2026 IEEE International Symposium on Workload Characterization (IISWC) | 979-8-3195-4283-0/26/$31.00 ©2026 IEEE | DOI: 10.1109/IISWC71020.2026.00035

TESSERA: A Workload-Driven Simulation and Design-Space Exploration Framework for Heterogeneous NPUs

Arghadip Das†, Hoseok Kim†, Soomin Lee, Arnab Raha, Deepak A Mathaikutty, and Vijay Raghunathan
kim5396@purdue.edu

***Abstract*—AI model architectures are diversifying rapidly. Although dense matrix multiplication still underlies today's CNNs and transformers, emerging architectures (state-space models, long convolutions via the fast Fourier transform (FFT), Kolmogorov–Arnold networks, and spiking networks) are not multiply–accumulate (MAC) dominated; they spend much of their computation on vector and non-MAC primitives that homogeneous, MAC-centric neural processing units (NPUs) serve poorly. This has motivated heterogeneous NPUs (HPUs) built from non-identical tiles. Prior heterogeneous designs, however, vary only one or two coarse knobs (typically MAC precision or array size) and are evaluated on narrow workloads, and no existing framework supports HPU design at a fine granularity, where tiles differ across many architectural dimensions at once.**

We present TESSERA, an analytical simulator and design-space-exploration (DSE) framework for HPU microarchitecture design. TESSERA searches the joint space of tile-level heterogeneity: beyond array size and precision, it varies the knobs along which tiles can differ, including tile-type composition (large *Big*, small *Little*, and non-MAC *Special-Function* tiles), dataflow, sparsity mode, MAC engine type, and special-function units for non-MAC operators (FFT, spiking-integrate, polynomial). Unlike prior simulators, which model a single homogeneous tile type, TESSERA models non-MAC tiles with their own energy, area, and timing models and maps operators across a mix of tiles with a heterogeneity-aware compiler. A multi-seed pipeline that pairs a stratified sweep with genetic-algorithm refinement returns Pareto-optimal designs, with cost models calibrated to a 7 nm node and cross-validated against NVIDIA's Deep Learning Accelerator (NVDLA). Across a 20-workload suite, the best general-purpose HPU found by TESSERA (∼200 mm^2 Big+Little+Special-Function) achieves +46.91% mean iso-area energy savings over the best iso-area homogeneous baseline.

***Index Terms*—Heterogeneous Architecture, Neural Processing Unit, Mixed-Precision, Design-Space Exploration, Energy Efficiency, High Performance.**

†Equal contribution.

This work was supported in part by the Center for the Co-Design of Cognitive Systems (CoCoSYS), one of the centers in the Joint University Microelectronics Program (JUMP) 2.0, a Semiconductor Research Corporation (SRC) program sponsored by the DARPA.

Arghadip Das (corresponding author, e-mail: das169@purdue.edu), Hoseok Kim (e-mail: kim5396@purdue.edu), Soomin Lee (e-mail: lee5782@purdue.edu), and Vijay Raghunathan (e-mail: vr@purdue.edu) are with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA.

Arnab Raha (e-mail: arnab.raha@intel.com) and Deepak A Mathaikutty (e-mail: deepak.a.mathaikutty@intel.com) are with the Advanced Architecture Research Team, NPU IP, CGAI (CCG), Intel Corporation, Santa Clara, CA 95054 USA.

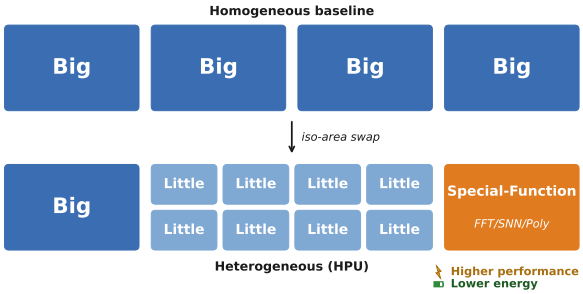


Fig. 1. TESSERA's key idea: a homogeneous NPU replicates one MAC-centric tile (top); emerging non-MAC operators leave much of its silicon dark. At the *same total area*, TESSERA's design-space exploration reallocates the chip into a heterogeneous mix of *Big*, *Little*, and *Special-Function* (FFT/SNN/poly) tiles (bottom), yielding heterogeneous NPUs (HPUs) with +46.91% mean iso-area energy savings over the best homogeneous baseline.

I. Introduction

Neural-network workloads are diversifying rapidly, moving well beyond the dense general matrix multiplications (GEMMs) of convolutional neural networks (CNNs) and transformers [1]–[3] to include selective state-space scans (Mamba [4]), long convolutions executed via the fast Fourier transform (FFT; Hyena [5]), Kolmogorov–Arnold networks built from polynomial basis functions (KAN [6]), and spiking neural networks (SNNs) [7]. A single multimodal inference pass now mixes many of these operator types at once. The hardware on which these workloads run, however, has not diversified at the same rate.

