
	Sobol w/ th=0.6 (Fig. 2(c) ❷)	128	85.99	91.72	55.93	79.66
		256	86.62	92.36	59.32	84.75
		512	87.26	92.99	61.02	86.44
		1024	87.26	93.63	66.10	93.22
		2048	87.26	93.63	71.19	93.22
		4096	87.26	94.27	74.58	94.92
		8192	89.17	94.27	81.36	96.61
CNN Baseline	FP32	N/A	89.81	94.90	98.31	100.00
	8-bit	N/A	84.71	89.81	96.61	100.00
	7-bit	N/A	84.71	89.17	93.22	100.00
	6-bit	N/A	82.80	90.45	33.90	55.93
	5-bit	N/A	80.25	89.81	30.34	45.79
	4-bit	N/A	73.25	82.17	23.73	45.76

software models were trained and evaluated on two separate infant cry datasets, one obtained from [19] and one from the prior study in [4], hereafter referred to as Dataset1 and Dataset2, respectively.

A. Accuracy Comparison

The CNN baseline model comprises three convolutional layers and a final dense classification layer. The three convolutional layers use 16, 32, and 64 filters, respectively, each with ReLU activation, batch normalization, and max pooling. The classifier then applies global average pooling, flattening, and a five-unit dense output layer that produces the class scores. The network takes the 2D variable-length acoustic frame features as input and was trained for 40 epochs using the Adam optimizer and a cross-entropy loss weighted by inverse class frequency. We evaluated this baseline at full precision (FP32) and at reduced bit-widths of 8, 7, 6, 5, and 4 bits. For **CryHD**, we swept the HV dimension from D=128 to 8,192 and the Sobol encoding threshold (see Fig. 2(c) ❸) over $th \in \{0.3, 0.5, 0.6\}$ to examine how the quasi-random projection affects accuracy. Bit-width for the CNN and dimension D for **CryHD** serve as the respective knobs that trade classification accuracy against hardware cost, allowing us to compare how each design degrades under compression.

Table I reports Top-1 and Top-2 accuracy on both datasets. On Dataset1, **CryHD** achieves 89.17% Top-1 accuracy at D = 8,192, closely matching the FP32 CNN (89.81%) and maintaining strong performance across all dimensions. Its Top-2 accuracy of 94.27% similarly approaches the 94.90% of the FP32 CNN. On Dataset2, the FP32 CNN is stronger in absolute terms (98.31% Top-1), but its accuracy is highly sensitive to quantization: it holds at 8-bit (96.61%) and 7-bit (93.22%) precision yet falls sharply to 33.90% at 6-bit and 23.73% at 4-bit. **CryHD**, by contrast, retains 81.36% Top-1 and 96.61% Top-2 at D=8,192 and degrades gradually as D is reduced. The same pattern appears in milder form on Dataset1, where the CNN declines from 84.71% at 8-bit to 73.25% at 4-bit. Overall, **CryHD** nearly matches the FP32 CNN on Dataset1 and stays far more stable than the low-precision CNN configurations on both datasets. This stability under model compression is the property that matters most for low-power, hardware-constrained deployment.

B. Hardware Cost Comparison

The inference engines for the CNN baseline and proposed **CryHD** were described in Verilog and synthesized using Synopsys Design Compiler v2024.09-SP with a 45nm standard cell library. For an iso-frequency comparison representative of the strict thermal and power constraints of wearable devices, all designs were synthesized

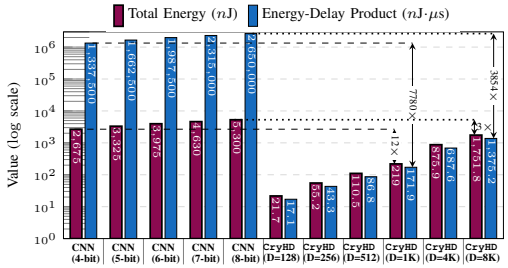


Fig. 3: Hardware execution efficiency comparison.

