Neuro-Photonix: Enabling Near-Sensor Neuro-Symbolic AI Computing on Silicon Photonics Substrate

Deniz Najafi, *Graduate Student Member, IEEE*, Hamza Errahmouni Barkam, *Student Member, IEEE*, Mehrdad Morsali, *Graduate Student Member, IEEE*, SungHeon Jeong, *Student Member, IEEE*, Tamoghno Das, *Student Member, IEEE*, Arman Roohi, *Senior Member, IEEE*, Mahdi Nikdast, *Senior Member, IEEE*, Mohsen Imani, *Member, IEEE*, and Shaahin Angizi, *Senior Member, IEEE*

***Abstract*—Neuro-symbolic Artificial Intelligence (AI) models, blending neural networks with symbolic AI, have facilitated transparent reasoning and context understanding without the need for explicit rule-based programming. However, implementing such models in the Internet of Things (IoT) sensor nodes presents hurdles due to computational constraints and intricacies. In this work, for the first time, we propose a near-sensor neuro-symbolic AI computing accelerator named Neuro-Photonix for vision applications. Neuro-photonix processes neural dynamic computations on analog data while inherently supporting granularity-controllable convolution operations through the efficient use of photonic devices. Additionally, the creation of an innovative, low-cost ADC that works seamlessly with photonic technology removes the necessity for costly ADCs. Moreover, Neuro-Photonix facilitates the generation of HyperDimensional (HD) vectors for HD-based symbolic AI computing. This approach allows the proposed design to substantially diminish the energy consumption and latency of conversion, transmission, and processing within the established cloud-centric architecture and recently designed accelerators. Our device-to-architecture results show that Neuro-Photonix achieves 30 GOPS/W and reduces power consumption by a factor of 20.8 and 4.1 on average on neural dynamics compared to ASIC baselines and photonic accelerators while preserving accuracy.**

***Index Terms*—Artificial Intelligence, processing-in-sensor, neuro-symbolic AI, photonic accelerator.**

I. Introduction

THE Internet of Things (IoT) market is projected to reach \$1100 billion by 2025, supported by 75 billion connected devices in smart homes and cities. The vast data generated by these pervasive devices offers significant potential for intelligent IoT services, which can revolutionize modern life and drive AI advancements [1]. However, current IoT devices rely heavily on cloud computing due to a lack of autonomous intelligence, leading to significant power demands and under-utilization of data [2], [3], [4]. Therefore, a shift towards a data-centric approach with more intelligent algorithms is imperative to tackle these challenges. This involves enabling IoT nodes to process data locally and generate interpretations for the cloud. In this way, there's a pressing need for transparent and interpretable IoT models to ensure efficiency and safety. Traditional models like Deep Neural Networks (DNNs) show promise but are critiqued for their 'black-box' nature, hindering abstract reasoning and context understanding [5], [6]. Neuro-symbolic AI presents a compelling solution, blending neural network pattern recognition with symbolic AI's logical reasoning capabilities. This fusion aims to provide transparent and logical solutions for a broad range of IoT tasks [7].

Neuro-symbolic AI is particularly valuable for reasoning tasks, enabling capabilities on sensing devices that were not feasible with traditional methods. Besides, such models aim to bolster model robustness and transparency and facilitate complex decision-making near the sensor. A significant challenge, however, lies in optimizing the interaction between hardware and algorithms. Current state-of-the-art solutions often require too much power, are large, or demand excessive computational resources [8], [9], [10], [11]. Overcoming these obstacles is crucial for fully realizing the potential of neuro-symbolic AI,

Received 16 July 2024; revised 14 December 2024; accepted 30 January 2025. Date of publication 4 February 2025; date of current version 4 June 2025. This work was supported in part by the Semiconductor Research Corporation (SRC); in part by the National Science Foundation (NSF) under Grant 2046226, Grant 2006788, Grant 2216772, Grant 2216773, Grant 2247156, Grant 2127780, Grant 2319198, Grant 2321840, and Grant 2312517; in part by DARPA Young Faculty Award, ONR Young Investigator Program Award under Grant N00014-21-1-2225 and Grant N00014-22-1-2067; and in part by the Air Force Office of Scientific Research under Grant FA9550-22-1-0253. The review of this article was arranged by Associate Editor Rabia Yazicigil. *(Corresponding author: Shaahin Angizi.)*

Deniz Najafi, Mehrdad Morsali, and Shaahin Angizi are with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: dn339@njit.edu; mm2772@njit.edu; shaahin.angizi@njit.edu).

Hamza Errahmouni Barkam, SungHeon Jeong, Tamoghno Das, and Mohsen Imani are with the Department of Computer Science, University of California, Irvine, Irvine, CA 92697 USA (e-mail: herrahmo@uci.edu; sungheoj@uci.edu; tamoghnd@uci.edu; m.imani@uci.edu).

Arman Roohi is with the Department of Electrical and Computer Engineering, University of Illinois Chicago, Chicago, IL 60607 USA (e-mail: aroohi@uic.edu).

Mahdi Nikdast is with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA (e-mail: mahdi.nikdast@colostate.edu).

Digital Object Identifier 10.1109/TCASAI.2025.3537968

and this paper dives into the latest advancements and challenges in the hardware-algorithm optimization process, focusing on maximizing the benefits of neurosymbolic integration.

On the hardware side, advancements have primarily concentrated on augmenting CMOS image sensors to accelerate the processing of DNN workloads. One strategy involves integrating CMOS image sensors and processors onto a single chip, a concept known as Processing-Near-Sensor (PNS) [12], [13], [14]. Another strategy entails embedding computation units within individual pixels, termed Processing-In-Sensor (PIS) [3], [4], [15], [16]. The PIS framework handles pre-Analog-to-Digital Converter (pre-ADC) data processing before transmission to the on- or off-chip processor. Despite these advancements, particular challenges persist, including the energy consumption associated with ADC, Digital-to-Analog Converters (DAC), and sense amplifiers in PIS, thereby constraining the integration of all DNN layers within the pixel array [2], [17], The primary focus of existing research has been expediting the initial layer of DNNs and offloading the subsequent layers to a digital accelerator, primarily due to the limited resources of PIS systems. Consequently, the enduring issue of power-intensive peripherals and DAC/ADC units exists despite efforts to minimize their energy consumption for sensing and computation purposes [3], [18], [19], [20]. Red-Eye [21] utilizes charge-sharing tunable capacitors to perform convolution operations. While this approach achieves energy savings compared to CPU/GPU implementations at the cost of reduced accuracy, maintaining high computational accuracy significantly increases energy consumption, requiring up to 100× more energy per frame. Macsen [3], a PIS platform, processes the first convolutional layer of Binary Weight Neural Networks (BWNN) using a correlated double sampling procedure, enabling computation speeds of 1000 fps. However, it faces substantial challenges due to its large area overhead and high power consumption, primarily caused by the SRAM-based PIS methodology. Besides, the notable area overhead and power consumption associated with recent PNS/PIS units necessitate additional memory for storing intermediate data [2], [4], [16]. Moreover, the computational speed stemming from electronic systems operating at limited gigahertz frequencies cannot match the high speeds and extensive parallelism characteristic of optical systems with photo-detection rates surpassing 100GHz [20], [22], [23], [24], [25].

In recent years, several designs have focused on utilizing photonics to enhance computational speed. [26] developed a design employing photonics to implement basic logic gates. Additionally, [27] introduced a nanophotonic accelerator for CNN implementations, incorporating shifter and adder components within the CNN. While these designs leverage silicon photonics to improve speed and parallelism, none have effectively implemented the MAC operation using silicon photonics to enhance efficiency in implementing neural networks. Apart from that, few studies have attempted to incorporate neuro-symbolic methods into in/near-sensor computing environments, as detailed by Arrotta et al. in [28]. However, these implementations are predominantly confined to relatively straightforward classification tasks. Conversely, Wan et al. [8] proposed an advancement in accelerating visual reasoning applications through the factorization of holographic perceptual representations. While this approach promises significant acceleration, it necessitates a multi-tiered 3D memory hierarchy, which is currently impractical for in/near-sensor integration. This discrepancy underscores a critical gap in the literature, highlighting the need for more feasible integration strategies that can accommodate complex computational frameworks in constrained environments.

In this work, we propose a sensor-aware neuro-symbolic AI framework that synergistically combines the strengths of deep learning and symbolic AI on a Silicon Photonics substrate. Unlike other traditional neuro-vector-symbolic architectures (NVSA), which depend on conventional electronic hardware and use hybrid analog-digital acceleration to optimize neural network quantization, necessitating expensive ADCs and resulting in high power consumption, the proposed Neuro-Photonix architecture utilizes a silicon photonics substrate. This approach leverages photonic devices for neural dynamic computation, thereby minimizing the need for costly conventional ADCs, significantly reducing energy consumption, and achieving faster processing speeds. The main contributions of the work are as follows. (1) We introduce *Neuro-Photonix*, a cutting-edge optical near-sensor accelerator tailored for Neuro-Symbolic AI. It offers high performance and energy efficiency, capable of handling the complete processing of neural dynamics and symbolic AI hyper-vector generation; (2) We design a neuro-symbolic AI model compatible with the hardware in terms of bit-precision on the input and the weights while maintaining state-of-the-art performance on complex reasoning tasks such as the RAVEN dataset [29]; and (3) We present a comprehensive study of the different hyperparameters of the algorithm and the design space of the hardware and its impact on performance markers such as accuracy, energy consumption, and latency. All these contributions enable the first-of-its-kind near-sensor neuro-symbolic AI system to perform reasoning tasks near the sensor realistically.