We refer to AI accelerators as neural processing units (NPUs); the term is generic, spanning the deployment spectrum from battery-powered edge and client devices to datacenter inference servers. Commercial NPUs from Intel [8], Qualcomm [9], AMD [10], and MediaTek [11] replicate identical compute tiles built around large multiply–accumulate (MAC) arrays, vector digital signal processors (DSPs), and matched SRAMs, which are designs implicitly tuned for the dense GEMMs of CNNs and transformers. Emerging operators do not stress this substrate evenly. On commercial NPUs, MAC utilization varies sharply: dense CNNs saturate the array, while state-space-model (SSM) scans, FFT-based long convolutions, KAN polynomial bases, and graph-neural-network (GNN)

2835-2238/26/$31.00 ©2026 IEEE
DOI 10.1109/IISWC71020.2026.00035

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

gathers spend most of their time outside it [12]–[14], so end-to-end inference latency degrades on hardware never built for them. The same mismatch leaves substantial *dark silicon*. A quantized INT8 layer never touches the FP16 datapath, a depthwise convolution barely fills a 32×32 MAC array, and an FFT or polynomial operator's MAC-fabric lowering wastes most of the multiplier silicon.

These inefficiencies carry two costs, and both matter across the deployment spectrum. The primary cost is performance: emerging operators stall on a substrate built for dense GEMM, slowing inference on edge and datacenter NPUs alike. The secondary cost is energy: underused datapaths still draw power, which shortens battery life on edge devices and raises the operating cost and carbon footprint of datacenter inference.

An NPU is built from *tiles*, and each tile contains one or more *cores* (compute units such as a MAC array, a vector DSP, or a special-function unit) together with local memory and ports. Tile-level heterogeneity (i.e., composing a chip from non-identical tiles whose cores differ in type, precision, array size, dataflow, memory capacity, or specialized operator support) is one response to this mismatch, but no existing study or tool treats it as a primary design knob. Prior work shows that exposing architectural diversity improves latency, energy, or area efficiency for specific workload classes by varying MAC precision [15], array dimensions [16], or coarse functional partitioning (a CNN core plus a recurrent-neural-network (RNN) core [17], and plug-in coarse-grained reconfigurable arrays (CGRAs) for nonlinear large-language-model (LLM) operators [18]). However, each of these studies fixes a *single coarse-grained knob* of heterogeneity and evaluates it on a *narrow* workload spectrum. This coarse, hand-partitioned heterogeneity is now reaching commercial silicon: MediaTek's Dimensity 9500 (NPU 990) couples a high-performance compute engine with a low-power compute-in-memory (CIM) core in a dual-NPU design [19]. There is no tool today that lets an architect ask: *which mix of tile types, MAC array dimensions, precisions, sparsity modes, dataflows, memory capacities, and special-function units is right for a given workload suite?* The simulators that enable design-space-exploration (DSE) (e.g., Timeloop [20], SCALE-Sim [21], MAESTRO [22], and Voyager [23]) all assume a homogeneous tiling design, modeling a single uniform tile with no notion of mixed tile types or specialized non-MAC compute units. Consequently, no existing study or tool supports systematic exploration of multi-knob tile-level heterogeneity across diverse workloads.

We present **TESSERA**, an analytical simulator and DSE framework that designs heterogeneous NPUs (HPUs) by treating structural tile-level heterogeneity (Big, Little, and Special-Function (FFT/SNN/polynomial) tile types, each with a per-tile configurable precision set, composed per workload mix) as a joint design space. The two MAC-based types play complementary roles: a *Big* tile is the general-purpose workhorse—a wide datapath that covers the broad operator mix at any precision from FP16 to INT4—whereas a *Little* tile is a power-optimized, low-precision unit built for peak energy efficiency on the quantized layers that dominate modern inference. *Special-Function* tiles add dedicated datapaths for the non-MAC operators (FFT, SNN, polynomial) that map poorly onto either. TESSERA jointly explores tile-type composition, MAC array geometry, precision (including INT4/INT8 quantization), sparsity mode, dataflow, memory-hierarchy capacity, and specialized functional units (FFT, SNN, polynomial). Given a target neural network suite, it produces per-operator and per-tile breakdowns of latency, energy, power, and utilization, together with tile-area estimates. These characterizations drive a multi-seed DSE pipeline (a stratified random sweep over the joint knob space followed by a per-area-budget *genetic-algorithm (GA) refinement* seeded from the sweep bests) that returns Pareto-optimal architectures for the workload mix.

Our contributions are as follows:

- **Heterogeneity-aware analytical simulator.** Unlike prior simulators, which model one homogeneous tile type, TESSERA models a chip as a mix of non-identical tiles, including non-MAC tiles for FFT, spiking-integrate, and polynomial operators, coordinated by a chip-level orchestrator that captures DRAM bandwidth sharing, cross-tile activation caching, network-on-chip (NoC) traffic, and per-tile power gating. (Section III-C)
- **Multi-seed DSE pipeline.** A stratified random sweep over the 12-knob design space (3 seeds, $\sim$2.94 M samples), followed by per-area-budget GA refinement, searches the joint heterogeneous space and returns Pareto-optimal architectures. (Section III-E)
- **20-workload characterization and three-group taxonomy.** The 20 workloads, spanning ten architectural families, fall into three groups by how much they benefit from heterogeneity: quantized LLMs/CNNs, FP16 transformer/SSM, and bandwidth-bound. (Section V-C)
- **Silicon-grounded calibration.** Per-module energy, area, and timing are calibrated against ASAP7 7 nm synthesis [24] and cross-validated end-to-end against the open-source NVIDIA Deep Learning Accelerator (NVDLA) [25] at two design points, backed by a system-level register-transfer-level (RTL) gating study. (Section III-D)

