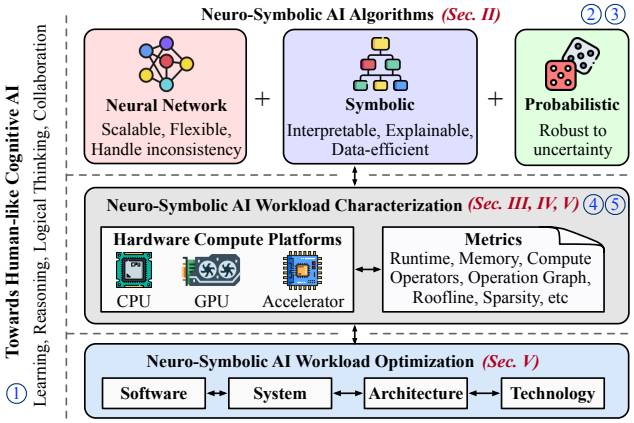
Towards Cognitive AI Systems: Workload and Characterization of Neuro-Symbolic AI

Zishen Wan[1], Che-Kai Liu[1], Hanchen Yang[1], Ritik Raj[1], Chaojian Li[1], Haoran You[1], Yonggan Fu[1], Cheng Wan[1], Ananda Samajdar[2], Yingyan (Celine) Lin[1], Tushar Krishna[1], and Arijit Raychowdhury[1]

[1]*Georgia Institute of Technology, Atlanta, GA, USA* [2]*IBM Research, Yorktown Heights, NY, USA*

***Abstract*—The remarkable advancements in artificial intelligence (AI), primarily driven by deep neural networks, are facing challenges surrounding unsustainable computational trajectories, limited robustness, and a lack of explainability. To develop next-generation cognitive AI systems, neuro-symbolic AI emerges as a promising paradigm, fusing neural and symbolic approaches to enhance interpretability, robustness, and trustworthiness, while facilitating learning from much less data. Recent neuro-symbolic systems have demonstrated great potential in collaborative human-AI scenarios with reasoning and cognitive capabilities. In this paper, we aim to understand the workload characteristics and potential architectures for neuro-symbolic AI. We first systematically categorize neuro-symbolic AI algorithms, and then experimentally evaluate and analyze them in terms of runtime, memory, computational operators, sparsity, and system characteristics on CPUs, GPUs, and edge SoCs. Our studies reveal that neuro-symbolic models suffer from inefficiencies on off-the-shelf hardware, due to the memory-bound nature of vector-symbolic and logical operations, complex flow control, data dependencies, sparsity variations, and limited scalability. Based on profiling insights, we suggest cross-layer optimization solutions to improve the performance, efficiency, and scalability of neuro-symbolic computing. Finally, we discuss the challenges and potential future directions of neuro-symbolic AI from both system and architectural perspectives.**



Challenges	Research Opportunities (Sec. VI)
Limited evaluation tasks	① Building more cognitive datasets/testbeds
Ad-hoc model design	② Unifying neuro-symbolic-probabilistic models
Limited extensibility	③ Developing efficient software frameworks
Large #algos and #HW	④ Benchmarking diverse neuro-symbolic workloads
Heterog. compute kernels	⑤ Designing cognitive architectures

Fig. 1: Overview of neuro-symbolic AI systems, workload characterizations, optimization solutions, challenges, and research opportunities in improving the performance of next-generation cognitive AI.

I. Introduction

The remarkable advancements in AI have had a profound impact on our society. These advancements are primarily driven by deep neural networks and a virtuous cycle involving large networks, extensive datasets, and augmented computing power. As we reap the benefits of this success, there is growing evidence that continuing our current trajectory may not be viable for realizing AI's full potential. First, the escalating computational requirements and energy consumption associated with AI are on an unsustainable trajectory [1], threatening to reach a level that could stifle innovation by restricting it to fewer organizations. Second, the lack of robustness and explainability remains a significant challenge, likely due to inherent limitations in current learning methodologies [2], [3]. Third, contemporary AI systems often operate in isolation with limited collaboration among humans and other AI agents. Hence, it is imperative to develop next-generation AI paradigms that address the growing demand for enhanced efficiency, explainability, and trust in AI systems.

Neuro-symbolic AI represents an emerging AI paradigm that integrates the neural and symbolic approaches with probabilistic representations to enhance explainability, robustness and facilitates learning from much less data in AI (Fig. 1). Neural methods are highly effective in extracting complex features from data for vision and language tasks. On the other hand, symbolic methods enhance explainability and reduce the dependence on extensive training data by incorporating established models of the physical world, and probabilistic representations enable cognitive systems to more effectively handle uncertainty, resulting in improved robustness under unstructured conditions. The synergistic fusion of neural and symbolic methods positions neuro-symbolic AI as a promising paradigm capable of ushering in the third wave of AI [4]–[7].

Neuro-symbolic AI promises possibilities for systems that acquire human-like communication and reasoning capabilities, enabling them to recognize, classify, and adapt to new situations autonomously. For example, neuro-vector-symbolic architecture [8] is able to reach 98.8% accuracy on spatial-temporal reasoning tasks, greatly surpassing human performance (84.4%), neuro-only ResNet (53.4%) and GPT-4 performance (89.0%). In addition to its superior performance in vision and language [9]–[11], neuro-symbolic AI holds sig-

nificant potential for enhancing explainability and trustworthiness of collaborative human-AI applications [12]–[14]. These applications include collaborative robotics, mixed-reality systems, and human-AI interactions, where robots can seamlessly interact with humans in environments, agents can reason and make decisions in an explainable manner, and intelligence is pervasively embedded and untethered from the cloud.

Despite the promising algorithmic performance, the higher memory intensity, greater kernel heterogeneity, and access pattern irregularity of neuro-symbolic computing lead to an increasing divergence from the current hardware roadmap that largely optimizes for matrix multiplication and convolution [15]–[21] and lead to severe inefficiencies and underutilization of hardware. Therefore, understanding its computational and memory demands is essential for efficient processing on both general-purpose and custom hardware.

Our goal in this work is to quantify the workload characteristics and potential system architecture for neuro-symbolic AI. To this end, we first systematically review and categorize state-of-the-art neuro-symbolic AI workloads in a structured manner (Sec. II). We then characterize seven representative neuro-symbolic workloads on general-purpose and edge platforms, analyzing their runtime, memory, compute operators, operation graph, hardware utilization, and sparsity characteristics (Secs. III, IV, V). Our workload characterization provides new observations and insights, including the following:

- Neuro-symbolic AI models typically exhibit high latency compared to neural models, prohibiting them from real-time applications.
- The neural components mainly consist of MatMul and Convs, while the symbolic components are dominated by vector/element-wise tensor and logical operations which are computed inefficiently on off-the-shelf CPUs/GPUs and may result in system bottlenecks.
- The hardware inefficiency of symbolic operations typically is due to low ALU utilization, low cache hit rates, and high volume of data movement.
- The neural workloads are compute-bounded while the symbolic workloads are typically memory-bounded and face potential scalability issues.
- The symbolic operations may depend on the neural module results or need to compile into the neural structure, thus lying on the critical path of end-to-end neuro-symbolic systems.
- Some neural and vector-symbolic components demonstrate a high level of unstructured sparsity with variations under different task scenarios and attributes.

Inspired by our workload profiling insights, we recommend several cross-layer software and hardware optimization solutions to improve the efficiency and scalability of neuro-symbolic systems (Sec. V). Finally, we explore the research opportunities in neuro-symbolic computing and share our outlook on the road ahead (Sec. VI).

To the best of our knowledge, this is one of the first works to assess neuro-symbolic computing from both system and architectural perspectives. We aim to inspire the design of next-generation cognitive computing systems through synergistic advancements in neuro-symbolic algorithms, systems, architecture, and algorithm-hardware co-design.

II. Neuro-Symbolic AI Algorithms

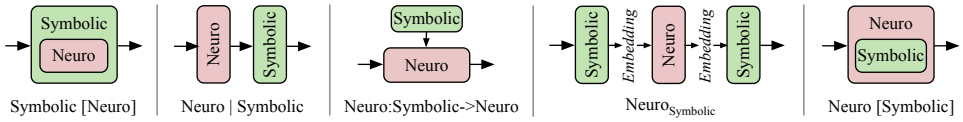
In this section, we systematically review and categorize the recent research progress in neuro-symbolic AI algorithms.

Overview. Neuro-symbolic AI represents an interdisciplinary approach that synergistically combines symbolic reasoning with neural network (NN) learning to create intelligent systems, leveraging the complementary strengths of both to enhance the accuracy and interpretability of the resulting models. Given that neuro-symbolic algorithms incorporate both symbolic and neural components, various paradigms can be categorized based on how these components are integrated into a cohesive system. Inspired by Henry Kautz's taxonomy [35], we systematically categorize these algorithms into five paradigms (Tab. I). We elaborate on each of these paradigms below. Additionally, Tab. II provides examples of several underlying operations based on the categorization in Tab. I.

Symbolic[Neuro] refers to an intelligent system that empowers symbolic reasoning with the statistical learning capabilities of NNs. These systems typically consist of a comprehensive symbolic problem solver that includes loosely-coupled neural subroutines for statistical learning. Examples include DeepMind's AlphaGo [22] and AlphaZero [36], which use Monte-Carlo Tree Search (MCTS) as the symbolic solver and NN state estimators for learning statistical patterns.