The rest of the paper is structured as follows: Section II delves into current advancements in Silicon Photonics acceleration mechanisms and Neuro-Symbolic AI. Section III outlines the proposed Neuro-Photonix's circuit-to-architecture co-design methodology. Section IV details the innovative correlated and photonics-optimized hardware mapping technique for the Multiply-Accumulate (MAC) engine and encoder. Section V describes the bottom-up evaluation framework and presents a thorough analysis of the evaluation results and comparison. Finally, Section VI concludes the paper.

II. Background and Related Work

Silicon Photonics Acceleration. Silicon-photonic-based accelerators, noted for their higher operational bandwidth compared to electronic ones and their resolution of fan-in/fan-out issues, are preferred for improving DNN and machine vision tasks [20], [27], [30], [31]. These devices are categorized into coherent and non-coherent designs. Coherent designs use a single wavelength with the weight and activation parameters encoded in the amplitude, phase, or polarization of the optical

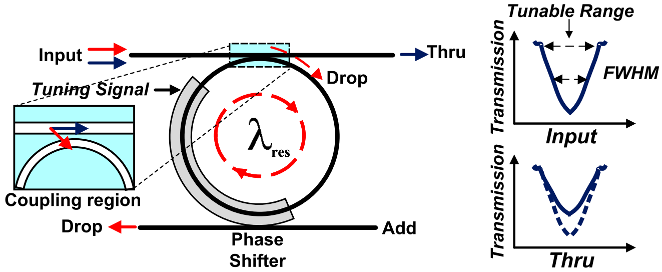


Fig. 1. MR input and through ports' spectra after imprinting a parameter (using tuning signal). By adjusting the MR's resonant wavelength (λ_{res}) using the phase shifter, part of the input signal drops into the ring (through the coupling region) towards the drop port while the remaining propagates towards the through port.

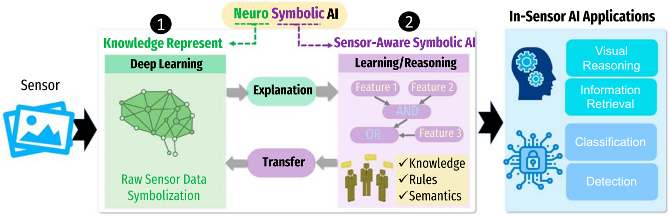


Fig. 2. Neuro-symbolic AI framework consisting of neural dynamic and symbolic AI models. Neural network models are responsible for information embedding, and symbolic models build up knowledge for symbolic decision-making.

signal [32]. Non-coherent designs utilize multiple wavelengths to perform parallel computations, encoding parameters in the signal's amplitude [20], [22]. Micro-Ring resonators (MRs) are employed to manage specific wavelengths and can be dynamically adjusted using mechanisms like microheaters or PIN junctions to selectively interact with designated wavelengths [20], [22].

In non-coherent systems, fine adjustments of MRs, effectively serving as weights, modulate the intensity of incoming light at specific wavelengths, [20], [22]. MRs are essential for encoding weight and enhancing MAC operations by modifying the transmission spectrum of light. This modification is achieved by tuning the MRs' resonant wavelengths to overlap with the input light's wavelength, thus embedding the desired parameters into the transmission spectrum (as shown in Fig. 1). The resonant wavelength is calculated with the formula $\lambda_{res} = \frac{n_{eff} \times L}{m}$, where n_{eff} is the effective refractive index, L the MR's circumference, and m the resonant mode order [33]. MR tuning circuits are employed to regulate the n_{eff} of MR devices, enabling precise control over the output optical signal. These circuits typically use Thermo-Optic (TO) [34] or carrier injection Electro-Optic (EO) tuning [35] methods, both of which modify the n_{eff} and cause a shift in the resonant wavelength ($\Delta\lambda_{res}$). To minimize the power overhead of TO tuning [34], we will employ a technique known as Thermal Eigen Decomposition (TED) [36]. TED enables the collective tuning of all MRs within an MR bank to compensate for significant $\Delta\lambda$MR shifts. This approach effectively mitigates the impact of thermal crosstalk (i.e., undesired phase changes) in MRs while significantly reducing power consumption. Prior research has delved into the enhancement of DNNs by leveraging both coherent and non-coherent photonic technologies. One notable proposal is LightBulb [31], a specialized accelerator for fully binarized Convolutional Neural Networks (CNNs) that employs photonic XNOR operations and popcounts instead of traditional floating-point MAC operations. While LightBulb achieves lower computation latency and requires less memory storage, its design is hampered by increased power consumption due to the extensive use of ADCs. Robin [22] and CrossLight [20] present low-bit-width weight-input CNN accelerators that require adjustment of both activation and weight values within the MRs, focusing solely on convolution layer operations similar to other designs. Another design outlined in [37] offers a CNN accelerator that supports mixed-precision weight-input by employing both wavelength-division multiplexing and time-division multiplexing within a non-coherent silicon photonic framework. Nonetheless, the continued reliance on DACs and ADCs as interlayer conversion mechanisms significantly increases the architectural footprint and power consumption. HolyLight [27], a nanophotonic accelerator, seeks to improve CNN inference throughput using MR-based adders and shifters instead of ADCs. Yet, the extensive application of MRs for encoding both activation and weight parameters leads to increased delays and power usage and diminishes the system's adaptability for various DNN applications.

Neuro-Symbolic AI. In the realm of machine learning for near-sensor IoT, we face unique challenges that differ from traditional edge computing scenarios. Notably, near-sensor AI algorithms are required to effectively process highly noisy analog data. For these algorithms to be viable, they must be highly efficient, memory-centric, robust, and transparent in their computations. These requirements make the deployment of reasoning capabilities directly within sensors particularly demanding. Symbolic AI is recognized for its strong performance in reasoning tasks, robustness against noisy conditions, and ability to make transparent decisions. However, our initial experiments have exposed a critical vulnerability; symbolic AI models lose a significant amount of their robustness when subjected to post-training quantization. This issue makes it challenging to deploy fully trained, advanced models within the limited bit precision environments typical of near-sensor devices. Additionally, symbolic AI struggles to extract information directly from raw sensor data. Given these constraints, there is a clear need for a hybrid approach.

By combining neural networks, which excel at extracting information, with symbolic AI, known for its decision-making strength, we can integrate what is known as Neuro-Symbolic AI into IoT devices. This hybrid approach aims to leverage the strengths of both domains to address the distinctive challenges faced by in-/near-sensor AI. As illustrated in Fig. 2, neuro-symbolic AI algorithms have been introduced as a hybrid solution that combines deep learning capabilities with symbolic AI's transparency. Computationally, the neural dynamic part ❶ can be realized using various DNN models. However, the symbolic AI part ❷ requires programmability to support various learning and reasoning algorithms that can be achieved using HyperDimensional Computing (HDC). HDC has demonstrated proficiency in addressing diverse tasks and datasets, establishing itself as a robust framework well-suited for applications requiring lightweight and highly efficient training [38]. A noteworthy attribute contributing to HDC's popularity is its model weights resilience, specifically the dimensions of its HyperVectors (HVs), as evidenced in prior works [39]. Furthermore, HDC exhibits resilience in handling non-idealities from hardware implementations, as corroborated by recent literature [40].

III. Neuro-Photonix

In this work, we propose the Neuro-Photonix as a high-performance, energy-efficient, and versatile PNS accelerator to support neural symbolic AI at the edge. The distinct hardware needs for neural processing and symbolic reasoning present avenues for tailored optimizations. Neural dynamic components within such models can function efficiently at reduced precision through analog near-sensor computation on a silicon photonics substrate. This enables swift single-cycle MAC operations and exceptionally high energy efficiency while yielding dependable latent representations. Conversely, symbolic reasoning necessitates higher precision to employ symbols and derive precise decisions effectively. In pursuit of achieving an effective neural symbolic AI architecture, we introduce a novel design aimed at enhancing precision through integrating HDC following the neural dynamic Component. Furthermore, according to previous studies, the approach of initially training neural networks and subsequently deriving an HDC model from the trained network led to a significant improvement in the accuracy of the overall design [41].

A. Operational Flow

Fig. 3 shows the high-level operational flow of the proposed Neuro-Photonix architecture. This architecture comprises a m×n sensor array, a novel Low-overhead Modulation Unit (LMU), and a Reconfigurable Optical Core (ROC) as the main components. The operational sequence commences with the global shutter image sensor capturing the input frame (step ❶), followed by processing through the proposed LMU. In step ❷, the LMU transforms the sensor array data into a light beam characterized by varying wavelength (λ) and intensity, which is then transmitted to the ROC via the generated waveguide. Within the LMU, a novel Comparator-based Converter (CBC) is employed to convert the analog data into digital form efficiently and with minimal overhead. The ROC is responsible for running energy-efficient and low-precision dynamic neural networks, followed by converting the outcomes of the neural network into HV. This HV then serves as the foundational element of HDC. To accomplish this, in step ❸, each layer of a DNN is sequentially processed within the ROC. The ROC incorporates a Convolver unit (CU) tasked with handling the various layers of the DNN. The output of CU is then fed back to the LMU in step ❹ as input to execute the next layer in the DNN. This iterative process continues to implement the entire network.In step ❺, the output of the DNN is encoded into an HV using the Hyper-vector Encoder (HVE). This involves reconfiguring the weights of the DNN within the ROC into high-dimensional encoding matrix values. After the creation of the HV, these vectors are transferred to the cloud for further processing (step ❻). The key advantage of the ROC lies in its simplicity in managing weight data, which only requires mapping onto MRs. It is worth noting that, in contrast to previous designs, the activation values are directly modulated into the core's input using Vertical-Cavity Surface Emitting Lasers (VCSEL). This is achieved by adjusting the driving currents of the VCSELs, streamlining the process significantly. It is worth mentioning that certain long-cavity lasers, such as Fabry-Perot [42] lasers, are used for integrating light sources on silicon. However, these lasers typically require high threshold currents, often in the tens of milliamps, leading to significant power consumption and cooling challenges. In contrast, VCSELs overcome these limitations by significantly reducing the lasing threshold current.