TESSERA's DSE returns HPUs that beat the best iso-area homogeneous designs across the suite. Peak savings reach $60.10 \pm 1.18\%$ on ResNet-50 (mean $\pm$ stdev, 3 seeds). The GA-refined Big+Little+Special-Function design at $\sim$200 mm^2 reaches **+46.91% mean iso-area energy savings** and **2.44 mean TOPS/W** (tera-operations per second per watt); a $\sim$100 mm^2 variant is **1.5–2.4$\times$ faster** than NVDLA-large on INT8/SSM/ViT workloads. A complementary system-level RTL gating study delivers **28% more MACs at 93.6% lower power**.

II. Background and Motivation

A. Generic NPU Tile

Production NPUs are built by replicating an identical tile template across the chip. Fig. 2 shows the canonical layout.

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

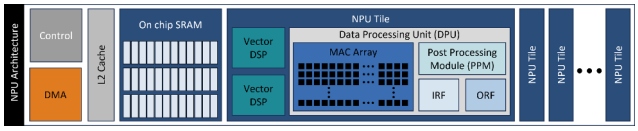


Fig. 2. Generic NPU tile template: a MAC array and vector DSP fed from an SRAM scratchpad through input/output register files, with load/store ports to a shared DRAM channel.

Each tile pairs a MAC fabric (typically a 16×16 to 64×64 systolic array supporting INT8 and FP16) with a vector DSP that executes activations, normalizations, elementwise operations, and softmax. A 256 KB–2 MB on-chip SRAM scratchpad feeds the compute units through an input register file (IRF) and an output register file (ORF), and dedicated load/store ports stage tensors to and from off-chip DRAM. Multiple tiles share a single DRAM channel and communicate over a mesh, ring, bus, or NoC interconnect. The resulting three-level energy hierarchy ($\sim$1–3 pJ/byte at IRF/ORF, $\sim$5 pJ/byte at SRAM, 40–200 pJ/byte at DRAM [26], [27]) shapes every tiling, reuse, and dataflow decision an NPU compiler makes.

Commercial NPUs from Intel [8], Qualcomm [9], AMD [10], and MediaTek [11] all replicate this template identically: every tile exposes the same precisions, the same operator set, and the same memory hierarchy, all driven by a single clock domain. The bet is that the tile is well-matched to a workload mix dominated by dense convolution and matrix multiplication, and that adding tiles is the right knob for scaling performance. The remainder of this section examines whether that bet still holds.

B. Operator Diversity in Modern Workloads

Neural-network workloads have moved past dense GEMM as the single dominant kernel. A typical workload mix now spans dense CNNs and vision transformers [1], [2] where MAC fabrics still pay off, SSMs [4], [28] and Hyena [5] long-convolution layers that lean on selective scans and FFT, KANs [6] that evaluate polynomial bases per edge, SNNs that integrate-and-fire, GNNs that scatter and gather, hybrid attention/SSM LLMs (Nemotron-H [29], Hymba [30]) that interleave qualitatively different layers within a single inference, and post-training-quantized LLMs (LLaMA [3], Mixtral [31], Nemotron [29] in INT4/INT8 forms) that ship with $4\times$ higher arithmetic intensity than their FP16 counterparts. Each family stresses a different subset of the tile.

Quantitative evidence: end-to-end breakdown on real silicon. We measured per-operator latency for six representative workloads on the Intel Lunar Lake (LNL) NPU [8] via OpenVINO's [32] per-op profiling counters, grouping each operator into *MAC* (Conv/MatMul/FC), *Vector* (elementwise, activation, normalization, softmax; targeted at the DSP's single-instruction multiple-data (SIMD) datapath), or *Other* (FFT, polynomial, SNN integrate, SSM scan, gather/scatter; no dedicated hardware, lowered onto MAC or DSP). Fig. 3 reports the breakdown. ResNet-50 is the only MAC-bound workload. Hyena spends $\sim$30% in FFT lowered onto MAC; OLMoE's expert-routing softmax/gating dominates the Vector

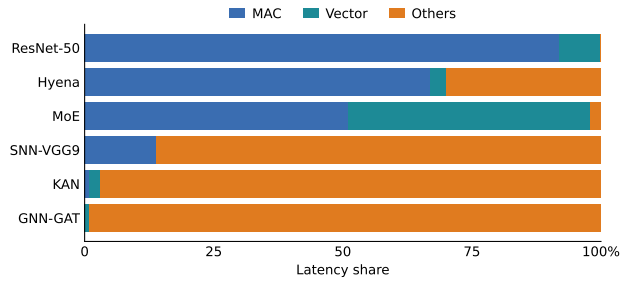


Fig. 3. Per-operator inference latency breakdown on the Intel LNL NPU, grouped into MAC, Vector, and Other (non-MAC) operators. Only ResNet-50 is MAC-bound.

share; SNN-VGG9 spends $\sim$47% in leaky integrate-and-fire (LIF) integration; KAN's wall time is entirely polynomial basis evaluation; GNN-GAT is dominated by gather/scatter. Prior measurement studies report MAC utilization below 10% on commercial NPUs for GNN [14], Mamba/SSM [12], and KAN-style [13] networks. The dark-silicon problem is no longer hypothetical; it is the common case for emerging model families.