Neuro|Symbolic refers to a hybrid system that combines a neural system and a symbolic system in a pipeline, where each component typically specializes in complementary tasks within the pipeline. To the best of our knowledge, the majority of neuro-symbolic algorithms fall into this category. For example, IBM's neuro-vector-symbolic architecture (NVSA) [8] uses an NN as the perception frontend for semantic parsing and a symbolic reasoner as the backend for probabilistic abductive reasoning on the RAVEN [37] and I-RAVEN [38] datasets. Probabilistic abduction and execution (PrAE) learner [26] adopts a similar approach where the difference lies in features are first projected to high-dimensional vectors in NVSA, whereas PrAE utilizes the original features directly as the NN's input. Other examples include vector symbolic architecture-based image-to-image translation (VSAIT) [11], neuro-probabilistic soft logic (NeuPSL) [23], neural probabilistic logic programming (DeepProbLog) [39], neuro-answer set programming (NeurASP) [10], neural symbolic dynamic reasoning [40], neural symbolic concept learner (NSCL) [9], abductive learning (ABL) [24], and neuro-symbolic visual question answering (NSVQA) [25] on the CLEVRER dataset [41].

Neuro:Symbolic→Neuro approach incorporates symbolic rules into NNs to guide the learning process, where symbolic knowledge is compiled into the structure of neural models for enhancing the model interpretability. For instance, logical NNs (LNNs) [27], [42] encode knowledge or domain expertise



Category	Category Description	Neuro-Symbolic Algorithm	Underlying Operation	If Vector
Symbolic[Neuro]	End-to-end **symbolic** system that uses **neural** models internally as a subroutine	**AlphaGo** [22]	NN, MCTS	Vector
Neuro|Symbolic	Pipelined system that integrates **neural** and **symbolic** components where each component specializes in complementary tasks within the whole system	**NVSA** [8]	NN, mul, add, circular conv.	Vector
		NeuPSL [23]	NN, fuzzy logic	Vector
		NSCL [9]	NN, add, mul, div, log	Vector
		NeurASP [10]	NN, logic rules	Non-Vector
		ABL [24]	NN, logic rules	Non-Vector
		NSVQA [25]	NN, pre-defined objects	Non-Vector
		VSAIT [11]	NN, binding/unbinding	Vector
		PrAE [26]	NN, logic rules, prob. abduction	Vector
Neuro:Symbolic→Neuro	End-to-end **neural** system that compiles **symbolic** knowledge externally	**LNN** [27]	NN, fuzzy logic	Vector
		Symbolic Math [28]	NN	Vector
		Differentiable ILP [29]	NN, fuzzy logic	Vector
Neuro$_{\text{Symbolic}}$	Pipelined system that maps **symbolic** first-order logic onto embeddings serving as soft constraints or regularizers for **neural** model	**LTN** [30]	NN, fuzzy logic	Vector
		DON [31]	NN	Vector
Neuro[Symbolic]	End-to-end **neural** system that uses **symbolic** models internally as a subroutine	**GNN+attention** [32]	NN, SpMM, SDDMM	Vector
		ZeroC [33]	NN (energy-based model, graph)	Vector
		NLM [34]	NN, permutation	Vector

TABLE I: Review of recent neuro-symbolic AI algorithms into five categories, with their underlying operations and vector formats.

TABLE II: Enumeration of the underlying operations based on Tab. I.

Underlying Operations	Examples
Fuzzy logic (LTN)	$F = \forall x(isCarnivor(s)) \rightarrow (isMammal(x))$ $\{isCarnivor(s){:}[0,1], isMammal(x){:}[1,0]\} \rightarrow F = [1,0]$
Mul, Add, and Circular Conv. (NVSA)	$X_i \in \{+1,-1\}^d \rightarrow (X_i \cdot X_j)/(X_i + X_j)$
Logic rules (ABL)	Domain: $animal(dog).carnivore(dog).mammal(dog)$ Logical formula: $mammal(x) \wedge carnivore(x)$ ABL: $hypos(x) : -animal(x), mammal(x), carnivore(x)$
Pre-defined objects (NSVQA)	`equal_color`: $(entry, entry) \rightarrow Boolean$ `equal_integer`: $(number, number) \rightarrow Boolean$

as symbolic rules (first-order logic or fuzzy logic) that act as constraints on the NN output. Other examples include the application of deep learning for symbolic mathematics [28] and differentiable inductive logic programming (ILP) [29].

Neuro$_{\text{Symbolic}}$ is a type of hybrid approach that combines symbolic logic rules with NNs. It involves mapping symbolic logic rules onto embeddings that serve as soft constraints or regularizers on the NN's loss function. Logical tensor networks (LTNs) [30], for instance, use logical formulas to define constraints on the tensor representations, which have proven successful in knowledge graph completion tasks. These tasks aim to predict missing facts or relationships between entities. Other examples of this approach include deep ontology networks (DONs) [31] and tensorization methods [43]. As inference is still governed by NNs, it remains a research question whether this approach will compromise interpretability.

Neuro[Symbolic] refers to a system that empowers NNs with the explainability and robustness of symbolic reasoning. Unlike **Symbolic[Neuro]**, where symbolic reasoning is used to guide the neural model learning process, in **Neuro[Symbolic]**, the neural model incorporates symbolic reasoning by paying attention to a specific symbolic at certain conditions. For instance, graph neural networks (GNNs) are adopted for representing symbolic expressions when endowed with attention mechanisms [32]. In particular, this attention mechanism can be leveraged to incorporate symbolic rules into GNN models, enabling selective attention to pertinent symbolic information in the graph. Other examples include neural logic machines (NLM) [34] and Zero-shot concept recognition and acquisition (ZeroC) [33]. ZeroC leverages the graph representation where the constituent concept models are represented as nodes and their relations are represented by edges.

Each neuro-symbolic category reflects different kernel operators and data dependencies. *Therefore, this paper takes one of the first steps towards understanding its computing characteristics and aims to serve as a cornerstone for the design and deployment of future neuro-symbolic systems.*

III. Representative Neuro-Symbolic Models

This section presents selected widely-used neuro-symbolic AI workloads as representative ones for our analysis. We consider them representative because they are diverse in terms of applications, model structures, and computational patterns.

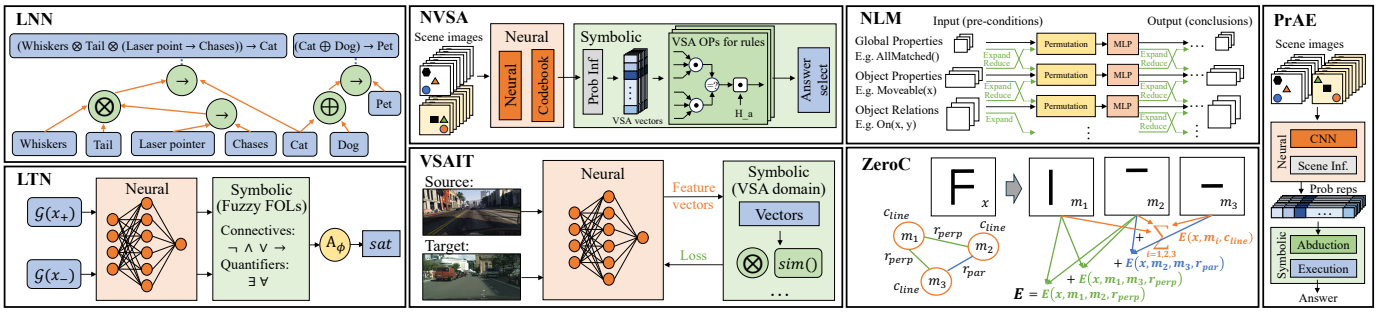
A. Model Overview.

We select seven neuro-symbolic AI models for profiling analysis (Tab. III): LNN on logic program tasks [27], LTN on querying and reasoning tasks [30], NVSA [8] on the Raven's Progressive Matrices task [37], NLM on relational reasoning and decision making tasks [34], VSAIT on unpaired image-to-image translation tasks [11], ZeroC on cross-domain classification and detection tasks [33], and PrAE on spatial-temporal reasoning tasks [26]. These selected workloads represent Neuro:Symbolic→Neuro, Neuro$_{\text{Symbolic}}$, Neuro|Symbolic, and Neuro[Symbolic] systems (Sec. II), respectively. Interested readers could refer to their references for more details.

B. Logical Neural Network (LNN)

LNN is a neuro-symbolic framework to simultaneously provide key properties of both neural (learning) and symbolic logic (knowledge and reasoning) – toward direct interpretability, utilization of rich domain knowledge realistically, and the general problem-solving ability of a full theorem prover [27].