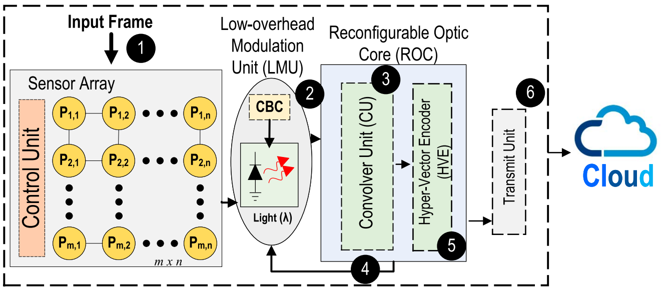


Fig. 3. High-level operational flow of the proposed Neuro-Photonix architecture.

B. Architecture Design

The main architecture of the proposed Neuro-Photonix system is depicted in Fig. 4, comprising the LMU and the proposed ROC. The LMU is comprised of a CBC followed by a Light Driver Unit (LDU). This block is responsible for emitting a laser with varying intensities and wavelengths. Subsequently, this light enters into the ROC architecture. The ROC comprises Optical Core Banks (OCB) consisting of MRs followed by the Accumulation unit, which is essential for executing neural network layers and conducting encoding calculations crucial for HDC. These operations are conducted based on the MRs within the banks.

In this architectural setup, the input light data first enters the OCB. Within the OCB, each neural network layer is executed by enabling the multiplication operation in its banks, leveraging the MRs for computation. Then, by using the Accumulation unit, the complete MAC operation needed for each layer of the neural network is performed. The output from each layer is subsequently fed back to the LMU as input for the next layer in the network. This feedback loop enables iterative processing of the input data through successive neural network layers. It is worth noting that the primary advantage of this architecture lies in mapping neural network weights to MRs, coupled with the activation of the layer results within the input light through the LDU. The weight values of the neural network are sourced

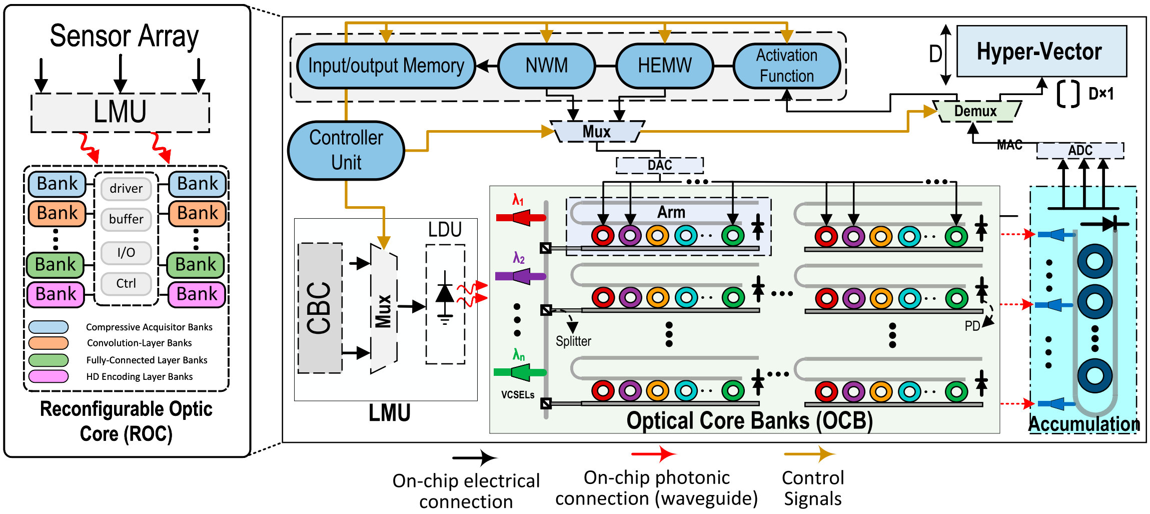


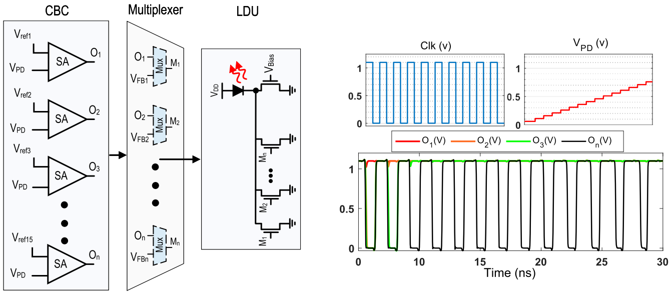
Fig. 4. Neuro-Photonix architecture consisting of a sensor array and the optical core.

from the Neural Weight Memory (NWM), facilitating efficient and integrated processing of neural network computations. This approach optimizes the utilization of optical components for weight mapping and activation, enhancing the overall performance and flexibility of the neural network implementation.

After implementing the entire network, the outputs must be multiplied with the encoding matrix to generate the HV. This multiplication operation, essential for calculating the HV, is efficiently executed using MAC operations within the OCB. To this end, the weights stored in the MRs undergo reconfiguration using Hyper-vector Encoding Memory Weights (HEMW). This process entails transferring each value of the encoding matrix, stored in the HEMW, to the MR elements. This step is crucial for encoding the neural network's output into an HV representation to improve robustness. By reconfiguring the weights stored in the MRs using the HEMW, the architecture enables the transformation of neural network outputs into hyper-dimensional vectors. We have adopted a modular design approach, allowing individual components to be adapted or replaced based on specific application requirements. This modularity ensures flexibility and facilitates the integration of new technologies while maintaining system functionality. Furthermore, we have aligned our design with emerging standards in photonic and electronic integration to enhance compatibility with existing platforms and ensure long-term scalability. A detailed explanation of the various components of the Neuro-Photonix architecture is elaborated in the following:

(a) Low-overhead Modulation Unit (LMU). The input image is acquired using a global-shutter RGB image sensor equipped with a photodiode in each pixel. The intensity of incident light generates a voltage drop in the photodiode, which is then transferred to the LMU. Within the LMU, the analog output from the pixels is directed to the CBC, which converts the analog data to 4-bit digital data. Notably, the utilization of CBC eliminates the need for power-intensive and area-consuming conventional ADC. The resulting digital data is subsequently transmitted to the LDU. This unit is designed to convert electrical input into light output at a specific wavelength and intensity using VCSEL. The input to the LDU originates either from the CBC unit or the preceding layer of the neural network. Fig. 5 illustrates the detailed design of the LMU. The CBC unit comprises 15 comparators that compare the analog output of the photodiode with reference voltages (V_{ref}) spanning the range of pixel output voltages. Each comparator in the CBC unit produces either a '0' or a '1' based on the level of the input voltage it receives. This arrangement allows the CBC to generate 16 distinct outputs, representing 4-bit digital data (Fig. 5(a)). The generated data is then fed to the LDU to produce light with different wavelengths and intensities therefore, this design eliminates the need for a high-power, accurate ADC. It is important to note that a Multiplexer (Mux) is required between the CBC and LDU to select either the data coming from the pixel array or the output of the previous neural network layer (calculated in OCB) for transmission to the LDU. The LDU consists of 15 transistors that are controlled by the output of the Mux, as shown in Fig. 5(b). The number of activated transistors in the LDU determines the current flow, resulting in higher light-intensity generation. It is noteworthy that conventional ADCs combine latch memory and an encoder with comparators to generate the appropriate output [43]. The output of such ADCs is typically in thermometer code, requiring conversion to binary [44]. In the proposed design, we replace the conventional power-intensive ADC with a CBC, whose output, generated by 15 comparators, directly serves as the input to the LDU to produce light at the desired wavelengths. The CBC relies solely on comparators to generate the thermometer code, which is directly fed into the LDU, eliminating the need for binary conversion. This approach reduces overall power consumption and improves efficiency by avoiding the use of latch memory and an encoder.

(b) Optical Core Banks (OCB). This unit comprises multiple MR banks, visually represented with color-coding as illustrated in Fig. 4. The various banks of OCB consist of a compressive acquisition, convolutional layer, fully-connected



(a) (b) (c)

Fig. 5. Schematic of (a) Comparator-based Convertor (CBC) and (b) Light Driver Unit (LDU), (c) Sample waveforms of CBC input from the pixel and respective outputs.

layer, and HD encoding layer. These components perform different DNNs and transform their outputs into HV by adjusting weight parameters and conducting matrix-vector multiplication operations as necessary.