C. Three Scopes of Heterogeneity

Operator diversity manifests at three architectural scopes: *across models* (a deployed system-on-chip (SoC) serves FFT-heavy long convolutions, LIF-bound SNNs, polynomial-bound KANs, INT4 LLMs, and dense-GEMM CNNs that cannot share a single tile template), *within a model across layers* (artificial-neural-network/SNN (ANN–SNN) networks alternate MAC and spiking-integrate layers; Nemotron-H [29] and Hymba [30] interleave attention and SSM blocks), and *within a layer across groups* (group-level quantization, per-edge KAN bases of varying degree). TESSERA targets the *across-models* scope (one silicon design per workload mix) and the *across-layers* scope via the mapper (Section III-B); within-layer heterogeneity is future work (Section VII). What is needed is a chip composed of *different* tile types tailored to the operator mixture of the target workload suite: a *heterogeneous* architecture where tiles differ in MAC array geometry, supported precisions, sparsity mode, dataflow, memory capacities, and the presence or absence of dedicated functional units.

D. The Need for a Heterogeneous Design Framework

Prior heterogeneous-NPU studies have repeatedly demonstrated the benefits of moving away from homogeneous tiling, but each fixes a single coarse-grained knob and evaluates on a narrow workload spectrum. DNPU [17] hard-wires a CNN core paired with an RNN core; Spantidi et al. [15] vary MAC precision alone (the big.LITTLE NPU); Maleki et al. [16] vary array dimensions alone; PICACHU [18] adds a plug-in CGRA for nonlinear LLM operators. Production silicon is moving the same way: MediaTek's Dimensity 9500 (TSMC N3P) ships a dual-NPU SoC delivering 100 TOPS [19], making heterogeneous-NPU design a near-term commercial reality, not

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

just a research idea. Yet no prior simulator supports cross-knob exploration: production DSE simulators (Timeloop [20], SCALE-Sim [21], MAESTRO [22], Voyager [23]) all model homogeneous substrates with a single tile type and no dedicated models for non-MAC accelerators (FFT, SNN, polynomial). An architect today therefore has no principled way to ask: *which mix of tile types, precisions, MAC array geometries, sparsity modes, dataflows, memory capacities, and special-function units is right for a given workload suite?*

Closing this gap requires four co-designed capabilities that resist adoption in isolation: (i) multi-tile heterogeneity (independent tile types with the knobs above); (ii) dedicated energy/area/cycle models for non-MAC accelerators; (iii) a heterogeneity-aware mapper that filters operators by tile compatibility and schedules across tiles with NoC-aware splitting; and (iv) a cross-knob search loop over the joint design space. A hetero-aware mapper without non-MAC models cannot evaluate Special-Function tiles; cross-knob search without those prerequisites is uninformative. TESSERA is, to our knowledge, the first framework to provide all four within a single tool.

These four capabilities are tightly coupled, so retrofitting them onto a homogeneous simulator would require abstractions that such simulators lack. Supporting non-identical tile types changes operator compatibility, cross-tile mapping and splitting, NoC communication, shared DRAM-bandwidth arbitration, non-MAC cost models, and the search loop itself. TESSERA therefore builds on FlexNPU's per-tile analytical model [33] and adds the heterogeneity-aware components of Section III-B through Section III-E. TESSERA also stays at the architecture-design stage and deliberately omits later-stage features that existing tools already provide, namely exhaustive kernel-level mapping, cycle-accurate traces, RTL generation, quantization-accuracy evaluation, and physical-design feedback. Those tools remain complementary downstream.

E. Iso-Area Intuition

At a fixed chip area $N \times A$, replacing N identical MAC tiles with a heterogeneous mix of Big, Little, and optionally Special-Function tiles extracts more performance and energy efficiency per mm^2 than the homogeneous baseline. Fig. 1 sketches the contrast. The top is the kind of homogeneous chip today's commercial NPUs ship: N identical FP16+INT8 MAC tiles of footprint A each; the bottom is a heterogeneous reallocation: a smaller number of large Big tiles, several smaller Little tiles, and optionally a Special-Function tile whose primary compute is an FFT, SNN-integrate, or polynomial unit rather than a MAC array, with total area held at $N \times A$. We use "Big" and "Little" by analogy with CPU big.LITTLE [34]: the Big tile is large and typically carries the wider, higher-precision datapath, the Little tile is small and typically restricted to a narrower, lower-precision set—though the supported precisions are a per-tile knob, not a fixed property of either type. This reallocation buys three things at the same $N \times A$:

- *Lower energy at the same area.* Lower-precision tiles consume less per MAC, and idle tiles can be power-gated to a small residual.
- *Lower latency at the same area.* The freed area is reinvested into more parallelism on the tiles the workload actually exercises.
- *Workloads that homogeneous cannot serve well.* Specialized functional units change the cost model asymptotically for FFT ($O(N \log N)$ butterflies vs. $O(N^2)$ on a MAC-array lowering, a $\sim$100$\times$ blow-up at Hyena's typical N=512), and by large constant factors for LIF (a few gates per neuron vs. a multiplier-array lowering that contributes nothing to the result) and polynomial operators (KAN inference is reduced from a long multiply–accumulate chain hopping through SRAM at every step, to a d-cycle Horner-rule fused multiply-add pipeline with the accumulator pinned in a register), within the same area budget.