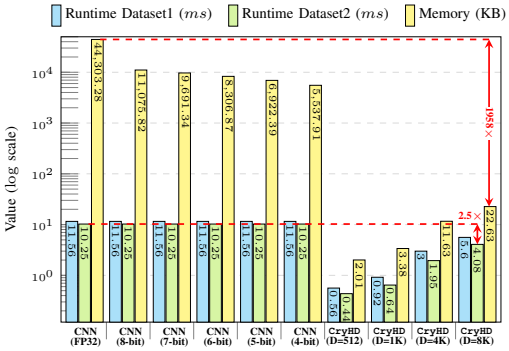


Fig. 4: Runtime and memory footprint comparison.

at a target frequency of 200 MHz (5.0 ns clock period). Fig. 3 illustrates the significant energy consumption savings achieved by the proposed design. The CNN baseline is limited by its serialized, multi-cycle convolutional datapath: even at reduced precision, the 8-bit model needs about 500 μs to process a single acoustic frame. In contrast, the highly parallel datapath of **CryHD** processes all D HV dimensions concurrently. This spatial parallelism allows the framework to map, bind, and bundle an entire audio frame in 157 clock cycles (corresponding to the number of distinct features), reducing the total inference latency to merely 0.785 μs, a 637× reduction relative to the 8-bit CNN.

Furthermore, Fig. 3 compares the energy and Energy-Delay Product (EDP) of the two designs. At 4-bit precision, the multi-layer CNN baseline consumes 2,675 nJ per inference with an EDP of 1,337,500 nJ · μs. At D=1,024, **CryHD** requires only 219.0 nJ per inference, which is a 12× lower energy and a 7,780× lower EDP than the 4-bit CNN, while avoiding the accuracy loss that sub-8-bit quantization causes. At its largest configuration, D=8,192, where accuracy peaks, **CryHD** still consumes only 1,751.8 nJ, a 3× reduction relative to the 8-bit CNN at 5,300 nJ. The EDP gap is wider still because of the latency advantage: at D=4,096, **CryHD** reaches an EDP of 687.6 nJ·μs, 3,854× lower than the 8-bit CNN at 2,650,000 nJ·μs. These results show that the parallel, projection-based HDC datapath removes much of the spatial and temporal overhead of the deep network, which makes **CryHD** suited to the power and thermal limits of always-on wearable devices.

C. Memory Footprint and Runtime

We profiled the software inference runtime and memory footprint of both frameworks on an Intel Core Ultra 7 155H processor with 16 GB of RAM as illustrated in Fig. 4. The two designs differ most in static memory. The CNN baseline requires 44,303.28 KB for the FP32 model and remains large, at 11,075.82 KB, even when quantized to 8-bit. **CryHD** instead stores only the base projection matrix and the canonical class HVs, so even at the largest dimension, D=8,192, the full model occupies 22.63 KB. This is a 489× reduction relative to the 8-bit CNN and a 1,958× reduction relative to the FP32 model. At D=1,024 the footprint falls to 3.38 KB, which fits within the constrained SRAM of edge wearable microcontrollers.

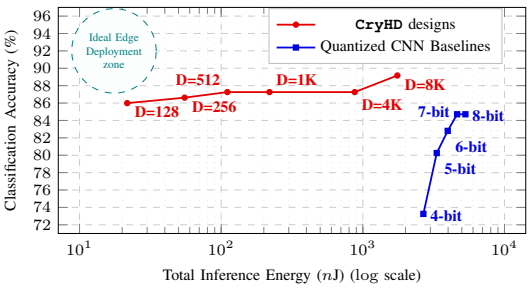


Fig. 5: Energy-Accuracy Pareto frontier evaluating architectural trade-offs (based on Dataset1 and Sobol w/ *th=0.6*—see Table I and Fig. 3).

The runtime follows a similar trend. Because the CNN is dominated by serialized convolutional layers, its software runtime is essentially fixed at 11.56 ms per inference on Dataset1 and 10.25 ms on Dataset2, independent of bit-precision. **CryHD** replaces these layers with parallelizable similarity checks over the associative memory, which lowers the runtime and scales with D. On Dataset1, **CryHD** runs in 5.60 ms at D=8,192 (2.1× faster than the FP32 CNN) and in 0.92 ms at D=1,024 (12.5× faster). The same pattern holds on Dataset2, where D=8,192 runs in 4.08 ms (2.5×) and D=1,024 in 0.64 ms (16×). Together with the hardware energy results, these measurements support hyperdimensional learning as a lightweight alternative for on-device infant cry classification.