Algorithm Description. LNNs create a one-to-one correspondence between neurons and the elements of logical



Representative Neuro-Symbolic AI Workloads		Logic Neural Network [27]	Logic Tensor Network [30]	Neuro-Vector-Symbolic Architecture [8]	Neural Logic Machine [34]	Vector Symbolic Architecture Image2Image Translation [11]	Zero-shot Concept Recognition and Acquisition [33]	Probabilistic Abduction and Execution [26]
Abbreviation		LNN	LTN	NVSA	NLM	VSAIT	ZeroC	PrAE
Neuro-Symbolic Category		Neuro:Symbolic→Neuro	$\text{Neuro}_{\text{Symbolic}}$	Neuro|Symbolic	Neuro[Symbolic]	Neuro|Symbolic	Neuro[Symbolic]	Neuro|Symbolic
Learning Approach		Supervised	Supervised/Unsupervised	Supervised/Unsupervised	Supervised/Unsupervised	Supervised	Supervised	Supervised/Unsupervised
Deployment Scenario	**Application**	Learning and reasoning, Full theorem prover	Querying, learning, reasoning (relational and embedding learning, query answering)	Fluid intelligence, Abstract reasoning	Relational reasoning, Decision making	Unpaired image-to-image translation	Cross-domain classification and detection, Concept acquisition	Fluid intelligence, Spatial-temporal reasoning
	Advantage vs. Neural Model	Higher interoperability, resilience to incomplete knowledge, generalization	Higher data efficiency, comprehensibility, out-of-distribution generalization	Higher joint representations efficiency, abstract reasoning capability, transparency	Higher generalization, logic reasoning, deduction, explainability capability	Address semantic flipping and hallucinations issue in unpaired image translation tasks	Higher generalization, concept acquisition and recognition, compositionality capability	Higher generalization, transparency, interpretability, and robustness
	Dataset	LUBM benchmark [44], TPTP benchmark [45]	UCI [46], Leptograpsus crabs [47], DeepProbLog [48]	RAVEN [37], I-RAVEN [38], PGM [49]	Family graph reasoning, sorting, path finding [50]	GTA [51], Cityscapes [52], Google Maps dataset [53]	Abstraction reasoning [54], Hierarchical-concept corpus [55]	RAVEN [37], I-RAVEN [38], PGM [49]
Computation Pattern	**Datatype**	FP32	FP32	FP32	FP32	FP32	INT64	FP32
	Neural	Graph	MLP	ConvNet	Sequential tensor	ConvNet	Energy-based network	ConvNet
	Symbolic	FOL/Logical operation	FOL/Logical operation	VSA/Vector operation	FOL/Logical operation	VSA/Vector operation	Graph, vector operation	VSA/Vector operation

TABLE III: Selected neuro-symbolic AI workloads for analysis, representing a diverse of categories, applications, and computational patterns.

formulae, using the observation that weights of neurons can be constrained to act as, e.g., AND or OR gates. At a high level, LNNs use parameterized functions to represent logical connectives (e.g. $\wedge, \vee$). This is done by defining constraints that ensure the functions behave like the corresponding logical operators. These logical connectives are implemented using learnable parameters, subject to certain constraints to maintain their logical properties. LNN then combines facts and logical rules within a neural network framework by mapping such an NN to weighted real-valued logics via Łukasiewicz logic [30].

Advantage over Neural Model. Compared with neural models, LNNs exhibit remarkable per-neuron interoperability via full logical expressivity, improved tolerance to incomplete knowledge via truth bounds, and diverse-task generality via omnidirectional inference. The LNN structure is compositional and modular, and the representation is disentangled and in probabilistic semantics, excelling in theorem prover tasks.

C. Logical Tensor Network (LTN)

LTN is a neuro-symbolic framework that supports querying, learning, and reasoning with both rich data and abstract knowledge [30]. By representing the degree of real-world knowledge as continuous and differentiable fuzzy first-order logic (FOL), LTN provides a uniform language to compute efficiently AI tasks such as multi-label classification, relational learning, data clustering, semi-supervised learning, regression, embedding learning and query answering [56]–[58].

Algorithm Description. LTN introduces a fully differentiable logical language whereby the elements of an FOL signature are grounded onto data using neural computational graphs and first-order fuzzy logic semantics. Connectives ($\wedge, \vee, \neg, \rightarrow$) are transformed into real values with fuzzy logic. Object features are represented with vectors with real values. Quantifiers ($\forall, \exists$) are interpreted with approximate aggregations [30]. With the fuzzy FOL input being transformed into tensors, a network is then exploited to compute the degree of truth with a given embedded tensor representation of constants and symbols.

Advantage over Neural Model. LTN offers the prospect of expressing knowledge using logical axioms over data and thus provides better explainability, data efficiency, and out-of-distribution generalization capability over neural networks.

D. Neuro-Vector-Symbolic Architecture (NVSA)

NVSA is a neuro-symbolic architecture that advances fluid intelligence and abstract reasoning capability assessed by Raven's progressive matrices (RPM) [8]. NVSA synergistically combines neural network visual perception and vector-symbolic probabilistic reasoning to facilitate a differentiable and computationally efficient abduction reasoning process.

Algorithm Description. NVSA enables reasoning generalization by exploiting the powerful operators on holographic distributed representations that synergistically combine neural and symbolic to co-design visual perception and probabilistic reasoning. The neural frontend consists of a neural network and a codebook to construct perceptual representations where it transduces visual sensory to fixed-width vector-symbolic representations and maintains perceptual uncertainty. The symbolic backend consists of probabilistic scene inference, symbolic rule reasoning, and rule execution where it maps the inferred probability into vector space to substitute the exhaustive probability computations into algebraic operations.

Advantage over Neural Model. Neural models suffer from the binding problem and superposition catastrophe that prevents them from providing an adequate description of objects or situations that can be represented by hierarchical and nested compositional structures [8]. NVSA bypasses this problem and exhibits superior accuracy over all neural methods and even

human performance on RPM tests which has been a widely used assessment of fluid intelligence and abstract reasoning.

E. Neural Logic Machine (NLM)

NLM is a neuro-symbolic architecture for both inductive learning and logical reasoning [34]. NLM exploits the power of neural networks as function approximators and logic programming as symbolic processors for objects with properties, relations, logic connectives, and quantifiers.

Algorithm Description. NLM is a neural realization of logic machines with the key intuition that logic operations such as logical `AND`s and `OR`s can be efficiently approximated by neural networks and the wiring among neural modules can realize the logic quantifiers. NLM encompasses a multi-layer multi-group architecture that takes object properties and relations as input tensors, performs sequential logic deduction computations, and outputs conclusive properties or relations of the objects. As the number of layers increases, NLM is able to form higher levels of abstraction.

Advantage over Neural Model. NLM exhibits perfect generalization capability in relational reasoning and decision-making compared to neural approaches. After being trained on small-scale tasks (e.g., sorting short arrays), NLMs can recover lifted rules and generalize to large-scale tasks (e.g., sorting longer arrays). Most of these tasks are hard to accomplish for neural networks (such as memory networks [59], graph neural networks [60]) or inductive logic programming [29] alone.

F. Vector Symbolic Architecture-Based Image-to-Image Translation (VSAIT)

VSAIT is a neuro-symbolic architecture that can effectively address semantic flipping issues when the distribution gap (shift in semantic statistics) between source and target domains is large [11]. VSAIT exploits the vector-symbolic architecture (VSA) to ensure photorealism in computer graphics applications and learn downstream tasks using translated images.

Algorithm Description. VSAIT addresses semantic flipping by learning an invertible mapping in a holographic vector space to ensure consistency between source and translated images. VSAIT extracts features and uses locality-sensitive hashing with a neural network to encode source, target, and translated images into the random vector-symbolic hyperspace. VSAIT learns to generate images with hypervectors similar to those in the target domain, unbinds source information (e.g., texture and color), and binds target information as well as vice versa to recover source content.

Advantage over Neural Model. Neural models still suffer from significant artifacts and hallucinations related to semantic flipping, while VSAIT ensures robustness to semantic flipping and significantly reduces image hallucinations observed for unpaired image translation between domains with large gaps.

G. Zero-Shot Concept Recognition and Acquisition (ZeroC)

ZeroC is a neuro-symbolic architecture that can recognize and acquire novel visual concepts in a zero-shot manner [33]. ZeroC exploits the symbolic graph structure to acquire model concepts and relations and apply them to cross-domain classification and detection tasks at inference time.

Algorithm Description. The key components of ZeroC are concepts and relations. Each concept consists of a graph and an energy-based model. The concept graph describes the concept as a composition of its constituent concepts and relations, and the concept energy-based model recognizes the concept in the input data. Each relation is also represented with a graph and an energy-based model, where the relation graph is an edge that connects the two related concepts, and the hierarchical concept is composed of constituent concepts as nodes and relations as edges according to a graph structure. During zero-shot concept recognition and acquisition in inference, the new hierarchical concept models are derived from the graph of the new hierarchical concept and energy-based models of their constituent concepts and relations. Concepts and relations can be viewed as templates for objects and their connections, which then get grounded during inference with specific images, where those objects and relations are assigned actual values.

Advantage over Neural Model. ZeroC exhibits remarkable zero-shot concept recognition and acquisition capability, which is still beyond the reach of neural models that require many examples (as in typical supervised learning) or many tasks drawn from the same distribution (as in few-shot learning) to learn a novel concept. ZeroC is able to transfer hierarchical concepts across different domains at inference, unlocking potential applications in more diverse tasks, such as AI for scientific discovery and composable neural systems.