1. Neural Dynamic Implementation. The OCB contains MR banks associated with specific weight values and partitioned into arms. These MRs modulate the intensity of incoming light based on their assigned weight values, selectively affecting light of the same wavelength as the MR. Implementing neural networks requires careful consideration of the MAC operation implementation, which is essential for efficient computation and performance. This operation involves multiplying each input by its corresponding weight and then accumulating these products to generate the output activations. To this end, in OCB, the input light data, modulated to a specific intensity and wavelength within the LDU to represent required activation values, is transmitted through an arm containing the MRs that house the weight values (see Fig. 4). As the light passes through the arm, each MR influences the intensity of light at the wavelength corresponding to that particular MR to result in W×A which is known as partial product. At the end of each arm, a photonic detector is positioned to sum up extracted partial products and convert the processed light passing through the arm into a voltage signal. This process mirrors the MAC operation, which is essential for implementing various layers of neural networks. It's important to note that the number of MAC operations in each arm is limited by the number of MRs within that arm. Therefore, implementing large layers that require a large kernel size can be achieved by segmenting the required MAC operations, each fitting within an arm's capacity. In the accumulation phase shown in Fig. 4, the summation of the MAC operations from each segment is computed to produce the overall result of the MAC operation. This involves adding together the individual products obtained from the MAC operations performed within each segment, resulting in the final output. It's important to note that this segmentation approach enables the implementation of complex neural network layers within the limitations of the hardware architecture. After the accumulation part, Neuro-Photonix requires 8 ADC with a precision of 4 to feed the results back into the system. Furthermore, due to the mentioned limitations, an electronic memory (NWM) is necessary to store the weights of the entire neural network. Another component in the electronic domain is the activation function that has been implemented at the end of each layer. During the implementation of each layer, the relevant weights are transferred from the NWM into the MR banks to execute the necessary MAC operations. This process ensures that the neural network computations can be carried out effectively within the hardware constraints. In addition to storing weights, additional memory is needed to store the output of each layer as activations, which serve as inputs for the subsequent layer in the neural network. It is noteworthy that there are implementations of non-linear activation functions utilizing optical designs to ensure uniformity across architectures. For instance, the work presented in [45] implements the GELU function. However, using memristors alongside MRs introduces design complexity and integration challenges. An MZI-based ReLU-like activation function has been implemented in [46]. Nevertheless, employing MZI-based activation functions in MR-based architectures imposes additional design overheads due to the higher power consumption of MZIs compared to MRs and having different integration processes.

2. Symbolic AI's Hyper Vector Implementation. Upon implementing the DNN, it becomes necessary to map the outputs to HV representations to enhance efficiency [41]. This mapping is achieved by multiplying the DNN's output data into the encoding HV matrix, whose values are stored in the HEMW. These stored values are utilized to reconfigure the MRs within the banks, thereby constructing the encoding matrix within the OCB. The encoding matrix is pivotal in mapping the network's output to a high-dimensional vector with a dimensionality of D, thereby enhancing precision and effectiveness in data representation. To create the HV, matrix-vector multiplication is employed to multiply the output data of the DNN with the encoding matrix. This process is initiated by inputting the DNN outputs, which are converted into light using the LDU, into the OCB containing the encoding matrix. The results of these operations yield an HV representation with a dimensionality of D. Fig. 4 visualizes this process, highlighting the flow and transformation of data as it progresses through the encoding and mapping stages to produce the HV representation ultimately.

IV. Hardware Mapping

Neuro-Photonix employs a correlated hardware mapping technique to effectively manage the computational demands of both neural dynamics and symbolic AI HV generation components. The goal is to evenly distribute the workload, thereby boosting processing speed while taking into account the physical constraints of the silicon photonics substrate and the power and area requirements for edge acceleration. We explore the design space of each component separately as follows.

A. Photonics-Friendly MAC Engine

In the proposed architecture, the OCB is responsible for executing various layers of DNNs, including the convolutional layer and fully connected layer. A weight stationary method is followed for facilitating the weight reuse processing method

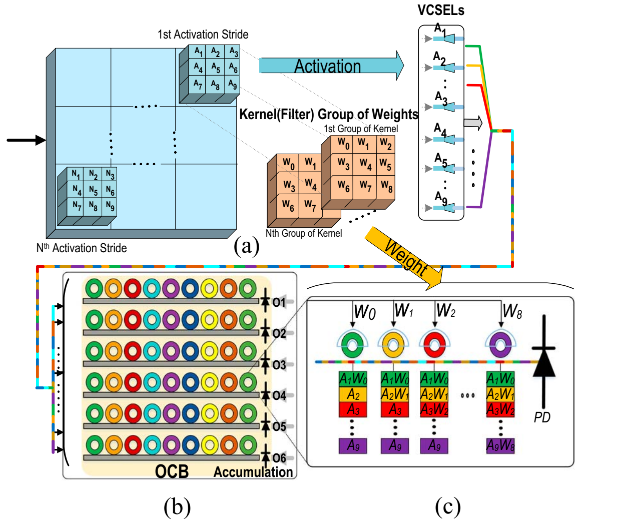

VCSELs
1st Activation Stride
A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9
Activation
Kernel(Filter) Group of Weights
1st Group of Kernel
W_0 W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8
N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9
Nth Group of Kernel
Nth Activation Stride
(a)
Weight
O1
O2
O3
O4
O5
O6
OCB
Accumulation
W_0 W_1 W_2 W_8
A_1W_0 A_2 A_3 A_9
A_1W_0 A_2W_1 A_3 A_9
A_1W_0 A_2W_1 A_3W_2 A_9
A_1W_0 A_2W_1 A_3W_2 A_9W_8
PD
(b)
(c)

Fig. 6. (a) Activation modulation, (b) Hardware mapping for 6 Strides, (c) Implementing a 3×3 kernel in an arm.

mentioned in V.E. To compute these layers, the weights of each DNN layer stored in the NWM need to be transmitted to the MRs situated within the OCB banks (refer to Fig. 6(a)). It is crucial to note that the NWM capacity is determined based on the maximum number of weights needed to accommodate the largest neural network layers. In this design, ResNet-18 is utilized as the benchmark for the maximum network size, thus a 5.5MB NWM is adequate to store the necessary 4-bit precision weights in the proposed design. In the initial step, the weights of each DNN layer are transferred from the NWM to the MRs. In the proposed design, each bank row, an arm, contains 9 MRs. This configuration is chosen due to the common use of the 3×3 kernel size in most DNN implementations. Additionally, to support kernel sizes of 5×5 and 7×7, each bank in the OCB is designed with a total of 54 MRs, which means each bank consists of 6 arms. This total of 54 MRs per bank ensures that even the maximum kernel size of 7×7 can be processed in a single clock cycle. The proposed design allows for efficient utilization of each DNN layer by implementing the MAC operation in a single cycle. For example, to implement a layer of DNN using a 3×3 kernel size, each bank can execute 6 strides (O1-O6), with each arm responsible for one stride, as illustrated in Fig. 6(b). This process involves multiplying the input light data (A1-A9) by the weights stored in the MRs of an arm (W0-W8), as shown in Fig. 6(c). The weights, received in groups of nine from NWM, are stored in each arm to perform each kernel multiplication. A Photo Detector (PD) performs the summation operation, allowing the MAC result to be directly sent out without requiring an Accumulation unit at the end of each arm. This component is deactivated (shown in gray in Fig. 6(b)) during the implementation of the 3×3 kernel. Moreover, to implement larger kernel sizes such as 5×5, 25 MRs are required for the multiplication process. This requires three arms in a bank, totaling 27 MRs, to execute one stride effectively. Since each arm cannot directly sum all 25 multiplication elements, an additional summation (Accumulation) is necessary. It should be noted that a single bank in the OCB can execute 2 strides for a 5×5 kernel size, resulting in 4 MRs being inactive. Lastly, to implement a kernel size of 7×7, a total of 49 MRs are needed for one stride, requiring an entire bank to allocate the necessary resources effectively. It is worth noting that when implementing a fully connected layer of a DNN, the entire bank is utilized for the calculation. The MAC operations required for the network are divided into 9 operations, which can be executed within each arm by mapping the corresponding weights into the MRs of the arm. This approach efficiently executes the fully connected layer within the OCB architecture. In total, the OCB consists of 96 banks arranged in an array with 8 columns and 12 rows. Each bank is equipped with 54 MRs, up to 5184 MRs across all banks. This configuration enables a maximum of 5184 MAC operations to be executed in each operational cycle of the OCB.

B. Photonics-Friendly Encoder

To achieve the conversion of output data from the DNN output into HV with acceptable efficiency, the encoding HV matrix should have a dimension of 1024. This dimension is adequate to ensure the accuracy discussed in more detail in the experimental results section. The encoding matrix is considered to be $N \times D$, where N represents the maximum output length of a DNN that can be implemented using the OCB, and D denotes the dimensionality of the HV representation. This matrix dimensionality enables efficient mapping and transformation of DNN outputs into HV representations within the constraints and capabilities of the OCB architecture. Similar to the process described in the photonics-friendly MAC engine, mapping the input data to HV involves transferring the values of the encoding HV matrix from its dedicated memory (HEMW). The matrix multiplication required for this mapping is performed by executing the MAC operation between the input data and the encoding HV matrix. For clarity, consider the scenario where the output data from the DNN implementation consists of N features. This output is processed by multiplying it row-wise with an encoding matrix of dimensions $N \times D$ to produce a resulting HV of size $1 \times D$. It is worth noting that if the number of MAC operations needed to compute the HV exceeds the MAC capacity of the OCB, the HV can be calculated in multiple iterations. In each iteration, the MRs are adjusted with appropriate values from the encoding HV matrix transferred by the HEMW to compute a portion of the HV. After calculating each part, the relevant data from the HEMW updates the MRs to proceed with computing the entire HV, as shown in Fig. 7. This iterative approach ensures efficient computation of HV even when exceeding the initial MAC capacity constraints of the OCB. It is important to mention that, during the implementation of the neural network, intermediate data for each layer is extracted at the end of the accumulation unit in each iteration. Subsequently, each column of the proposed OCB architecture, equipped with an ADC, converts the analog data into digital form, which is then stored in the input/output memory for use in subsequent layers. Therefore, this architecture employs 8 ADCs to perform the necessary data calculations within the banks of each column.