Which mix wins for which workload is precisely the question TESSERA's DSE answers (Section V).

III. TESSERA Framework

TESSERA takes a workload suite together with a candidate architecture and returns single-inference latency, energy, area, power, and per-tile utilization. Driving the same pipeline with a search loop produces Pareto-optimal heterogeneous architectures over a $> 10^{14}$-point design space.

Fig. 4 shows the four cooperating layers: *Inputs* (Section III-A), *cost-aware compiler* (Section III-B; precision, fusion, mapping, op-splitting, dataflow), *heterogeneity-aware simulator* (Section III-C; per-tile module pipeline plus chip-level orchestrator), *calibration layer* (Section III-D; synthesized RTL + DRAM literature), and the *DSE engine* (Section III-E; multi-seed stratified sweep + GA refinement).

A. Inputs

Workload specification. A workload is a directed acyclic graph (DAG) of operators, each carrying a type (23-entry vocabulary: 5 MAC-class, 15 DSP-class, 3 special (SNN-integrate, FFT, polynomial)), shape, precision (INT4/INT8/FP16/BF16/FP32), and per-operand sparsity rates. Workloads are imported from ONNX [35] or extracted from PyTorch [36].

Architecture configuration. An architecture lists one or more *tile templates*, per-template instance counts, an interconnect topology (mesh/bus/ring/NoC), DRAM size, bandwidth, and latency. Each tile template exposes a small set of architectural knobs: enabled modules (MAC array, DSP, special unit, load/store ports), MAC engine type and dimensions, supported precisions, sparsity mode (none, activation-/weight-sided, two-sided, or structured N:M [37]), SRAM capacity and banking, IRF/ORF granularity, DSP unit count and SIMD width, and double-buffering. Each tile type is clocked in its own fixed domain. The same schema describes a homogeneous chip (one template), a mixed-precision chip with two tile types, or a three-tile-type chip (e.g., an FP16+INT8 Big, an INT4+INT8 Little, and a Special-Function tile). A representative homogeneous baseline mirroring an Intel LNL-class NPU [8] is expressed as a set of identical FP16+INT8 MAC tiles with matched SRAM

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

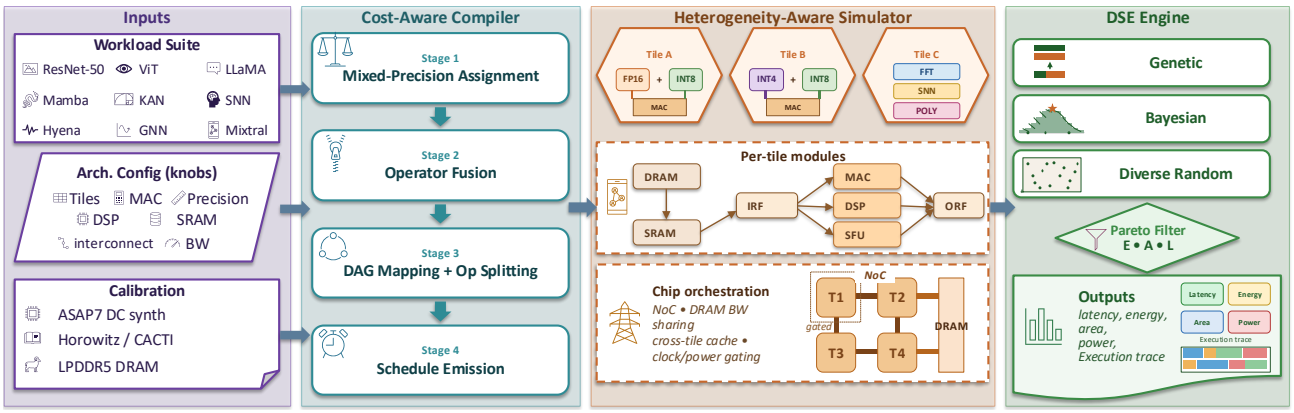


Fig. 4. TESSERA framework overview: four cooperating layers—inputs, cost-aware compiler, heterogeneity-aware simulator, and silicon calibration—wrapped in a multi-seed DSE search loop.

and DSP units, sharing a mesh interconnect and a single DRAM channel.

B. Cost-Aware Compiler

The compiler converts a (workload, architecture) pair into an execution plan in four ordered passes. Each pass tags operators for the simulator and DSE; no machine code is emitted.

(1) Mixed-precision assignment. Default policy (Conv/MatMul/Pool→INT8; LayerNorm/RMSNorm/Softmax/SNN/FFT/polynomial/SSM scan→FP16) with a name-based override forcing FP16 on accuracy-sensitive layers (attention QKV/output projection, LM head, classifier, embedding) [38]. An *aggressive* mode demotes all convolutions to INT4.

(2) Operator fusion. A greedy left-to-right scan matches three-op (Conv+BN+Act, Conv+Add+Act) and two-op patterns; matched groups fold post-processing into the tile's post-processing module (PPM), skipping the SRAM round-trip for intermediate tensors.