H. Probabilistic Abduction and Execution (PrAE) Learner

PrAE is a neuro-symbolic learner for spatial-temporal cognitive reasoning tasks serving as an indicator of human fluid intelligence [26]; central to the PrAE learner is the process of probabilistic abduction and execution on a probabilistic scene representation, akin to the mental manipulation of objects.

Algorithm Description. PrAE learner consists of neural visual perception and symbolic logical reasoning. The neural visual frontend operates on object-based representation and predicts conditional probability distributions on its attributes. A scene inference engine then aggregates all object attribute distributions to produce a probabilistic scene representation. The symbolic logical backend abduces, from the representation, hidden rules that govern the time-ordered sequence via inverse dynamics. An execution engine executes the rules to generate the representation in a probabilistic planning manner. PrAE learner system is trained end-to-end in an analysis-by-synthesis manner without any visual attribute annotations.

Advantage over Neural Model. PrAE learner exhibits superior capability in spatial-temporal cognitive reasoning and fluid intelligence than neural models. Additionally, the PrAE learner offers human-level systematic generalizability, as well as transparency and interpretability to incorporate knowledge, which is hard to achieve with classic deep models.

IV. Workload Characterization Methodology

This section presents our neuro-symbolic AI workload profiling methodology (Sec. IV-A) and operator characterization taxonomy (Sec. IV-B) that will be leveraged in Sec. V.

A. Workload Profiling Methodology

We first conduct function-level profiling to capture statistics such as runtime, memory, invocation counts, tensor sizes, and sparsity of each model, by leveraging the built-in PyTorch Profiler [61]. We also perform post-processing to partition the characterization results into various operation categories. The experiments are conducted on a system with Intel Xeon Silver 4114 CPU and Nvidia RTX 2080 Ti GPU (250W), as well as edge SoCs such as Xavier NX (20W) and Jetson TX2 (15W).

B. Workload Characterization Taxonomy

On top of function-level profiling, we further conduct compute operator-level profiling for further analysis. We classify each neural and symbolic workload of the LNN, LTN, NVSA, NLM, VSAIT, ZeroC, and PrAE neuro-symbolic models into six operator categories: convolution, matrix multiplication (MatMul), vector/element-wise tensor operation, data transformation, data movement, and others [4].

Convolution: refers to operations involving overlaying a matrix (kernel) onto another matrix (input) and computing the sum of element-wise products. This process is slid across the entire matrix and transforms the data. Convolution is common in neural networks and leads to high operational intensity.

Matrix Multiplication: refers to general matrix multiplication (GEMM) with two matrices, either dense or sparse. Fully-connected layers in neural networks use GEMM as their primary mathematical operation. Multiplication of large, dense matrices is typically computationally intensive but highly parallelizable. There is typically a trade-off between the generality of the sparsity and the overhead of hardware optimization. Sparse matrix multiplication requires efficient mechanisms to perform lookups into the tables of non-zero values.

Vector/Element-wise Tensor Operation: refers to operations performed element-wise on tensors (generalized matrices, vectors, and higher-dimensional arrays), including addition, subtraction, multiplication, and division, applied between two tensors element by element, as well as activation, normalization, and relational operations in neuron models.

Data Transformation: refers to operations that reshape or subsample data, including matrix transposes, tensor reordering, masked selection, and coalescing which is a process in which duplicate entries for the same coordinates in a sparse matrix are eliminated by summing their associated values.

Data Movement: refers to data transferring from memory-to-compute, host-to-device, and device-to-host, as well as operations such as tensor duplication and assignment.

Others: refers to operations such as fuzzy first-of-logic and logical rules that are leveraged in several symbolic AI workloads.

V. Workload Characterization Results

This section analyzes the performance characteristics of representative neuro-symbolic workloads and discusses their runtime latency and scalability (Sec. V-A), compute operators (Sec. V-B), memory usage and system bottleneck (Sec. V-C), operation graph (Sec. V-D), hardware utilization (Sec. V-E), and sparsity (Sec. V-F).

A. Compute Latency Analysis

End-to-end latency breakdown. We first characterize the end-to-end latency of representative neuro-symbolic AI workloads (Fig. 2). We can observe that (1) Compared to neural workloads, symbolic workloads are not negligible in computing latency and may become a system bottleneck. For example, the neural (symbolic) workloads account for 54.6% (45.4%), 48.0% (52.0%), 7.9% (92.1%), 39.4% (60.6%), 16.3% (83.7%), 73.2% (26.8%), and 19.5% (80.5%) runtime of LNN, LTN, NVSA, NLM, VSAIT, ZeroC, and PrAE models, respectively. Notably, the symbolic workload dominates the NVSA's runtime, predominately due to the sequential and computational-intensive rule detection during the involved reasoning procedure. (2) The real-time performance cannot be satisfied, e.g., RTX 2080Ti GPU takes 380 s and TX2 takes 7507 s for RPM task in NVSA. Even if more computing resources are available to reduce neural inference time, the significant overhead of vector-symbolic-based reasoning still prohibits real-time execution. (3) The symbolic operations may not be well accelerated by GPU. For example, symbolic counts for 92.1% of total NVSA inference time on RTX 2080Ti while its floating-point operations (FLOPS) count for only 19% of total FLOPS, indicating inefficient hardware computation.

Takeaway 1: *Neuro-symbolic AI models typically exhibit high latency compared to neural models, prohibiting them from real-time applications. Symbolic operations are typically processed inefficiently on off-the-shelf CPU/GPUs and may result in system bottlenecks.*

End-to-end latency scalability. We evaluate the end-to-end runtime across various task sizes and complexities, as shown in Fig. 2c of RPM task for NVSA. We can observe that (1) The neural vs. symbolic runtime proportion remains relatively stable across various task sizes. For example, when task size increases from 2×2 to 3×3, the symbolic runtime slightly changes from 91.59% to 87.35%. (2) The total runtime increases quadratically with task size evolving. For example, the total runtime increases 5.02× in the above case, indicating the potential scalability bottleneck of neuro-symbolic models.

Takeaway 2: *The neural and symbolic components runtime ratio remains relatively stable while total latency explodes with the cognitive reasoning task complexity evolving. The potential scalability bottleneck calls for highly scalable and efficient architecture.*

Recommendation 1: *Optimization on neuro-symbolic workloads from algorithm-system-hardware-technology cross-layer perspectives is highly desirable for achieving real-time, efficient and scalable cognitive systems.*

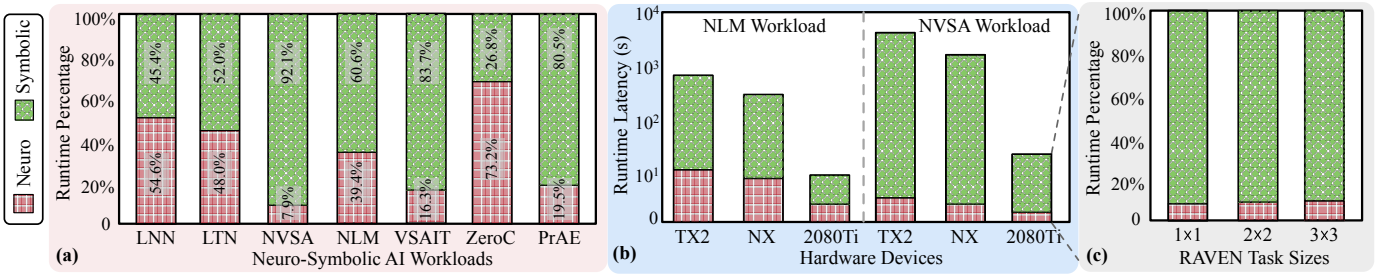


Fig. 2: Neural and symbolic runtime latency characterization. (a) Benchmark seven representative neuro-symbolic workloads (LNN, LTN, NVSA, NLM, VSAIT, ZeroC, PrAE) on the CPU+GPU system, showing symbolic may serve as system bottleneck. **(b)** Benchmark NVSA and NLM workloads on Jetson TX2, Xavier NX, and RTX GPU, showing that real-time performance cannot be satisfied. **(c)** Benchmark NVSA workload on various RPM task sizes on RTX GPU, indicating the potential scalability problem and consistent symbolic bottleneck.

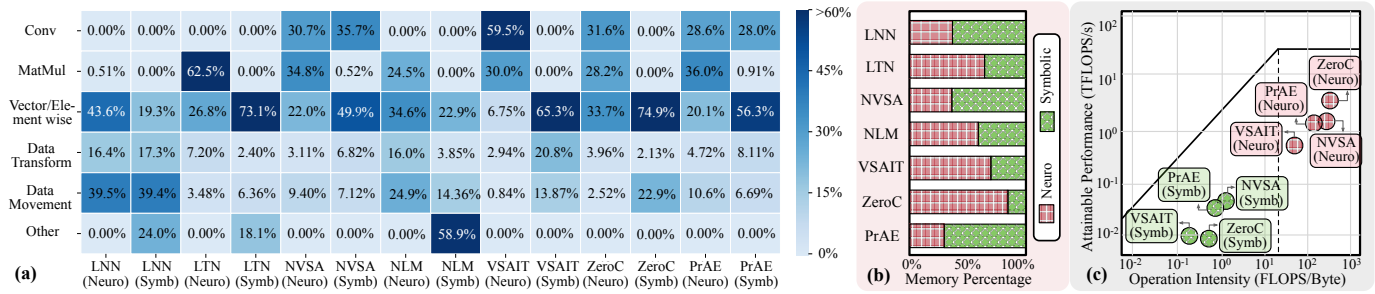


Fig. 3: Compute operators, memory and roofline characterization. (a) Compute operator runtime ratio of representative neuro-symbolic workloads, indicating neural operations mainly consisting of MatMul and Conv, while symbolic operations with vector/tensors. **(b)** Benchmark memory usage during computation and **(c)** roofline analysis on RTX 2080Ti GPU, showing typically neural operations are compute-bounded and symbolic operations are memory-bounded.