D. Energy-Accuracy Pareto Analysis

Fig. 5 places both frameworks on an energy-accuracy plane for Dataset1, where a desirable edge model sits toward the top-left, combining high accuracy with low energy (the "Ideal Edge Deployment" zone). The CNN baseline struggles to approach this ideal zone. At a standard 8-bit precision, the CNN consumes a substantial 5,300 nJ per inference. Attempting to constrain this energy consumption through aggressive network quantization yields diminishing outcomes; scaling down to 4-bit precision reduces the energy footprint to 2,675 nJ, but triggers a catastrophic degradation in classification accuracy, which plummets to 73.25%. The CNN therefore faces a hard floor: further energy savings come only at the cost of accuracy. The proposed **CryHD** lies much closer to the ideal deployment zone. At its maximum dimensionality (D=8,192), **CryHD** achieves a highly competitive 89.17% accuracy while consuming only 1,751.8 nJ. This configuration operates at a lower energy cost than the most aggressively quantized 4-bit CNN, while delivering roughly 16 percentage points higher accuracy. As D is reduced, **CryHD** degrades gracefully: at D=1,024 and D=512 the energy falls to 219.0 nJ and 110.5 nJ, respectively, while accuracy stays above 86%. **CryHD** can thus trade energy for accuracy across a wide range without the sharp accuracy loss seen in the quantized CNN.

E. Soft Errors Analysis

As discussed in Section I, wearable devices reduce power through techniques such as voltage scaling and low-power memories, and these same techniques raise the rate of soft errors, that is, transient bit-level faults in stored parameters and in computation. Because conventional neural networks depend on precise weight values, such faults can corrupt their predictions, whereas a classifier that tolerates them intrinsically can remain reliable without the area and energy overhead of dedicated error-correction circuitry. We evaluated fault tolerance across three operational types in **CryHD** (with D=4,096): transient bit-flips in feature encoding output registers, persistent faults during associative memory accumulation in training, and bit-flips during inference. Fault rates from 1% to 20% were applied over 1,000 Monte Carlo trials. For the quantized CNN baselines (7-bit, 8-bit, and 10-bit), we inject random bit-flips into the individual layers.

The results (mean values) reported in Table II demonstrate that **CryHD** degrades gradually as the fault rate increases, with accuracy declining smoothly across all stages rather than collapsing, evaluating under Top-1 classification. Even at a 20% fault rate during inference, it maintains 82.80% on Dataset1 and 70.14% on Dataset2, while at the training stage it retains 84.08% and 70.85%, respectively. The CNN baselines fall much more steeply: with faults injected into all layers at the same 20% rate, the 7-, 8-, and 10-bit models drop to roughly 63% on Dataset1 and to about 24% on Dataset2, the latter close to random guessing for a five-class problem. The contrast is in how each model declines, since **CryHD** loses accuracy slowly while the CNN loses most of its accuracy on Dataset2.

TABLE II: Reliability Assessment Across Hardware Stages and Network Layers Under Soft Error Rates