(3) DAG-aware mapping with op-splitting. Operators are visited in topological order. For each operator o, the mapper filters tiles by op-type and precision compatibility, then for each compatible tile T computes an earliest start time

$$t_{\text{start}}(o,T) = \max\Big(\text{tile_finish}[T], \max_{(f_j,T_j)\in\text{preds}(o)} \big(f_j + \mathbf{1}[T_j{\neq}T]\,\Delta_{\text{NoC}}\big)\Big), \tag{1}$$

where $\text{tile_finish}[T]$ is the cycle at which T finishes the work already scheduled on it, and the second term is the latest time at which a predecessor's output is available on T. Predecessor j finishes at cycle f_j on its assigned tile T_j, and the indicator $\mathbf{1}[T_j{\neq}T]$ adds the NoC delay Δ_{NoC} only when j ran on a different tile. The mapper then computes a roofline cycle estimate (the larger of the compute-bound and bandwidth-bound cycle counts)

$$\hat{C}_{\text{op}}(T) = \max\Big(\Big\lceil \frac{\text{MACs}}{R_T C_T \eta_T} \Big\rceil,\ \Big\lceil \frac{\text{bytes}}{B_{\text{DRAM}}(T)} \Big\rceil\Big), \tag{2}$$

where MACs and bytes are the operator's MAC count and DRAM traffic, $B_{\text{DRAM}}(T)$ is the DRAM bandwidth available to tile T in bytes per cycle, $R_T{\times}C_T$ is the MAC array shape, and η_T is a per-MAC throughput multiplier (>1 when sparsity skipping applies). The mapper places o on the tile minimizing $t_{\text{start}} + \hat{C}_{\text{op}}$ (completion time, not just cycle estimate). For MAC-class ops with multiple compatible tiles, it evaluates an even split along the output-channel (OC), batch (B), or input-channel (IC) dimension with explicit reduce/concat cost

$$C_{\text{reduce}} = \max_i \big(\lceil B_i^{\text{out}}/B_{\text{NoC}} \rceil + \Delta_{\text{NoC}}\big), \tag{3}$$

where B_i^{out} is the byte count of partition i's output returned over the NoC (a partial-sum reduction for IC splits, a concatenation for OC/B splits) and B_{NoC} is the NoC bandwidth in bytes per cycle, accepting the split only if its finish time beats single-tile placement. When the split dimension does not divide evenly across the participating tiles, each tile receives a whole number of elements along that dimension and the final partition absorbs the remainder, so the partitions are unequal in size. Each partition is then costed on its assigned tile with Eqs. (1)–(3) before the split is accepted. Dataflow (weight-stationary (WS), output-stationary (OS), row-stationary (RS), or AUTO) is picked per operator, because the best dataflow for a matmul depends on its shape and no single fixed dataflow such as WS is best across all shapes. In an $M{\times}K$ by $K{\times}N$ matmul the operands are sized $M{\cdot}K$ and $K{\cdot}N$ and the output $M{\cdot}N$. OS holds the $M{\cdot}N$ partial sums in the array and accumulates them in place, avoiding repeated partial-sum traffic to SRAM, so OS pays off only when the output is far larger than the operands. WS is the default, holding the $K{\cdot}N$ weights and reusing them across the M activation rows. AUTO therefore chooses OS when $M{\cdot}N$ exceeds both $K{\cdot}N$ and $M{\cdot}K$ by $4\times$, WS otherwise. The $4\times$ threshold is a heuristic rather than an analytically derived value, set large enough that AUTO stays on WS unless the output clearly dominates.

What changes under a heterogeneous architecture. The compatibility filter routes each op to the smallest compatible tile (FP16 MATMUL→Big, INT8 Conv→any, FFT→Special-Function);

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

splittable MAC ops are partitioned across Big+Little so the bulk runs on the small tiles. FP16-only ops on chips with one FP16-capable tile serialize.

(4) Schedule emission. A scheduler converts the per-op mapping into an execution schedule (latency mode parallelizes distinct-tile assignments; throughput mode pipelines multi-batch).

C. Heterogeneity-Aware Simulator

The simulator consumes the compiled execution plan, the architecture configuration, and the calibration tables and returns latency, energy, area, and per-tile utilization. It is built on the FlexNPU [33] dataflow-aware analytical model, extended with non-MAC accelerator modules and a chip-level orchestrator that handles heterogeneous tile mixes. The tile-template plus NoC abstraction is general enough to model not just single-NPU multi-tile designs but also multi-NPU SoCs such as MediaTek's dual-NPU Dimensity 9500 [19], where each NPU instance is expressed as a tile cluster sharing an inter-NPU NoC channel.

1) Per-Tile Module Pipeline: A tile is modeled as seven hardware modules; three are compute cores (the MAC array, the DSP, and the special-function unit) and the rest handle memory and data staging. Given a compiled operator, the tile simulator routes the operator through one of three execution paths (MAC, DSP, or Special-Function) and accumulates cycles and energy at each module.