B. Compute Operator Analysis

Fig. 3a partitions the neural and symbolic workloads of the LNN, LTN, NVSA, NLM, VSAIT, ZeroC, and PrAE workloads into six operator categories (Sec. IV-B) with runtime latency breakdown. We make the following observations:

Neural Workload Analysis. The neural workload is dominated by the MatMul and activation operations. LTN (neuro) is dominated by MatMul due to its heavy MLP components, while NVSA, VSAIT, and PrAE's (neuro) majority runtime is on MatMul and convolution because they adopt the neural network as the perception backbone for feature extraction. By contrast, a large portion of LNN and NLM's (neuro) runtime is on vector and element-wise tensor operations due to the sparse syntax tree structure composed of proposition logic and the sequential logic deduction computations on multi-group architecture. Notably, data movement also takes up a significant amount of LNN (neuro) runtime because of its unique bidirectional dataflow during reasoning inference.

Symbolic Workload Analysis. The symbolic workload is dominated by vector and scalar operations that exhibit low operational intensities and complex control flows. Both LNN, LTN, and NLM's (symbolic) have a large number of logic operations, posing parallelism optimization opportunities in their database queries and arithmetic operations, especially for larger symbolic models. Meanwhile, LNN (symbolic) is severally data movement-bounded due to its sparse and irregular memory accesses and bidirectional inference, where model-aware dataflow architecture would likely be beneficial for alleviating this bottleneck. NVSA, VSAIT, and PrAE's (symbolic) are composed of vectors for vector-symbolic operations. Notably, these operations usually stem from high-dimensional distributed vector computations (e.g., binding, bundling) for symbolic representation, which are difficult to process efficiently on GPUs. Therefore, the challenges of accelerating these computations will become increasingly important as the task and feature complexities further grow.

Takeaway 3: *The neural components mainly consist of MatMul and Convs, while the symbolic components are dominated by vector/element-wise tensor and logical operations which are computed inefficiently on GPUs. The data transfer overhead arising from the separate neural and symbolic execution on GPUs and CPUs poses efficient hardware design challenges.*

Recommendation 2: *From the architecture level, custom processing units can be built for efficient symbolic operations (e.g., high-dimensional distributed vectors, logical operation, graph, etc). For non-overlap neural and symbolic components, reconfigurable processing units supporting both neural and symbolic operations are recommended.*

C. Memory and System Analysis

Memory Usage Analysis. Fig. 3b characterizes the memory usage of the LNN, LTN, NVSA, NLM, VSAIT, ZeroC, and PrAE workloads during computation. We can observe that (1) PrAE (symbolic) consumes a high ratio of memory due to its

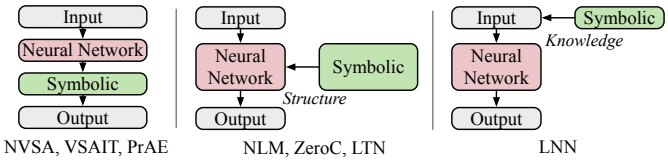


Fig. 4: Operator graph analysis. Symbolic operation depends on neural results or needs to compile in neural structure as the critical path. Complex control and symbolic-only phase operation result in inefficiency and low hardware resource utilization.

large number of vector operations depending on intermediate results and exhaustive symbolic search. NVSA (symbolic) slightly alleviates the vector-symbolic operation memory by leveraging probabilistic abduction reasoning. ZeroC (neuro) contains energy-based models and process images in a large ensemble thus taking much memory. (2) In terms of storage footprint, neural weights and symbolic codebooks typically consume more storage. For example, neural network and holographic vector-inspired codebook account for >90% memory footprint in NVSA, because NVSA neural frontend enables the expression of more object combinations than vector space dimensions, requiring the codebook to be large enough to contain all object combinations and ensure quasi-orthogonality.

System Roofline Analysis. Fig. 3c employs the roofline model to quantify the memory boundedness of RTX 2080Ti GPU versions of the selected workloads. We observe that the symbolic components are in the memory-bound area while neural components are in the compute-bound area. For example, NVSA and PrAE symbolic operations require streaming vector elements to circular convolution computing units, increasing the memory bandwidth pressure. Optimizing the compute dataflow and leveraging the scalable and reconfigurable processing element can help provide this bandwidth.

Takeaway 4: *The symbolic operations are memory-bounded due to large element streaming for vector-symbolic operations. The neural operations are compute-bounded due to computational-intensive MatMul and Convs. Neural weights and vector-inspired codebooks typically account for most memory storage while the symbolic components require large intermediate caching during computation.*

Recommendation 3: *From the algorithm level, model compression (e.g., quantization and pruning) and efficient factorization of neural and symbolic components can be used to reduce memory and data movement overhead without sacrificing cognitive reasoning accuracy.*

Recommendation 4: *From the technology level, emerging memory technology and compute-in-memory technique can alleviate the memory-bounded symbolic operations and improve scalability, performance, area, and energy efficiency of neuro-symbolic systems.*

D. Operation and Dataflow

Fig. 4 analyzes the operation dependency in representative neuro-symbolic workloads. We can observe that the reasoning computation of NVSA, VSAIT, and PrAE depends on the result of the frontend neural workload and thus lies on the critical path during inference. LNN, LTN, NLM, and ZeroC need to compile the symbolic knowledge in neural representation or input embeddings. The complex control results in inefficiency in CPU and GPU, and the vector-symbolic computation period results in low hardware utilization. There are opportunities for data pre-processing, parallel rule query, and heterogeneous and reconfigurable hardware design to reduce this bottleneck.

TABLE IV: Hardware inefficiency analysis. The compute, memory, and communication characteristics of representative neural and symbolic kernels in NVSA workload executed on CPU/GPU platform.

	Neural Kernel		Symbolic Kernel	
	sgemm_nn	relu_nn	vectorized_elem	elementwise
Compute Throughput (%)	95.1	92.9	3.0	2.3
ALU Utilization (%)	90.1	48.3	5.9	4.5
L1 Cache Throughput (%)	79.7	82.6	28.4	10.8
L2 Cache Throughput (%)	19.2	17.5	29.8	22.8
L1 Cache Hit Rate (%)	1.6	51.6	29.5	33.3
L2 Cache Hit Rate (%)	86.8	65.5	48.6	34.3
DRAM BW Utilization (%)	14.9	24.2	90.9	78.4

Takeaway 5: *The symbolic operations depend on the neural module results or need to compile into the neural structure, thus lying on the critical path of end-to-end neuro-symbolic systems. The vector-symbolic computation phase and complex control of neuro-symbolic components bring low hardware resource utilization and inefficiency in CPU/GPU.*

Recommendation 5: *From the system level, adaptive workload scheduling with parallelism processing of neural and symbolic components can be leveraged to alleviate resource underutilization and improve runtime efficiency.*

E. Hardware Inefficiency Analysis

The hardware inefficiencies of executing neuro-symbolic workloads mainly come from ALU underutilization, low cache hit rate, and massive data transfer. We leverage Nsight Systems and Nsight Compute tools [62], [63] to further characterize the GPU behavior of executing selected neuro-symbolic workloads. Tab. IV lists the compute, memory, and data movement characteristics of representative neural and symbolic kernels in NVSA as an example. We observe that typically in symbolic operations, the GPU ALU unit utilization is <10%, the L1 cache hit rate is around 20%, the L2 cache hit rate is around 40%, and DRAM bandwidth utilization is around 90% with several memory-bounded. The data transfer memory operations account for around 50% of total latency, where >80% is from host CPU to GPU. Additionally, the synchronization overhead and waiting for GPU operations to complete results in CPU underutilization.

Takeaway 6: *While the neural kernels exhibit high compute utilization and memory efficiency in GPUs, the symbolic operations typically suffer from low ALU utilization, low L1 cache hit rates, and high memory transactions, which results in low system efficiency.*

Recommendation 6: *From the architecture level, heterogeneous or reconfigurable neural/symbolic architecture with efficient vector-symbolic units and high-bandwidth NoC can be optimized to improve ALU utilization and reduce data movement, thus improving system performance.*

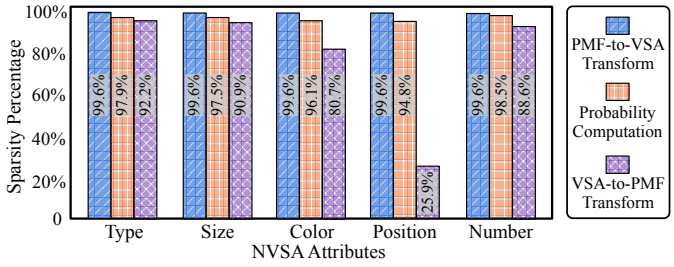


Fig. 5: Sparsity analysis. The Sparsity ratio of NVSA symbolic operations, shows a high degree of sparsity with variations in attributes.