Method	Stage	Injected Noise 1% D1	1% D2	2% D1	2% D2	5% D1	5% D2	10% D1	10% D2	20% D1	20% D2
CryHD† *Top-1 Acc.*	Encoding	86.62	73.83	85.99	71.39	84.71	70.14	83.44	67.76	80.89	62.29
	Training	86.62	74.58	85.35	74.58	85.35	72.95	84.71	72.95	84.08	70.85
	Inference	85.35	73.83	84.71	73.25	83.44	73.02	82.80	72.07	82.80	70.14
CNN (7-bit) *Top-1 Acc.*	Layer 1	88.32	65.20	88.11	55.66	83.03	39.27	73.38	30.02	29.27	24.15
	Layer 2	87.68	69.32	85.64	61.71	80.63	38.95	76.73	28.61	73.45	23.93
	Layer 3	88.12	77.80	87.88	77.32	87.24	76.58	85.64	73.15	76.82	50.37
	Final Layer	88.43	77.10	88.32	76.12	88.02	73.25	87.19	70.66	85.35	57.81
	All Layers	86.84	56.90	85.39	44.81	76.27	25.69	65.11	23.86	63.39	23.73
CNN (8-bit) *Top-1 Acc.*	Layer 1	88.32	65.63	87.73	60.17	82.55	43.19	70.71	27.95	35.49	24.08
	Layer 2	87.53	69.64	86.41	61.56	81.94	41.58	77.15	28.03	73.73	23.73
	Layer 3	88.22	79.00	88.04	78.42	87.20	77.39	84.97	73.02	78.80	54.47
	Final Layer	88.41	78.08	88.35	76.47	88.01	72.95	87.60	67.07	85.29	60.37
	All Layers	87.07	57.71	85.45	45.15	69.71	28.92	64.03	23.73	63.25	23.73
CNN (10-bit) *Top-1 Acc.*	Layer 1	88.32	68.19	88.33	62.00	85.65	45.05	72.84	31.46	39.73	24.68
	Layer 2	87.99	70.14	86.29	66.86	82.64	47.73	78.82	29.31	73.97	24.15
	Layer 3	88.21	79.15	88.04	78.22	87.30	77.02	85.65	73.83	81.18	57.49
	Final Layer	88.48	77.14	88.41	76.39	88.08	73.88	87.92	68.92	86.85	62.25
	All Layers	87.54	62.19	86.46	49.88	61.90	27.95	66.68	23.86	63.83	23.73

D1: Dataset1, D2: Dataset2. †: Sobol w/ *th=0.6* is adopted as the HV encoder source.

The layer-wise results in Table II further show that faults in the first two convolutional layers are far more damaging than faults in later layers. At a 20% fault rate, corrupting Layer 1 or Layer 2 alone lowers the 8-bit CNN's Dataset2 accuracy to 24.08% and 23.73%, since errors in the early layers propagate through the rest of the network. Faults confined to Layer 3 or the final layer are better tolerated, leaving 54.47% and 60.37%. This dependence on where the fault lands is characteristic of serialized, layered architectures. **CryHD**, by contrast, spreads information across all D dimensions, so no single location is critical, which is the source of its greater resilience.

IV. Conclusion

This paper presented **CryHD**, a lightweight HDC-based framework for infant cry classification. By replacing dense MAC-based deep learning with high-dimensional algebraic operations over hypervectors, **CryHD** closely matches the full-precision CNN while remaining far more stable than the CNN under low-precision quantization. On synthesized 45 nm hardware, **CryHD** lowers frame-level inference latency by 637×, energy consumption by up to 12×, and the energy-delay product by up to 7,780× relative to the quantized CNN baselines. At its accuracy-maximizing configuration (with an HV size of 8K), **CryHD** consumes 3× less energy than the 8-bit CNN baseline and reduces the model memory footprint by up to 1,958× relative to the FP32 CNN. **CryHD** also degrades gracefully under injected soft errors, retaining usable accuracy at fault rates where the CNN falls close to random, and does so without any dedicated error-correction hardware. These results position **CryHD** as a promising solution for real-time, accurate, and energy-efficient infant cry classification on next-generation wearable platforms.

Acknowledgment

This work was supported in part by the National Science Foundation under Grants No. 2339701, 2609436, and 2616837; NASA under Grant No. 80NSSC25C0335 and a second NASA award; the Vernon & Ruby Langlinais Non-Endowed Research Fund; the Lockheed Martin Corporation Endowed Professorship Fund; the Hector Stiftung; and gifts from NVIDIA and Google.