- *Compute module (MAC array).* An SRAM-budget tiling pass decomposes the operator along (M, K, N) into tiles that fit the working set (weights and double-buffered activations within the per-tile SRAM). Per-tile cycles follow an engine-specific model; for systolic arrays,

$$C_{\text{sys}} = \sum_{n,k} \Big[D + \sum_{m} (m_{\text{eff}} + k_{\text{eff}} + D - 2) \Big], \quad (4)$$

  where m, k, n index the (M, K, N) tiles produced by the tiling pass, m_{eff} and k_{eff} are the effective M and K extents of the current tile (smaller at boundary tiles), and D is the systolic pipeline depth, so each tile pays $m_{\text{eff}}+k_{\text{eff}}+D-2$ cycles of array fill and drain. Compute energy is sparsity-aware MAC count times per-MAC energy from the calibration table; six sparsity modes are supported.
- *DRAM module.* Burst-aligned reads/writes at per-tile bandwidth, $E_{\text{DRAM}} = (\text{bytes}_{\text{rd}} + \text{bytes}_{\text{wr}})\, E_{\text{DRAM/B}}$.
- *SRAM module.* Tiling-aware reuse from the chosen dataflow (WS/OS/RS), with grid heuristics as fallback.
- *IRF / ORF.* IRF pays write energy padded to write granularity, reads reduced by activation sparsity. ORF is K-tile-aware (first K-tile write-only, later ones read-modify-write).
- *DSP module.* A SIMD instruction set of 14 vector operations (`vadd`, `vmul`, `vexp`, `vreduce`, `vlut`, ...) covers every elementwise/normalization/activation/softmax operator; each high-level op decomposes into a vector sequence. SSM scan carries a sequence-length sequential multiplier.
- *Special-Function module (special-function unit, SFU).* Dedicated formulas for radix-2 FFT ($N \log_2 N$ cycles for an N-point transform), LIF ($\lceil N/N_{\text{par}} \rceil T$ cycles for N neurons integrated over T timesteps on N_{par} parallel lanes), and Horner polynomial (Nd cycles for N evaluations of a degree-d polynomial).

2) Total-Cycle Model: The seven-module pipeline interleaves DRAM staging, SRAM tiling-aware reuse, IRF/ORF activation/partial-sum feed, and PPM post-processing on top of each tile's compute path. With double-buffering (DB), total cycles overlap compute, memory, and DRAM:

$$C_{\text{tot}} = \begin{cases} \max(C_{\text{cmp}}, C_{\text{mem}}, C_{\text{DRAM}}) + C_{\text{LP}} + C_{\text{SP}} & \text{DB} \\ C_{\text{cmp}} + C_{\text{mem}} + C_{\text{DRAM}} + C_{\text{LP}} + C_{\text{SP}} & \text{no DB} \end{cases} \quad (5)$$

where C_{cmp} is the cycle count of the operator's compute path (MAC, DSP, or SFU), C_{mem} the on-chip SRAM and IRF/ORF staging cycles, C_{DRAM} the off-chip transfer cycles, and C_{LP}, C_{SP} are load/store-port direct-memory-access (DMA) cycles.

3) Latency, Energy, and Area Models: **Latency.** Per-tile latency $L = C_{\text{tot}}/f$; end-to-end latency is the makespan of the chip schedule including NoC delays and split-op reductions.

Energy. Total tile energy sums contributions from every enabled module:

$$\begin{aligned} E_{\text{tile}} = {} & E_{\text{cmp}} + E_{\text{DRAM}} + E_{\text{SRAM}} + E_{\text{IRF}} \\ & + E_{\text{ORF}} + E_{\text{DSP}} + E_{\text{spec}} - E_{\text{fuse}}, \end{aligned} \quad (6)$$

where each right-hand term is the calibrated energy of the corresponding module and $E_{\text{fuse}} = N_{\text{fused}} \cdot 2\,|\text{out}|\, E_{\text{SRAM/B}}$ accounts for SRAM round-trips avoided by fused intermediates. Each of the N_{fused} fused pairs skips one write and one read of an $|\text{out}|$-byte intermediate tensor at $E_{\text{SRAM/B}}$ per byte. Chip energy adds NoC transfer and static leakage from power-gated tiles.

Area. Tile area is the analytical sum

$$\begin{aligned} A_{\text{tile}} = {} & N_{\text{MAC}} \cdot \max_{p} A_{\text{MAC}}(p) \\ & + A_{\text{SRAM}} + A_{\text{DSP}} + A_{\text{spec}} + A_{\text{ports}}, \end{aligned} \quad (7)$$

with $N_{\text{MAC}} = R_T C_T$ the tile's MAC count and per-MAC area $A_{\text{MAC}}(p)$ taken over the largest supported precision p (multi-precision MACs include the wide datapath); IRF/ORF area folds into A_{ports}. Chip area is $A_{\text{chip}} = \sum_t A_{\text{tile}}(t) + A_{\text{NoC}}$, omitting floorplan dead space (flagged in Section VII; RTL gating study in Section V-A bounds it).

4) Heterogeneous Mechanisms: Two mechanisms make the analytical model heterogeneity-aware.

Dynamic DRAM bandwidth sharing. Only tiles whose previous operator has not yet finished are counted active; per-tile bandwidth is $\text{BW}_{\text{total}}/N_{\text{active}}$, so a subset of busy tiles gets full bandwidth rather than a static share.