F. Sparsity Analysis

Neuro-symbolic workloads also exhibit sparsity features. For example, Fig. 5 characterizes the sparsity of NVSA symbolic modules, including probabilistic mass function (PMF)-to-VSA transform, probability computation, and VSA-to-PMF transform, under different reasoning rule attributes. We can observe that NVSA has a high sparsity ratio (>95%) with variations for specific attributes and unstructured patterns. Similarly, ZeroC and LNN also demonstrate >90% sparsity ratio, while LTN features a dense computation pattern.

Takeaway 7: *Some neural and vector-symbolic components demonstrate a high level of unstructured sparsity with variations under different task scenarios and attributes.*

Recommendation 7: *From the algorithm and architecture level, sparsity-aware neural and symbolic algorithm and architecture design can benefit memory footprint, communication overhead, and computation FLOPS reduction.*

G. Uniqueness of Neuro-Symbolic vs. Neural Networks

To summarize, based on the above workload characterization, neuro-symbolic AI workloads differ from neural networks mainly in three aspects:

Compute kernels. Neuro-symbolic workloads consist of heterogeneous neural and symbolic kernels. The symbolic operators (e.g., vector, graph, logic) are processed inefficiently on off-the-shelf CPUs/GPUs with low hardware utilization and cache hit and may result in runtime latency bottleneck.

Memory. Symbolic operations are memory-bounded due to large element streaming for vector-symbolic operations. Symbolic codebooks typically account for large memory footprints and require large intermediate caching during computation.

Dataflow and scalability. Neuro-symbolic workloads exhibit more complex control than NNs. Symbolic operations either critically depend on the neural stage or need to compile in neural structure. Their irregular dataflow, data dependency, and sequential processing bring low parallelism scalability and inefficiency in CPU/GPU.

VI. Outlook and Research Opportunities

In this section, we discuss the challenges and opportunities for neuro-symbolic systems, and outline our vision for the future, focusing on the system and architecture perspectives.

Building ImageNet-like neuro-symbolic datasets. Neuro-symbolic systems hold great potential in achieving human-like performance [64]. However, their current applications are still limited to basic decision-making and reasoning problems [65], falling short of the broader vision of human cognitive abilities, such as deductive reasoning, compositionality, and counterfactual thinking. It is still an open question of how perception learned from other domains can be transferred to abstract reasoning tasks [26]. To significantly advance the metacognitive capabilities of neuro-symbolic systems, more challenging and suitable datasets are highly desirable to unleash its potential.

Unifying neuro-symbolic models. Integrating neural, symbolic, and probabilistic approaches offers promise to improve AI models' explainability and robustness. However, the current attempts to combine these complementary approaches are still in a nascent manner [66] - how to integrate them in a principled manner remains a fundamental and open challenge. We envision a unified framework to design algorithms that opportunistically combine neural and symbolic with probabilistic representations, and for quantifying scaling laws for neuro-symbolic inference versus large neural models.

Developing efficient software frameworks. Neuro-symbolic AI systems typically utilize underlying logic, such as fuzzy logic, parameterization, and differentiable structures, to support learning and reasoning capabilities. However, most system implementations create custom software for deduction for the particular logic, which limits modularity and extensibility [67]. Thus, new software frameworks are needed that can encompass a broad set of reasoning logical capabilities and provide practical syntactic and semantic extensions while being fast and memory-efficient. Moreover, new programming models and compilers that can facilitate the ease and efficient realization of the neuro-symbolic models are of significance to realize the full promise of neuro-symbolic AI paradigms.

Benchmarking diverse neuro-symbolic workloads. Given the proliferation of neuro-symbolic algorithms and the rapid hardware advancements, it is crucial to benchmark neuro-symbolic AI systems in a comparable and validated manner. To achieve this, from the system aspect, we need representative benchmarks that capture the essential workload characteristics (e.g., compute kernels, access patterns, and sparsity) of neural and symbolic models, and that can be quantitatively tested in human-AI applications. From an architectural and hardware perspective, we need modeling-simulation frameworks to enable the development of novel architectures for these workloads and build optimized modular blocks as libraries by leveraging workload characteristics. Benchmarking neuro-symbolic computing will guide ML researchers and system architects in investigating the trade-offs in accuracy, performance, and efficiency of various neuro-symbolic algorithms, and in implementing systems in a performance-portable way.

Designing cognitive hardware architectures. Neuro-symbolic workloads that combine neural, symbolic, and probabilistic methods feature much greater heterogeneity in compute kernels, sparsity, irregularity in access patterns, and higher memory intensity than DNNs. This leads to an increasing divergence with the current hardware roadmap that largely focuses on matrix multiplication and regular dataflow. There-

fore, we need novel architectures with dedicated processing units, memory hierarchies, and NoCs that can handle the additional complexities in computations and communications. Additionally, the architecture needs to provide flexibility with both configurable interconnects and full addressable memories to keep pace with neuro-symbolic AI algorithmic innovations.

VII. Related Work

Neural Network Characterization. Over the past years, computer and system architects have proposed a vast array of benchmarks, simulators, and custom architectures, most notably for characterizing and accelerating machine learning, specifically DNN training and inference across a variety of use cases ranging from mobile and edge [68]–[74] to large-scale cloud systems [75]–[78]. *However, neuro-symbolic systems exhibit divergent compute and memory features than DNNs and make current deep learning architecture inefficient, this paper thus takes the first step to characterize neuro-symbolic workloads to enable their efficient and scalable execution.*

Emerging Workload Characterization. Beyond DNNs, benchmarks and hardware accelerators have been explored in other applications like mixed-reality [79], [80], neuromorphic AI [81]–[83], robotics [84]–[90], genome sequencing [91], [92], graph [93], mobile vision [94], [95], fully homomorphic execution [96], [97], hyperdimensional computing [98]–[100], etc. *Neuro-symbolic AI shows the promising potential to be integrated into these systems and enable trustworthy applications. It is thus highly desirable to understand and optimize its system characteristics and performance.*

VIII. Conclusion

Neuro-symbolic AI is an emerging paradigm for next-generation efficient, robust, explainable, and cognitive AI systems. This paper systematically characterizes neuro-symbolic system performance, analyzes their workload operators, proposes optimization techniques for their performance and efficiency, and identifies the challenges and opportunities towards fulfilling next-generation neuro-symbolic AI systems.

Acknowledgements

We thank Mohamed Ibrahim, Sixu Li, and Yang (Katie) Zhao for the helpful discussions. This work was supported in part by CoCoSys, one of seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

References

[1] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. Aga, J. Huang, C. Bai, *et al.*, "Sustainable ai: Environmental implications, challenges and opportunities," *Proceedings of Machine Learning and Systems (MLSys)*, vol. 4, pp. 795–813, 2022.

[2] Z. Wan, A. Anwar, Y.-S. Hsiao, T. Jia, V. J. Reddi, and A. Raychowdhury, "Analyzing and improving fault tolerance of learning-based navigation systems," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pp. 841–846, IEEE, 2021.

[3] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, *et al.*, "Explainable ai (xai): Core ideas, techniques, and solutions," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–33, 2023.

[4] Z. Susskind, B. Arden, L. K. John, P. Stockton, and E. B. John, "Neuro-symbolic ai: An emerging class of ai workloads and their characterization," *arXiv preprint arXiv:2109.06133*, 2021.

[5] A. d. Garcez and L. C. Lamb, "Neurosymbolic ai: The 3 rd wave," *Artificial Intelligence Review*, pp. 1–20, 2023.

[6] X. Yang, Z. Wang, X. S. Hu, C. H. Kim, S. Yu, M. Pajic, R. Manohar, Y. Chen, and H. H. Li, "Neuro-symbolic computing: Advancements and challenges in hardware-software co-design," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2023.

[7] Z. Wan, C.-K. Liu, H. Yang, C. Li, H. You, Y. Fu, C. Wan, T. Krishna, Y. Lin, and A. Raychowdhury, "Towards cognitive ai systems: a survey and prospective on neuro-symbolic ai," *arXiv preprint arXiv:2401.01040*, 2024.

[8] M. Hersche, M. Zeqiri, L. Benini, A. Sebastian, and A. Rahimi, "A neuro-vector-symbolic architecture for solving raven's progressive matrices," *Nature Machine Intelligence*, pp. 1–13, 2023.

[9] 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," in *International Conference on Learning Representations (ICLR)*, 2019.

[10] Z. Yang, A. Ishay, and J. Lee, "Neurasp: Embracing neural networks into answer set programming," in *29th International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

[11] J. Theiss, J. Leverett, D. Kim, and A. Prakash, "Unpaired image translation via vector symbolic architectures," in *European Conference on Computer Vision (ECCV)*, pp. 17–32, Springer, 2022.

[12] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv preprint arXiv:2309.16650*, 2023.