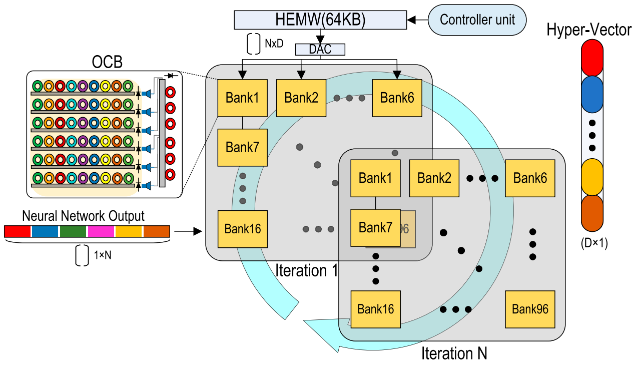


Fig. 7. Hyper-vector generation in OCB.

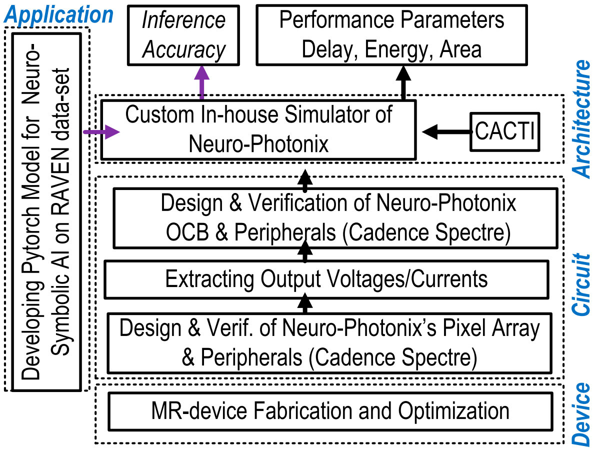

Application
Inference Accuracy
Performance Parameters Delay, Energy, Area
Developing Pytorch Model for Neuro-Symbolic AI on RAVEN data-set
Custom In-house Simulator of Neuro-Photonix
CACTI
Architecture
Design & Verification of Neuro-Photonix OCB & Peripherals (Cadence Spectre)
Extracting Output Voltages/Currents
Design & Verif. of Neuro-Photonix's Pixel Array & Peripherals (Cadence Spectre)
Circuit
MR-device Fabrication and Optimization
Device

Fig. 8. Proposed bottom-up evaluation framework.

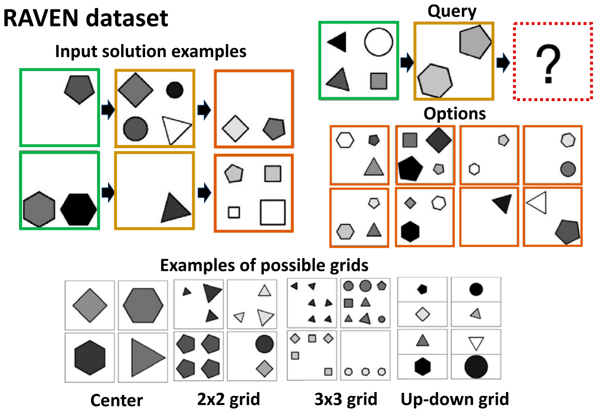


Fig. 9. Example from the RAVEN dataset.

V. Experimental Results

A. Bottom-Up Evaluation Framework

The framework, depicted in Fig. 8, comprises device, circuit, architecture, and application components. Beginning with the device level, we fabricated and optimized MR devices, acquiring circuit parameters for co-simulation with interface CMOS circuits in Cadence Spectre and SPICE. Progressing to the circuit level, we first implement the pixel array and peripheral circuitry using the 45nm NCSU Product Development Kit (PDK) library [47] in Cadence, from which we derive output voltages and currents. Subsequently, we develop all components excluding kernel banks (implemented in Cacti [48]) in Cadence Spectre. Our integrated framework combines feature extraction with reasoning at the application level and has been successfully deployed using our proprietary PyTorch AI library. Accordingly, weight parameters are extracted, quantized, fine-tuned, scaled to avoid losing performance, and mapped into the OCB which is then passed into the MRs as light using the VCSELs. The VCSELs are not fabricated but modeled in Verilog-A based on [49]. The VCSEL used in this design, derived from [49], has a delay of 10 ns and an approximate power consumption of 0.66 mW. At the architecture level, we create an in-house simulator tailored for our platform that computes both execution time and energy consumption necessary for neuro-symbolic AI alongside inference accuracy.

B. Reasoning Task

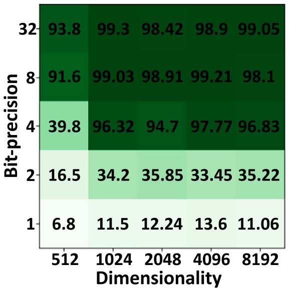
To rigorously assess the effectiveness of our algorithm, we compared our model with established models referenced in existing research, employing the RAVEN-10000 dataset. The RPM test [29] presents a prime challenge for assessing reasoning due to its combination of visual perception and abstract reasoning. This dataset includes 1,120,000 images and 70,000 instances of RAVEN's Progressive Matrices (RPM), spread evenly across seven distinct figure types. In RPM challenges, as illustrated in Fig. 9, participants complete a matrix by recognizing patterns within a series of geometric shapes, which tests their perceptual and logical abilities. This extensive dataset serves as a fundamental resource for training and evaluating AI models in reasoning tasks. In evaluating our models, we concentrated on several critical metrics: (1) Accuracy, where we calculated the percentage of puzzles correctly solved by each model, providing a direct metric of its effectiveness; (2) Computational efficiency, which examines the inference time and energy consumption per puzzle; (3) Robustness to information loss: We tested each model's ability to withstand noise and data alterations by applying quantization techniques, which is crucial for ensuring model reliability under varied conditions. These measures give a comprehensive overview of how our framework performs across different domains.

C. Accuracy Analysis

Our initial evaluation consists of a thorough comparison among several models at full precision and our edge-compatible framework, which includes various architectures such as ResNet+DRT7 [50], DCNet [51], PrAE [52], MRNet [53], SCL [54], NVSA [50], and Neuro-Photonix based on NVSA with adaptations, and human benchmarks. This set of models is documented in Table I. For instance, the NVSA model achieves an average accuracy of 98.5%, setting a benchmark for state-of-the-art reasoning tasks. Focusing on specific

TABLE I
PERFORMANCE COMPARISON OF NSAI ON THE RAVEN

Configuration	Accuracy			
	Average (%)	Centre (%)	3×3 (%)	L-R (%)
ResNet+DRT7	59.6	59.6	50.4	65.8
DCNet47	93.6	97.8	86.65	99.8
PrAE23	60.3	70.6	30.5	88.5
MRNet10	74.7	96.2	45.9	93.7
SCL44	87.2	99.9	63.5	96.8
NVSA	98.5	99.9	96.3	99.9
Neuro-Photonix (8-bit)	**97.91**	**99.03**	**95.15**	**99.39**
Neuro-Photonix (4-bit)	**93.85**	**94.76**	**90.83**	**96.06**
Human	84.4	95.5	99.39	86.4



(a)

Transfer Cost	Image	Hyper-Vector
Vector Size	16384	1024
Data Size (Byte)	65536	512
Energy (mJ)	7680	60
BLE Energy Efficiency	15mW/1Mb [55]	

(b)

Fig. 10. (a) Heatmap displaying accuracy across various dimensions and quantization levels for the CENTER RAVEN, (b) Data transfer cost of traditional reasoning at the cloud vs. Neuro-Photonix generated HVs.

model comparisons, the analysis between the high-performing NVSA and our framework, which share similar architectural foundations, offers significant insights. Although this model reached an impressive average accuracy of 98.5%, our framework was not far behind with an average accuracy of 97.91%; despite operating at a significantly reduced bit-precision, even for harder tasks such as 3×3, the accuracy drop was not higher than 2%. We achieved this high performance by strategically implementing 8-bit quantization.

Next, we explore the efficacy of Neuro-Photonix relative to standard full-precision benchmarks by examining average accuracy across a range of quantization scenarios for the neural block and the impact on dimensionality for the HDC block of the network. The findings are illustrated in Fig. 10(a), which presents a heatmap detailing the accuracy of the center configuration of the RAVEN over various dimensions (512, 1024, 2048, 8196) and bit precisions (2, 4, 8, 32, or full-precision). With standard 32-bit precision, without any quantization, the minimum dimensionality required to achieve near state-of-the-art performance is 1024. The accuracy results presented in Table I are based on 8-bit quantization. To evaluate the impact of reduced precision, we also tested the proposed design with 4-bit quantization. The NVSA model demonstrated strong robustness across varying precision levels, achieving 97.94% accuracy at full precision (32-bit) as the baseline, 97.91% at 8-bit, and 93.85% at 4-bit quantization. These results highlight that the optical implementation of the proposed design experiences only a negligible accuracy loss at 8-bit precision compared to the baseline. Furthermore, the findings confirm the model's effectiveness in resource-constrained environments, maintaining acceptable performance with minimal degradation at lower bit precisions. A reduction to a 512 dimension results in a substantial 6% decrease in accuracy at full precision. When precision is reduced to 8 bits, the accuracy of our framework slightly diminishes to 98.1% for dimensions greater than 1024, demonstrating the model's resilience to reduced precision. At this bit level, the impact of dimensionality becomes less pronounced, except at the 512 level, where the accuracy drops by 2.2% from the full-precision case and by almost 8% from state-of-the-art. This phenomenon is highly tied to the mathematical underpinnings of HDC and its connection to Random Fourier Features [56], where, depending on the complexity of the task, there is a requisite minimum dimensionality for adequate kernel approximation, which in this scenario it needs to be higher than 512. As the complexity of the tasks increases, higher dimensions are necessary to improve kernel approximation, which in turn brings higher performance.