REFERENCES

[1] C. Ji, T. B. Mudiyanselage, Y. Gao, and Y. Pan, "A review of infant cry analysis and classification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, no. 1, p. 8, 2021. [Online]. Available: https://doi.org/10.1186/s13636-021-00197-5

[2] S. Sirithepmontree, N. Katchamat, and S. Nuampa, "The application of machine learning for infant cries classification and pathological cries detection: A systematic review," *Simulation & Gaming*, 2026, first published online January 14, 2026. [Online]. Available: https://doi.org/10.1177/00368504251410776

[3] S. M. H. Hashemi, H. Kolivand, W. Khan, and T. Saba, "Infant cry analysis: A survey of datasets, features, and machine learning techniques," *IEEE Transactions on Affective Computing*, vol. 17, no. 1, pp. 21–40, 2026.

[4] S. Khooyooz, L. Ricard, J. Liu, M. Haghi, and N. TaheriNejad, "CryHop: Baby cry classification on Dunstan dataset using transfer learning and ensembling," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2026.

[5] A. Abbaskhah, H. Sedighi, and H. Marvi, "Infant cry classification by MFCC feature extraction with MLP and CNN structures," *Biomedical Signal Processing and Control*, vol. 86, p. 105261, 2023.

[6] B. M. Mala and S. Darandale, "Inertia weight-path finder algorithm based feature selection for infant cry classification using crying signals," *Multimedia Tools and Applications*, vol. 85, no. 1, p. 18, 2026. [Online]. Available: https://doi.org/10.1007/s11042-026-21322-6

[7] R. F. Junaidi, "Baby cry sound detection: A comparison of Mel spectrogram image on convolutional neural network models," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 4, pp. 355–369, 2024.

[8] Y. Qiu, X. Yang, S. Yang, Y. Gong, Q. Lv, and B. Yang, "Classification of infant cry based on hybrid audio features and ResLSTM," *Journal of Voice*, 2024.

[9] L. Ge and K. K. Parhi, "Classification Using Hyperdimensional Computing: A Review," *IEEE Circuits and Systems Magazine*, vol. 20, no. 2, pp. 30–47, 2020.

[10] S. Aygun, M. Moghadam, M. H. Najafi, and M. Imani, "Learning from Hypervectors: A Survey on Hyperdimensional Encoding," *ACM Computing Surveys*, pp. 1–35, 2026.

[11] D. Kleyko, D. A. Rachkovskij, E. Osipov, and A. Rahimi, "A survey on hyperdimensional computing aka vector symbolic architectures, part i: Models and data transformations," *ACM Comput. Surv.*, vol. 55, no. 6, Dec. 2022. [Online]. Available: https://doi.org/10.1145/3538531

[12] S. Zhang, K. Juretus, and X. Jiao, "Exploring hyperdimensional computing robustness against hardware errors," *IEEE Transactions on Computers*, vol. 74, no. 6, pp. 1963–1977, 2025.

[13] S. Jeong, H. E. Barkam, S. Yun, Y. Kim, S. Angizi, and M. Imani, "Exploiting boosting in hyperdimensional computing for enhanced reliability in healthcare," in *2025 Design, Automation & Test in Europe Conference (DATE)*, 2025, pp. 1–7.

[14] M. Imani, D. Kong, A. Rahimi, and T. Rosing, "Voicehd: Hyperdimensional computing for efficient speech recognition," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*, 2017, pp. 1–8.

[15] M. Moghadam, A. Masum, S. Aygun, and M. H. Najafi, "Independent and Dynamic Vector Symbolic Architecture for Hardware-Efficient Edge AI," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 34, no. 4, pp. 1110–1123, 2026.

[16] S. Liu and J. Han, "Toward energy-efficient stochastic circuits using parallel sobol sequences," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 7, pp. 1326–1339, 2018.

[17] S. Aygun and M. Hassan Najafi, "Sobol Sequence Optimization for Hardware-Efficient Vector Symbolic Architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 3, pp. 937–950, 2025.

[18] A. Rahimi, P. Kanerva, and J. M. Rabaey, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ser. ISLPED '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 64–69. [Online]. Available: https://doi.org/10.1145/2934583.2934624

[19] M. Ahmed, "Baby cry sense dataset," https://www.kaggle.com/datasets/mennaahmed23/baby-cry-sense-dataset, 2025, accessed: 2026-05-06.