[13] A. J. Scotte and V. De Silva, "Towards a neuro-symbolic framework for multimodal human-ai interaction.," in *ICPRAM*, pp. 932–939, 2023.

[14] J. Hsu, J. Mao, and J. Wu, "Ns3d: Neuro-symbolic grounding of 3d objects and relations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2614–2623, 2023.

[15] E. Qin, A. Samajdar, H. Kwon, V. Nadella, S. Srinivasan, D. Das, B. Kaul, and T. Krishna, "Sigma: A sparse and irregular gemm accelerator with flexible interconnects for dnn training," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 58–70, IEEE, 2020.

[16] A. Samajdar, J. M. Joseph, Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna, "A systematic methodology for characterizing scalability of dnn accelerators using scale-sim," in *2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 58–68, IEEE, 2020.

[17] H. Genc, S. Kim, A. Amid, A. Haj-Ali, V. Iyer, P. Prakash, J. Zhao, D. Grubb, H. Liew, H. Mao, *et al.*, "Gemmini: Enabling systematic deep-learning architecture evaluation via full-stack integration," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pp. 769–774, IEEE, 2021.

[18] H. Kwon, L. Lai, M. Pellauer, T. Krishna, Y.-H. Chen, and V. Chandra, "Heterogeneous dataflow accelerators for multi-dnn workloads," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 71–83, IEEE, 2021.

[19] C. Guo, J. Tang, W. Hu, J. Leng, C. Zhang, F. Yang, Y. Liu, M. Guo, and Y. Zhu, "Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization," in *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA)*, pp. 1–15, 2023.

[20] Y. N. Wu, P.-A. Tsai, S. Muralidharan, A. Parashar, V. Sze, and J. Emer, "Highlight: Efficient and flexible dnn acceleration with hierarchical structured sparsity," in *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1106–1120, 2023.

[21] F. Muñoz-Martínez, R. Garg, M. Pellauer, J. L. Abellán, M. E. Acacio, and T. Krishna, "Flexagon: A multi-dataflow sparse-sparse matrix multiplication accelerator for efficient dnn processing," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 252–265, 2023.

[22] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "Mastering

chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.
[23] C. Pryor, C. Dickens, E. Augustine, A. Albalak, W. Wang, and L. Getoor, "Neupsl: Neural probabilistic soft logic," *arXiv preprint arXiv:2205.14268*, 2022.
[24] W.-Z. Dai, Q. Xu, Y. Yu, and Z.-H. Zhou, "Bridging machine learning and logical reasoning by abductive learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.
[25] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.
[26] C. Zhang, B. Jia, S.-C. Zhu, and Y. Zhu, "Abstract spatial-temporal reasoning via probabilistic abduction and execution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9736–9746, 2021.
[27] R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I. Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma, *et al.*, "Logical neural networks," *arXiv preprint arXiv:2006.13155*, 2020.
[28] G. Lample and F. Charton, "Deep learning for symbolic mathematics," in *International Conference on Learning Representations (ICLR)*, 2019.
[29] R. Evans and E. Grefenstette, "Learning explanatory rules from noisy data," *Journal of Artificial Intelligence Research*, vol. 61, pp. 1–64, 2018.
[30] S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022.
[31] P. Hohenecker and T. Lukas, "Ontology reasoning with deep neural networks," *Journal of Artificial Intelligence Research*, vol. 68, pp. 503–540, 2020.
[32] L. C. Lamb, A. Garcez, M. Gori, M. Prates, P. Avelar, and M. Vardi, "Graph neural networks meet neural-symbolic computing: A survey and perspective," in *IJCAI-PRICAI 2020-29th International Joint Conference on Artificial Intelligence-Pacific Rim International Conference on Artificial Intelligence*, 2020.
[33] T. Wu, M. Tjandrasuwita, Z. Wu, X. Yang, K. Liu, R. Sosic, and J. Leskovec, "Zeroc: A neuro-symbolic model for zero-shot concept recognition and acquisition at inference time," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 9828–9840, 2022.
[34] H. Dong, J. Mao, T. Lin, C. Wang, L. Li, and D. Zhou, "Neural logic machines," in *International Conference on Learning Representations (ICLR)*, 2019.
[35] H. Kaut, "Robert s. engelmore memorial lecture at aaai 2020," *https://roc-hci.com/announcements/the-third-ai-summer/*, 2020.
[36] H. Zhang and T. Yu, "Alphazero," *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pp. 391–415, 2020.
[37] C. Zhang, F. Gao, B. Jia, Y. Zhu, and S.-C. Zhu, "Raven: A dataset for relational and analogical visual reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5317–5327, 2019.
[38] S. Hu, Y. Ma, X. Liu, Y. Wei, and S. Bai, "Stratified rule-aware network for abstract visual reasoning," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 35, pp. 1567–1574, 2021.
[39] R. Manhaeve, S. Dumančić, A. Kimmig, T. Demeester, and L. De Raedt, "Neural probabilistic logic programming in deepproblog," *Artificial Intelligence*, vol. 298, p. 103504, 2021.
[40] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, "Clevrer: Collision events for video representation and reasoning," in *International Conference on Learning Representations (ICLR)*, 2020.
[41] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, "Clevrer: Collision events for video representation and reasoning," in *International Conference on Learning Representations (ICLR)*, 2019.
[42] P. Sen, B. W. de Carvalho, R. Riegel, and A. Gray, "Neuro-symbolic inductive logic programming with logical neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 36, pp. 8212–8219, 2022.
[43] A. d. Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran, "Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning," *arXiv preprint arXiv:1905.06088*, 2019.
[44] Y. Guo, Z. Pan, and J. Heflin, "Lubm: A benchmark for owl knowledge base systems," *Journal of Web Semantics*, vol. 3, no. 2-3, pp. 158–182, 2005.
[45] G. Sutcliffe, "The tptp problem library and associated infrastructure," *Journal of Automated Reasoning*, vol. 59, no. 4, pp. 483–502, 2017.
[46] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.
[47] Ö. GENCER, "The research about morphometric characteristics on leptograpsus crabs," 2023.
[48] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, "Deepproblog: Neural probabilistic logic programming," *Advances in neural information processing systems (NeurIPS)*, vol. 31, 2018.
[49] D. Barrett, F. Hill, A. Santoro, A. Morcos, and T. Lillicrap, "Measuring abstract reasoning in neural networks," in *International conference on machine learning (ICML)*, pp. 511–520, PMLR, 2018.
[50] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.
[51] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision (ECCV)*, pp. 102–118, Springer, 2016.
[52] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 3213–3223, 2016.
[53] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1125–1134, 2017.
[54] F. Chollet, "On the measure of intelligence," *arXiv preprint arXiv:1911.01547*, 2019.
[55] M. Shanahan, K. Nikiforou, A. Creswell, C. Kaplanis, D. Barrett, and M. Garnelo, "An explicitly relational neural network architecture," in *International Conference on Machine Learning (ICML)*, pp. 8593–8603, PMLR, 2020.
[56] I. Donadello, L. Serafini, and A. D. Garcez, "Logic tensor networks for semantic image interpretation," *arXiv preprint arXiv:1705.08968*, 2017.
[57] F. Bianchi and P. Hitzler, "On the capabilities of logic tensor networks for deductive reasoning.," in *AAAI Spring symposium: combining machine learning with knowledge engineering*, 2019.
[58] S. Badreddine and M. Spranger, "Injecting prior knowledge for transfer learning into reinforcement learning algorithms using logic tensor networks," *arXiv preprint arXiv:1906.06576*, 2019.
[59] S. Sukhbaatar, J. Weston, R. Fergus, *et al.*, "End-to-end memory networks," *Advances in neural information processing systems (NeurIPS)*, vol. 28, 2015.
[60] V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," *Journal of Machine Learning Research (JMLR)*, vol. 24, no. 43, pp. 1–48, 2023.
[61] Maxim Lukiyanov, Guoliang Hua, Geeta Chauhan, and Gisle Dankel, "Introducing PyTorch Profiler - the new and improved performance tool," 2021. https://pytorch.org/blog/introducing-pytorch-profiler-the-new-and-improved-performance-tool/, accessed 2021-05-21.
[62] "NVIDIA Nsight Systems." https://developer.nvidia.com/nsight-systems.
[63] "NVIDIA Nsight Compute." https://developer.nvidia.com/nsight-compute.
[64] G. Booch, F. Fabiano, L. Horesh, K. Kate, J. Lenchner, N. Linck, A. Loreggia, K. Murgesan, N. Mattei, F. Rossi, *et al.*, "Thinking fast and slow in ai," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 35, pp. 15042–15046, 2021.
[65] A. d. Garcez, S. Bader, H. Bowman, L. C. Lamb, L. de Penning, B. Illuminoo, H. Poon, and C. G. Zaverucha, "Neural-symbolic learning and reasoning: a survey and interpretation," *Neuro-Symbolic Artificial Intelligence: The State of the Art*, vol. 342, no. 1, 2022.
[66] W. Wang and Y. Yang, "Towards data-and knowledge-driven artificial intelligence: A survey on neuro-symbolic computing," *arXiv preprint arXiv:2210.15889*, 2022.
[67] D. Aditya, K. Mukherji, S. Balasubramanian, A. Chaudhary, and P. Shakarian, "Pyreason: Software for open world temporal logic," *arXiv preprint arXiv:2302.13482*, 2023.
[68] A. Samajdar, P. Mannan, K. Garg, and T. Krishna, "Genesys: Enabling continuous learning through neural network evolution in hardware," in

2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 855–866, IEEE, 2018.