At a quantization of 4 bits, we identified a critical trade-off for our framework at the 1024 dimensionality. Here, there is a loss of approximately 3% in accuracy regarding reasoning tasks, though this loss is not as marked as at the 512 dimensionality. For dimensions beyond 1024, no improvements in accuracy are observed, indicating a critical threshold beyond which performance degradation becomes noticeable. Overall, for bit-precisions lower than 4 bits, we observe that the quality diminishes too much for it to be usable, given the complexity and difficulty of the RAVEN reasoning dataset, which highlights the intricate balance required between model efficiency and performance. This evaluation serves as a foundational step in understanding and optimizing the trade-offs inherent in our design of the hardware architecture of Neuro-Photonix.

D. Data Transmission Cost

As shown in Fig. 10(b), to evaluate the benefits of our framework, we estimate the transmission cost to the cloud, which is often overlooked but is particularly significant. We use Bluetooth Low Energy (BLE) 4.0 [55] as the communication protocol. In the case of traditional reasoning at the cloud, the transmission cost over Bluetooth is significant since the entire input image obtained after sensing has to be transmitted to the cloud. In contrast, even though the design has to bear the overhead of encoding near-sensor, the size of the packet being transmitted is drastically reduced, diminishing transmission costs to cloud infrastructure by 128×, and minimizing the overall system-level cost of reasoning, enabling near-sensor integration.

E. Energy Consumption and Performance

To evaluate the energy consumption and performance of Neuro-Photonix, we consider the ResNet18 network. During evaluations, we realized aside from conversion energy (ADCs and DACs), the tuning of MRs also accounts for a significant portion of the architecture's energy consumption, warranting consideration. The MRs can be tuned using either the TO, EO, or a hybrid approach known as TED. The TO tuning consumes

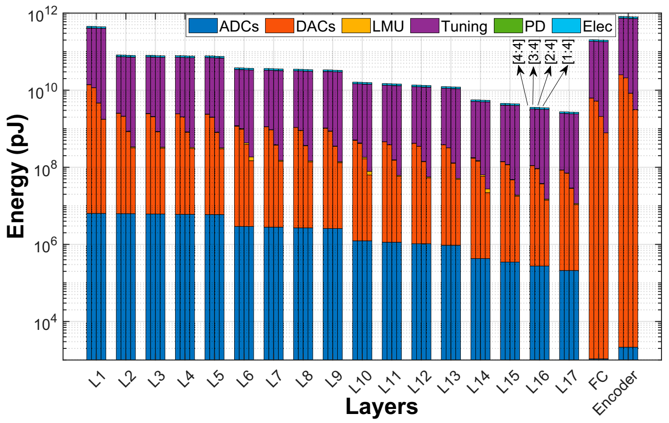


Fig. 11. Break-down of energy consumption for ResNET18 layers and encoder layer under different [W:A] in NRU.

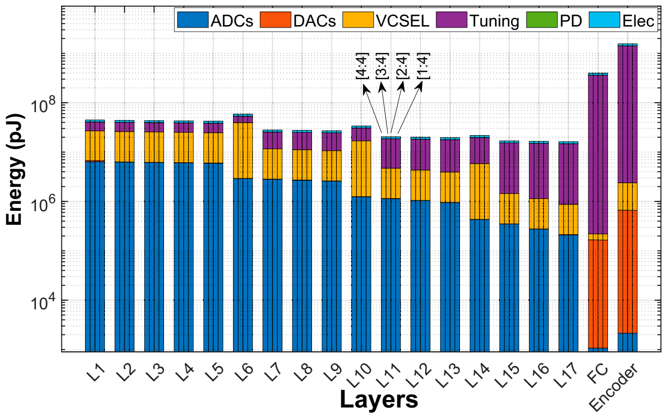


Fig. 12. Break-down of energy consumption for ResNET18 layers and encoder layer under different [W:A] in RU.

27.5 mW/FSR, offering a larger tunability range [34], while the EO tuning consumes 4 mW/nm but provides a smaller tuning range [35]. However, the TED approach effectively reduces overall tuning power consumption by combining both methods, optimizing power usage according to the tuning range. Even though the TED controlling method reduces tuning energy consumption, directly modulating the activation values to the right by controlling the driving current of the VCSELs eliminates the need for separate MR devices for activation modulation, thus halving the number of required MRs. This not only reduces the area and energy demands but also simplifies the integration process. Additionally, two evaluation methods are studied. In the first method, we assume that each time new activations are applied to the OCB. Then, the inputs are rearranged, and the MRs are retuned with weight values, regardless of whether the new activations require multiplication by new weights or not. This method is referred to as the Non-Re-Using Weights method (NRU). In the second method, we refrain from tuning new weights on the OCB until all activations intended to be multiplied by those weights are applied. Subsequently, we retune and rearrange the weights before applying their corresponding activations. The second method enables substantial savings in weight conversion energy consumed by DACs and MRs tuning energy.

Fig. 11 provides a breakdown of energy consumption components by layer for four weight and activation [W:A] configurations of the ResNET18 and encoder layer mapped on Neuro-Photonix in the NRU method. The [W:A] ratio refers to the ratio of Weight precision to Activation precision during the inference phase. This ratio is critical as it directly impacts the computational and memory requirements of the system. We observe that (1) The utilization of CBC and LDU components has resulted in a reduction of ADCs and DACs conversions for converting input image data and input activations from the previous layer, making the ADCs and DACs energy components comparable to other energy components; (2) ADCs, DACs, and tuning constitute the most significant portions of energy consumption; (3) As the number of weights in the fully connected layer of the neural dynamic and the encoder layer of the symbolic AI is larger than that of in the conv. layers, it resulted in higher DACs and tuning energies compared to the ADCs energy in these two layers; (4) Reducing the bit-width of the weights has only an insignificant impact (less than 1%) on overall energy consumption reduction. It is noteworthy that while bit precision impacts power consumption, its effect is minimal compared to the dominant energy consumption by HVs. Nonetheless, the adoption of HDC significantly mitigates the need for complex neural networks, especially in noisy environments. This approach also reduces the demands for data conversion, transmission, and processing in cloud-centric architectures, resulting in a substantial decrease in overall energy consumption.

Fig. 12 shows the breakdown of energy consumption utilizing the RU method. As depicted, utilization of the RU method has resulted in a significant reduction in the energy consumption of tuning and DACs, leading to a notable ∼800× and ∼500× reduction in the total energy required for processing the neural dynamics and symbolic AI, respectively. However, such reduction in energy consumption is smaller in the symbolic AI part as evident from Fig. 15(a) and 15(c), primarily due to the more significant number of weight parameters in the encoding layer, necessitating more DACs and tuning energy.

Layer-wise execution time of the ResNET18 layers and encoder layer using Neuro-Photonix architecture in NRU method under various [W:A] is displayed in Fig. 13. The encoder layer shows the largest execution time. In processing layers, tuning the MRs is the most time-consuming part. Consequently, with its higher number of weights, the encoder layer experiences a more significant delay than the other layers. Also, changing the bit-width of the weight has a minimal effect (less than 1%) on the execution time of the layers. Thus, reducing the bit-width and, consequently, the accuracy does not lead to a significant performance improvement. Fig. 14 shows the layer-wise execution time of ResNET18 in the RU setup. Utilizing the RU method has resulted in significant savings in the execution time of both the neural dynamics and symbolic AI parts due to the reduction in MR's tuning time. Compared to the NRU method, the RU method has reduced the required processing time of the neural dynamics layers by ∼400× and the symbolic

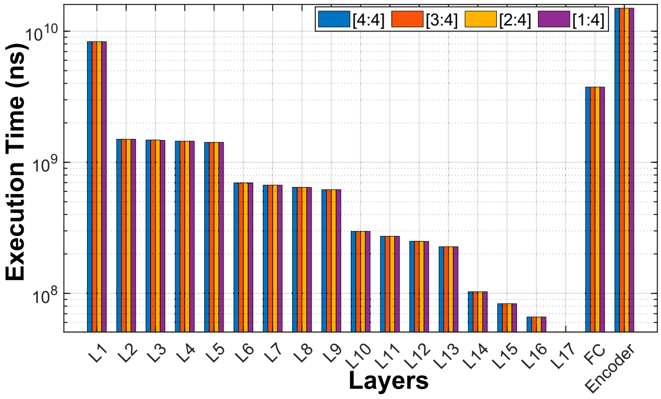


Fig. 13. Execution time of ResNET18 layers and encoder layer under different [W:A] settings in NRU method.

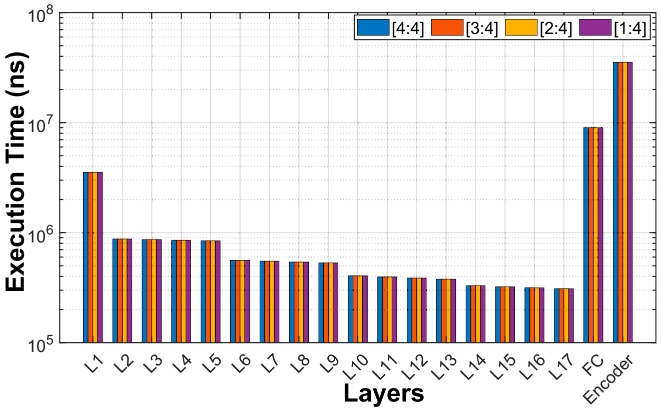


Fig. 14. Execution time of ResNET18 layers and encoder layer under different [W:A] settings in RU method.

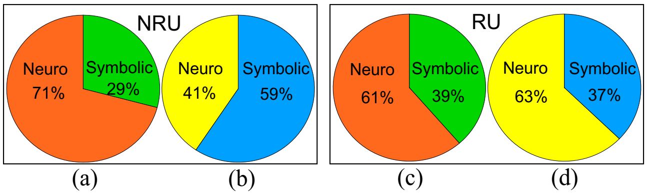


Fig. 15. Distribution of energy consumption (a) and (c) and execution time (b) and (d) for neuro-symbolic AI task.