[69] C. Banbury, V. J. Reddi, P. Torelli, J. Holleman, N. Jeffries, C. Kiraly, P. Montino, D. Kanter, S. Ahmed, D. Pau, *et al.*, "Mlperf tiny benchmark," *arXiv preprint arXiv:2106.07597*, 2021.

[70] V. Janapa Reddi, D. Kanter, P. Mattson, J. Duke, T. Nguyen, R. Chukka, K. Shiring, K.-S. Tan, M. Charlebois, W. Chou, *et al.*, "Mlperf mobile inference benchmark: An industry-standard open-source machine learning benchmark for on-device ai," *Proceedings of Machine Learning and Systems (MLSys)*, vol. 4, pp. 352–369, 2022.

[71] C. Zhou, F. G. Redondo, J. Büchel, I. Boybat, X. T. Comas, S. Nandakumar, S. Das, A. Sebastian, M. Le Gallo, and P. N. Whatmough, "Ml-hw co-design of noise-robust tinyml models and always-on analog compute-in-memory edge accelerator," *IEEE Micro*, vol. 42, no. 6, pp. 76–87, 2022.

[72] T. Tambe, E.-Y. Yang, G. G. Ko, Y. Chai, C. Hooper, M. Donato, P. N. Whatmough, A. M. Rush, D. Brooks, and G.-Y. Wei, "A 16-nm soc for noise-robust speech and nlp edge ai inference with bayesian sound source separation and attention-based dnns," *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 58, no. 2, pp. 569–581, 2022.

[73] T. Tambe, J. Zhang, C. Hooper, T. Jia, P. N. Whatmough, J. Zuckerman, M. C. Dos Santos, E. J. Loscalzo, D. Giri, K. Shepard, *et al.*, "22.9 a 12nm 18.1 tflops/w sparse transformer processor with entropy-based early exit, mixed-precision predication and fine-grained power management," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 342–344, IEEE, 2023.

[74] A. Ramachandran, Z. Wan, G. Jeong, J. Gustafson, and T. Krishna, "Algorithm-hardware co-design of distribution-aware logarithmic-posit encodings for efficient dnn inference," *arXiv preprint arXiv:2403.05465*, 2024.

[75] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, *et al.*, "Mlperf inference benchmark," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pp. 446–459, IEEE, 2020.

[76] P. Mattson, C. Cheng, G. Diamos, C. Coleman, P. Micikevicius, D. Patterson, H. Tang, G.-Y. Wei, P. Bailis, V. Bittorf, *et al.*, "Mlperf training benchmark," *Proceedings of Machine Learning and Systems (MLSys)*, vol. 2, pp. 336–349, 2020.

[77] S. Rashidi, S. Sridharan, S. Srinivasan, and T. Krishna, "Astra-sim: Enabling sw/hw co-design exploration for distributed dl training platforms," in *2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 81–92, IEEE, 2020.

[78] W. Won, T. Heo, S. Rashidi, S. Sridharan, S. Srinivasan, and T. Krishna, "Astra-sim2. 0: Modeling hierarchical networks and disaggregated systems for large-model training at scale," in *2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 283–294, IEEE, 2023.

[79] H. Kwon, K. Nair, J. Seo, J. Yik, D. Mohapatra, D. Zhan, J. Song, P. Capak, P. Zhang, P. Vajda, *et al.*, "Xrbench: An extended reality (xr) machine learning benchmark suite for the metaverse," *Proceedings of Machine Learning and Systems (MLSys)*, vol. 5, 2023.

[80] N. Li, M. Chang, and A. Raychowdhury, "E-gaze: Gaze estimation with event camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[81] J. Yik, S. H. Ahmed, Z. Ahmed, B. Anderson, A. G. Andreou, C. Bartolozzi, A. Basu, D. d. Blanken, P. Bogdan, S. Bohte, *et al.*, "Neurobench: Advancing neuromorphic computing through collaborative, fair and representative benchmarking," *arXiv preprint arXiv:2304.04640*, 2023.

[82] M. Chang, A. S. Lele, S. D. Spetalnick, B. Crafton, S. Konno, Z. Wan, A. Bhat, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, *et al.*, "A 73.53 tops/w 14.74 tops heterogeneous rram in-memory and sram near-memory soc for hybrid frame and event-based target tracking," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 426–428, IEEE, 2023.

[83] D. Wu, J. Li, Z. Pan, Y. Kim, and J. S. Miguel, "ubrain: A unary brain computer interface," in *Proceedings of the 49th Annual International Symposium on Computer Architecture (ISCA)*, pp. 468–481, 2022.

[84] Z. Wan, Y. Zhang, A. Raychowdhury, B. Yu, Y. Zhang, and S. Liu, "An energy-efficient quad-camera visual system for autonomous machines on fpga platform," in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 1–4, IEEE, 2021.

[85] S. M. Neuman, B. Plancher, T. Bourgeat, T. Tambe, S. Devadas, and V. J. Reddi, "Robomorphic computing: a design methodology for domain-specific accelerators parameterized by robot morphology," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 674–686, 2021.

[86] S. Krishnan, Z. Wan, K. Bhardwaj, N. Jadhav, A. Faust, and V. J. Reddi, "Roofline model for uavs: A bottleneck analysis tool for onboard compute characterization of autonomous unmanned aerial vehicles," in *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 162–174, IEEE, 2022.

[87] Q. Liu, Z. Wan, B. Yu, W. Liu, S. Liu, and A. Raychowdhury, "An energy-efficient and runtime-reconfigurable fpga-based accelerator for robotic localization systems," in *2022 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 01–02, IEEE, 2022.

[88] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, S. Neuman, G.-Y. Wei, D. Brooks, and V. J. Reddi, "Automatic domain-specific soc design for autonomous unmanned aerial vehicles," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 300–317, IEEE, 2022.

[89] S. M. Neuman, R. Ghosal, T. Bourgeat, B. Plancher, and V. J. Reddi, "Roboshape: Using topology patterns to scalably and flexibly deploy accelerators across robots," in *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA)*, pp. 1–13, 2023.

[90] V. Mayoral-Vilches, J. Jabbour, Y.-S. Hsiao, Z. Wan, A. Martínez-Fariña, M. Crespo-Alvarez, M. Stewart, J. M. Reina-Munoz, P. Nagras, G. Vikhe, *et al.*, "Robotperf: An open-source, vendor-agnostic, benchmarking suite for evaluating robotics computing system performance," *arXiv preprint arXiv:2309.09212*, 2023.

[91] D. Fujiki, A. Subramaniyan, T. Zhang, Y. Zeng, R. Das, D. Blaauw, and S. Narayanasamy, "Genax: A genome sequencing accelerator," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 69–82, IEEE, 2018.

[92] D. Fujiki, S. Wu, N. Ozog, K. Goliya, D. Blaauw, S. Narayanasamy, and R. Das, "Seedex: A genome sequencing accelerator for optimal alignments in subminimal space," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 937–950, IEEE, 2020.

[93] C. Gao, M. Afarin, S. Rahman, N. Abu-Ghazaleh, and R. Gupta, "Mega evolving graph accelerator," in *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 310–323, 2023.

[94] Z.-G. Liu, P. N. Whatmough, and M. Mattina, "Systolic tensor array: An efficient structured-sparse gemm accelerator for mobile cnn inference," *IEEE Computer Architecture Letters*, vol. 19, no. 1, pp. 34–37, 2020.

[95] Z.-G. Liu, P. N. Whatmough, Y. Zhu, and M. Mattina, "S2ta: Exploiting structured sparsity for energy-efficient mobile cnn acceleration," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 573–586, IEEE, 2022.

[96] J. Ma, C. Xu, and L. W. Wills, "Pytfhe: An end-to-end compilation and execution framework for fully homomorphic encryption applications," in *2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 24–34, IEEE, 2023.

[97] M. Nabeel, D. Soni, M. Ashraf, M. A. Gebremichael, H. Gamil, E. Chielle, R. Karri, M. Sanduleanu, and M. Maniatakos, "Cofhee: A co-processor for fully homomorphic encryption execution," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–2, IEEE, 2023.

[98] K. Schlegel, P. Neubert, and P. Protzel, "A comparison of vector symbolic architectures," *Artificial Intelligence Review*, vol. 55, no. 6, pp. 4523–4555, 2022.

[99] Z. Wan, C.-K. Liu, M. Ibrahim, H. Yang, S. Spetalnick, T. Krishna, and A. Raychowdhury, "H3dfact: Heterogeneous 3d integrated cim for factorization with holographic perceptual representations," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, 2024.

[100] S. Shou, C.-K. Liu, S. Yun, Z. Wan, K. Ni, M. Imani, X. S. Hu, J. Yang, C. Zhuo, and X. Yin, "See-mcam: Scalable multi-bit fefet content addressable memories for energy efficient associative search," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–9, IEEE, 2023.