AI layer by ~1000×. It is evident from Fig. 15(b) and (d) that the symbolic AI part has benefited the most from the RU method. As an example in the setting of [3:4], the total required energy for processing the layers of ResNET18 and the encoder layer is 2796 mJ for the NRU method and 4.1 mJ for the RU method. The processing time of the Neuro-Photonix working in the NRU and RU methods is 36.9 s and 56.4 ms, respectively. The pie charts in Fig. 15 display the distribution of energy consumption and execution time between the neuro and symbolic parts for the [3:4] configuration. We observed a significant change in the share of the Symbolic part's total execution time when transitioning from the NRU to the RU processing method, decreasing from 59% to 37%. This shift is primarily due to the tuning operation, the most time-consuming process in our processing core, which is far surpassing other component delays.

F. Comparison

1) Comparison With Electronic Accelerators: Given the pioneering nature of Neuro-Photonix as the first-of-its-kind near-sensor neuro-symbolic AI accelerator, direct performance comparisons with other designs prove challenging. Nevertheless, to shed light on its capabilities, we undertake a relative assessment of the neural dynamic component. We compare Neuro-Photonix power consumption with three prominent digital electronic accelerators, i.e., Eyeriss [57], YodaNN [58], and AppCip [16], each with distinct parallelism techniques and hardware mappings. Eyeriss utilizes a spatial architecture for energy-efficient processing, YodaNN is optimized for binary-weight CNNs, and AppCip implements parallel analog convolution-in-pixel. Our cross-layer simulation results demonstrate that the Neuro-Photonix significantly outperforms electronic accelerators for processing ResNET18. Specifically, Neuro-Photonix reduces power consumption by factors of 19×, 28×, and 17.6× compared to Eyeriss [57], YodaNN [58], and AppCip [16], respectively.

2) Comparison With Optical Accelerators: Table II presents our extensive simulation results for selected MR-based optical accelerators and Neuro-Photonix across various [W:A] configurations, compared with the baseline, an NVIDIA Geforce RTX 3060Ti GPU. The DNN accelerators under test include LightBulb [31], HolyLight [27], HQNNA [37], Robin [22], and CrossLight [20], as discussed in the background section. To ensure an unbiased evaluation, we built the designs from the ground up to closely match the original designs, using our evaluation framework and in-house simulator. The results are reported within a reasonable area constraint for all accelerators (~20-60mm^2). Taking into account the minimum phase cross-talk ratio in our architecture, we estimate the area of our proposed optical core to be (~5mm^2) [22].

Here we list our key observations. (1) We observe that the variants of Neuro-Photonix exhibit exceptional power efficiency compared to other designs on the VGG9 model running CIFAR100. For instance, Neuro-Photonix [3:4] consumes only 2.71 W, which can be supported by the low power budget of edge devices. In contrast, the best low-power accelerator, HolyLight [27], requires 66.9 W or more [22]. This significant power efficiency is achieved by (i) eliminating the need for MRs tuned by activation parameters, thereby saving the tuning power required for the MRs, and (ii) reducing the additional power and area requirements due to the extensive use of ADCs and DACs. (2) On average Neuro-Photonix reduces power consumption by ~73×, 24.68×, 30.9× compared with the baseline [32:32], HolyLight [4:4] [27], and CrossLight [4:4] [20], respectively. (3) As we reduce the weight bit-width, the power consumption can be reduced at the cost of accuracy degradation, where Neuro-Photonix [3:4] achieves ~2× power saving at the cost

TABLE II
PERFORMANCE COMPARISON WITH OPTICAL DESIGNS

Designs & [W: A]	Process node (nm)	Max Power (W)	KFPS/W	Accuracy (%)		
				MNIST	CIFAR10	CIFAR100
baseline [32:32]§	8	200	-	98.53	90.46	67.8
LightBulb [1:1] [31]	32	68.3	57.75	96.7	-	-
HolyLight [4:4] [27]	32	66.9	3.3	98.9	88.5	-
HQNNA [37]	45	-	34.6	-	89.68	61.95
Robin [1:4] [22]	45	106	46.5	-	62.5	45.6
CrossLight [4:4] [20]	-*	84-390	10.78-52.59	92.6	78.85	-
Neuro-Photonix [4:4]	**45**	**5.28**	**61.61**	**98.12**	**88.87**	**64.22**
Neuro-Photonix [3:4]	**45**	**2.71**	**117.65**	**98.05**	**86.3**	**61.04**
Neuro-Photonix [2:4]	**45**	**1.46**	**188.24**	**93.95**	**70.55**	**41.4**

§NVIDIA Geforce RTX 3060Ti GPU. *Data is not reported/not achievable in the paper.

of 3.17% accuracy drop. (4) Regarding throughput ($\frac{frame}{second}$) per watt, Neuro-Photonix [3:4] achieves 117.65 kilo FPS/W, doubling the inference performance compared to the best result reported for LightBulb [31]. Overall, taking into account the test accuracy results across three datasets, Neuro-Photonix-MX [4:4][3:4] provides the best performance-quality ratio with 84.4 kilo FPS/W.

(5) Our experiments show that Neuro-Photonix with [3:4] and [4:4] configurations maintain good accuracy across three datasets. Neuro-Photonix [4:4] achieves the second-highest accuracy on MNIST and CIFAR10, following HolyLight [27] and HQNNA [37], respectively, while offering higher KFPS/W. (6) We found that accuracy decreases with bit-width sensitivity in activations and weights. The main accuracy drop in Neuro-Photonix is due to the ADC-less imager affecting the first layer, which can be improved with hardware noise-aware training.

VI. CONCLUSION

This work introduces Neuro-Photonix as the first-of-its-kind near-sensor photonic accelerator for neural symbolic AI computing to realistically perform reasoning tasks near the sensor. In this architecture, dynamic neural implementation leverages the potential efficiency offered by photonic devices. The proposed architecture facilitates the MAC operation on analog input data, essential for various neural network layers implementation, in a single cycle. Additionally, this design supports HyperDimensional (HD) computing, enhancing the neural network's accuracy, particularly with low-precision input data. Furthermore, this approach minimizes overall power consumption, conversion latency, and processing time within both established cloud-centric architectures and newly designed accelerators. Simulation results demonstrate that Neuro-Photonix achieves 30 GOPS/W and reduces power consumption by a factor of 20.8 and 4.1 on average compared to recent ASIC baselines and photonic accelerators while preserving accuracy. For future work, we suggest further development of the Neuro-Photonix architecture to incorporate the similarity evaluation needed in HDC directly within the architecture, eliminating the traditional reliance on cloud-based processing. Additionally, the proposed design's potential for implementing MAC operations could be further leveraged by enabling approximate computing within the photonic framework.

REFERENCES

[1] E. Baccour et al., "Pervasive AI for IoT applications: A survey on resource-efficient distributed artificial intelligence," *IEEE Commun. Surveys & Tut.*, vol. 24, no. 4, pp. 2366–2418, 2022.

[2] R. Song, K. Huang, Z. Wang, and H. Shen, "A reconfigurable convolution-in-pixel CMOS image sensor architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 7212–7225, Oct. 2022.

[3] H. Xu et al., "MACSen: A processing-in-sensor architecture integrating MAC operations into image sensor for ultra-low-power BNN-based intelligent visual perception," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 68, no. 2, pp. 627–631, Feb. 2021.

[4] S. Angizi, S. Tabrizchi, D. Z. Pan, and A. Roohi, "PISA: A non-volatile processing-in-sensor accelerator for imaging systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 11, no. 4, pp. 962–972, Oct.–Dec. 2023.

[5] P. Lu, L. Qiu, W. Yu, S. Welleck, and K.-W. Chang, "A survey of deep learning for mathematical reasoning," 2022, *arXiv:2212.10535*.

[6] X. Yang et al., "Neuro-symbolic computing: Advancements and challenges in hardware–software co-design," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 71, no. 3, pp. 1683–1689, Mar. 2024.

[7] A. d. Garcez and L. C. Lamb, "Neurosymbolic AI: The 3rd wave," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 12387–12406, 2023.

[8] Z. Wan et al., "H3Dfact: Heterogeneous 3D integrated CIM for factorization with holographic perceptual representations," in *Proc. Des., Automat. & Test Europe Conf. & Exhib. (DATE),* Piscataway, NJ, USA: IEEE Press, 2024, pp. 1–6.

[9] J. Tian, Y. Li, W. Chen, L. Xiao, H. He, and Y. Jin, "Weakly supervised neural symbolic learning for cognitive tasks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 5, 2022, pp. 5888–5896.

[10] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," 2019, *arXiv:1904.12584*.

[11] M. Qu and J. Tang, "Probabilistic logic neural networks for reasoning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[12] S. J. Carey, A. Lopich, D. R. Barr, B. Wang, and P. Dudek, "A 100,000 FPS vision sensor with embedded 535535GOPS/W 256× 256 SIMD processor array," in *Proc. Symp. VLSI Circuits,* Piscataway, NJ, USA: IEEE Press, 2013, pp. C182–C183.

[13] T.-H. Hsu et al., "A 0.5-V real-time computational CMOS image sensor with programmable kernel for feature extraction," *IEEE J. Solid-State Circuits*, vol. 56, no. 5, pp. 1588–1596, May 2021.

[14] T. Yamazaki et al., "4.9 a 1ms high-speed vision chip with 3d-stacked 140GOPS column-parallel pes for spatio-temporal image processing," in *ProcIEEE Int. Solid-State Circuits Conf. (ISSCC),* Piscataway, NJ, USA: IEEE Press, 2017, pp. 82–83.

[15] H. Xu et al., "Senputing: An ultra-low-power always-on vision perception chip featuring the deep fusion of sensing and computing," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 69, no. 1, pp. 232–243, Jan. 2022.

[16] S. Tabrizchi, A. Nezhadi, S. Angizi, and A. Roohi, "AppCiP: Energy-efficient approximate convolution-in-pixel scheme for neural network acceleration," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 13, no. 1, pp. 225–236, 2023.

[17] A. El Gamal, D. X. Yang, and B. A. Fowler, "Pixel-level processing: Why, what, and how?" in *Proc. Sensors, Cameras, Appl. Digit. Photogr.*, vol. 3650, Bellingham, WA, USA: SPIE, 1999, pp. 2–13.

[18] J. Choi, S. Park, J. Cho, and E. Yoon, "An energy/illumination-adaptive CMOS image sensor with reconfigurable modes of operations," *IEEE J. Solid-State Circuits*, vol. 50, no. 6, pp. 1438–1450, 2015.

[19] T.-H. Hsu et al., "AI edge devices using computing-in-memory and processing-in-sensor: From system to device," in *Proc. IEEE Int. Electron Devices Meeting (IEDM),* Piscataway, NJ, USA: IEEE Press, 2019, pp. 22–5.

[20] F. Sunny, A. Mirza, M. Nikdast, and S. Pasricha, "CrossLight: A cross-layer optimized silicon photonic neural network accelerator," in *Proc. 58th ACM/IEEE Des. Automat. Conf. (DAC),* Piscataway, NJ, USA: IEEE Press, 2021, pp. 1069–1074.

[21] R. LiKamWa, Y. Hou, J. Gao, M. Polansky, and L. Zhong, "RedEye: Analog convnet image sensor architecture for continuous mobile vision," *ACM SIGARCH Comput. Archit. News*, vol. 44, no. 3, pp. 255–266, 2016.

[22] F. P. Sunny, A. Mirza, M. Nikdast, and S. Pasricha, "ROBIN: A robust optical binary neural network accelerator," *ACM Trans. Embedded Comput. Syst.*, vol. 20, no. 5s, pp. 1–24, 2021.

[23] M. Morsali et al., "Oisa: Architecting an optical in-sensor accelerator for efficient visual computing," in *Proc. Des., Automat. & Test Europe Conf. & Exhib. (DATE),* Piscataway, NJ, USA: IEEE Press, 2024, pp. 1–6.

[24] Q. Cheng, J. Kwon, M. Glick, M. Bahadori, L. P. Carloni, and K. Bergman, "Silicon photonics codesign for deep learning," *Proc. IEEE*, vol. 108, no. 8, pp. 1261–1282, Aug. 2020.

[25] M. Morsali et al., "Lightator: An optical near-sensor accelerator with compressive acquisition enabling versatile image processing," in *Proc. 61st ACM/IEEE Des. Automat. Conf.*, 2024, pp. 1–6.

[26] C. Wang, Y. Wang, and H. Li, "A new optical information processing device design for internet of brain things application," *IEEE Access*, vol. 6, pp. 75052–75061, 2018.

[27] W. Liu, W. Liu, Y. Ye, Q. Lou, Y. Xie, and L. Jiang, "Holylight: A nanophotonic accelerator for deep learning in data centers," in *Proc. Des., Automat. & Test Europe Conf. & Exhib. (DATE),* Piscataway, NJ, USA: IEEE Press, 2019, pp. 1483–1488.

[28] L. Arrotta et al., "Neuro-symbolic AI approaches for sensor-based human activity recognition," Doctoral thesis, 2024. [Online]. Available: https://air.unimi.it/handle/2434/1022155

[29] C. Zhang, F. Gao, B. Jia, Y. Zhu, and S.-C. Zhu, "Raven: A dataset for relational and analogical visual reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5317–5327.

[30] F. P. Sunny, A. Mirza, I. Thakkar, M. Nikdast, and S. Pasricha, "ARXON: A framework for approximate communication over photonic networks-on-chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 6, pp. 1206–1219, Jun. 2021.

[31] F. Zokaee, Q. Lou, N. Youngblood, W. Liu, Y. Xie, and L. Jiang, "LightBulb: A photonic-nonvolatile-memory-based accelerator for binarized convolutional neural networks," in *Proc. Des., Automat. & Test Europe Conf. & Exhib. (DATE),* Piscataway, NJ, USA: IEEE Press, 2020, pp. 1438–1443.

[32] Z. Zhao et al., "Hardware-software co-design of slimmed optical neural networks," in *Proc. 24th Asia South Pacific Des. Automat. Conf.*, 2019, pp. 705–710.

[33] W. Bogaerts et al., "Silicon microring resonators," *Laser & Photon. Rev.*, vol. 6, no. 1, pp. 47–73, 2012.

[34] P. Pintus et al., "PWM-driven thermally tunable silicon microring resonators: Design, fabrication, and characterization," *Laser & Photon. Rev.*, vol. 13, no. 9, 2019, Art. no. 1800275.

[35] S. Abel et al., "A hybrid barium titanate–silicon photonics platform for ultraefficient electro-optic tuning," *J. Lightw. Technol.*, vol. 34, no. 8, pp. 1688–1693, 2016.

[36] M. Milanizadeh, D. Aguiar, A. Melloni, and F. Morichetti, "Canceling thermal cross-talk effects in photonic integrated circuits," *J. Lightw. Technol.*, vol. 37, no. 4, pp. 1325–1332, 2019.

[37] F. Sunny, M. Nikdast, and S. Pasricha, "A silicon photonic accelerator for convolutional neural networks with heterogeneous quantization," in *Proc. Great Lakes Symp. VLSI*, 2022, pp. 367–371.

[38] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," *IEEE Circuits Syst. Mag.*, vol. 20, no. 2, pp. 30–47, Jun. 2020.

[39] S. Zhang, R. Wang, J. J. Zhang, A. Rahimi, and X. Jiao, "Assessing robustness of hyperdimensional computing against errors in associative memory," in *Proc. IEEE 32nd Int. Conf. Appl. Specific Syst., Architect. Processors (ASAP),* Piscataway, NJ, USA: IEEE Press, 2021, pp. 211–217.

[40] G. Karunaratne, M. Le Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian, "In-memory hyperdimensional computing," *Nature Electron.*, vol. 3, no. 6, pp. 327–337, 2020.

[41] D. Ma, C. Hao, and X. Jiao, "Hyperdimensional computing vs. neural networks: Comparing architecture and learning process," in *Proc. 25th Int. Symp. Qual. Electron. Des. (ISQED),* Piscataway, NJ, USA: IEEE Press, 2024, pp. 1–5.

[42] A. W. Fang, H. Park, O. Cohen, R. Jones, M. J. Paniccia, and J. E. Bowers, "Electrically pumped hybrid algainas-silicon evanescent laser," *Opt. Exp.*, vol. 14, no. 20, pp. 9203–9210, 2006.

[43] Y. K. Upadhyaya, M. K. Gupta, M. Hasan, and S. Maheshwari, "High-density magnetic flash ADC using domain-wall motion and pre-charge sense amplifiers," *IEEE Trans. Magn.*, vol. 52, no. 6, pp. 1–10, Jun. 2016.

[44] M. Ajanya and G. T. Varghese, "Thermometer code to binary code converter for flash ADC-a review," in *Proc. Int. Conf. Control, Power, Communication Comput. Technol. (ICCPCCT),* Piscataway, NJ, USA: IEEE Press, 2018, pp. 502–505.

[45] S. Afifi, F. Sunny, M. Nikdast, and S. Pasricha, "TRON: Transformer neural network acceleration with non-coherent silicon photonics," in *Proc. Great Lakes Symp. VLSI*, 2023, pp. 15–21.

[46] I. A. Williamson, T. W. Hughes, M. Minkov, B. Bartlett, S. Pai, and S. Fan, "Reprogrammable electro-optic nonlinear activation functions for optical neural networks," *IEEE J. Sel. Topics Quantum Electronics*, vol. 26, no. 1, pp. 1–12, Jan./Feb. 2019.

[47] "NCSU EDA Freepdk45." 2011. [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK45:Contents

[48] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "Cacti 5.1 technical report," HP Laboratories, Palo Alto, 2008.

[49] Z. Ruan, Y. Zhu, P. Chen, Y. Shi, S. He, X. Cai, and L. Liu, "Efficient hybrid integration of long-wavelength vcsels on silicon photonic circuits," *J. Lightw. Technol.*, vol. 38, no. 18, pp. 5100–5106, 2020.

[50] M. Hersche, M. Zeqiri, L. Benini, A. Sebastian, and A. Rahimi, "A neuro-vector-symbolic architecture for solving raven's progressive matrices," *Nature Mach. Intell.*, vol. 5, no. 4, pp. 363–375, 2023.

[51] T. Zhuo and M. Kankanhalli, "Effective abstract reasoning with dual-contrast network," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2021, pp. 1–13.

[52] C. Zhang, B. Jia, S.-C. Zhu, and Y. Zhu, "Abstract spatial-temporal reasoning via probabilistic abduction and execution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9736–9746.

[53] Y. Benny, N. Pekar, and L. Wolf, "Scale-localized abstract reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12557–12565.

[54] Y. Wu, H. Dong, R. Grosse, and J. Ba, "The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning," 2020, *arXiv:2007.04212*.

[55] J. Decuir et al., "Bluetooth 4.0: low energy," Cambridge, UK: Cambridge Silicon Radio SR plc, vol. 16, p. 180, 2010.

[56] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1–8.

[57] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[58] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "YodaNN: An architecture for ultralow power binary-weight CNN acceleration," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 1, pp. 48–60, Jan. 2018.