Cross-tile activation caching and synchronization. Each tile's SRAM splits between a working set and a FIFO-evicted activation cache. Three cases: local cache hit (same-tile producer/consumer $\Rightarrow$ no DRAM read), cross-tile DMA (NoC transfer at $\lceil B/B_{\text{NoC}} \rceil + h\, C_{\text{base}}$ cycles for B bytes moved at

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

NoC bandwidth B_{NoC} bytes per cycle over h hops with per-hop base latency C_{base}), or cache miss (full DRAM load). A pre-built consumer map enforces dependency synchronization across heterogeneous tiles.

Clock and power gating. Idle modules within an active tile are clock-gated (no dynamic energy). Tiles with no scheduled work are power-gated at 5% residual leakage, a figure consistent with header-switch power gating at modern technology nodes.

5) Heterogeneous Tile Types: The tile-template knobs let a single configuration schema represent three qualitatively different tile types that recur in our DSE outputs. The supported-precision set is itself a per-tile knob: any tile may declare an arbitrary subset of the available datatypes, so the precision labels below are illustrative of representative DSE points, not hard-wired tile classes.

- *Big tile.* A large systolic array with ample SRAM, two-sided sparsity, and dual DSP. Hosts the highest-precision operators in a configuration—e.g., FP16 attention, normalization, and high-accuracy convolution—and is typically provisioned with a wide precision set (e.g., FP16+INT8).
- *Little tile.* A small array with modest SRAM, a single DSP, and optional sparsity. Hosts the lower-precision, INT-quantizable convolution and matmul at substantially lower energy than the Big tile, and is typically provisioned with a narrower, lower-precision set (e.g., INT4+INT8).
- *Special-Function tile.* No MAC array. Includes one or more SFUs (FFT, SNN-integrate, polynomial), modest SRAM, and a single DSP for pre/post-processing. Hosts the non-MAC operator classes for which MAC lowering is asymptotically wasteful.

The same chip simulator orchestrates any cross product of these tile types; what differs across architectures is the per-type count and per-type sizing.

6) Outputs: For each (workload, architecture) pair the simulator emits end-to-end latency, total/average power, and area; per-tile active cycles, achieved/peak TOPS, TOPS/W, TOPS/mm^2, arithmetic intensity, and roofline class; per-module energy breakdowns (compute, DRAM, SRAM, IRF, ORF, DSP, special, NoC, leakage); and an execution trace in the Perfetto profiling format for visual inspection of tile utilization and cross-tile movement.

D. Calibration

Energy, area, and timing parameters come from Synopsys Design Compiler (DC) synthesis of every RTL module at the ASAP7 7 nm process design kit [24], 2 GHz target. SRAM area is taken from CACTI 7.0 [27]; SRAM leakage is taken from DC.

DRAM calibration. DRAM energy and bandwidth come from DRAM-process literature rather than logic synthesis. The ASAP7 calibration pairs with LPDDR5-6400 (40 pJ/byte, 51.2 GB/s rounded to 64 GB/s on the DSE grid, 100-cycle access latency). The bandwidth is a DSE knob and is swept over $\{16, 32, 64, 128, 256, 512\}$ GB/s.

System-level RTL validation. A dedicated study packages a homogeneous two-tile system and an iso-area heterogeneous (one FP16+INT8 + one INT4+INT8) system into 109 SystemVerilog files ($\sim$19.3k lines of code) synthesized at ASAP7. The homogeneous design clock-gates the FP16 datapath when running INT8; the heterogeneous design power-gates the INT4+INT8 tile when idle. Results in Section V-A.

External cross-validation. The same calibration flow matches TESSERA against NVIDIA NVDLA [25] at two design points spanning 32$\times$ in MAC density: `nv_small` (8$\times$8 INT8 systolic, 64 KB convolution buffer (CBUF)) and `nv_full` (32$\times$64 INT8+FP16, 512 KB CBUF), using the published NVDLA Primer's area and energy values as the external reference. NVDLA is the natural target here, and effectively the only one: it is the sole openly available production NPU that ships synthesizable RTL together with a published per-module area and energy breakdown, which is why analytical DSE simulators are likewise each validated against a single such reference (e.g., Timeloop [20] against the Eyeriss accelerator). NVDLA is an edge-class design, so this check exercises the framework at edge area and power scales; the same cost models and search apply to datacenter-scale NPUs, where only the area, power, and bandwidth budgets supplied to the DSE differ.

E. Design Space Exploration

The DSE engine drives the compiler+simulator over the joint heterogeneous knob space and returns a Pareto front in (energy, area, latency) together with the best design under a user-supplied scalar objective.

Knobs. The DSE varies 12 knobs spanning tile-type composition, MAC-array geometry, precision, on-chip memory, DRAM bandwidth, sparsity, dataflow, interconnect, and pipelining. The full knob space exceeds 10^{14} configurations; the per-knob value grid is enumerated in Section IV-E.

Search pipeline. The DSE runs in two stages, across multiple random seeds for confidence intervals. *Sweep:* stratified random sampling over the knob space (strata = area budget $\times$ architecture family); it is the source of the per-workload best-of-DSE numbers in Section V-B. *GA refinement:* a per-area-budget genetic algorithm seeded from the best sweep individuals, returning the best general-purpose heterogeneous architecture per area budget (Section V-C). Alongside these two stages, the DSE engine also provides a Bayesian optimization (BO) search backend (a surrogate model fit to evaluated configurations, with new candidates proposed by an acquisition function) as a sample-efficient alternative to the GA refinement when the simulation budget is constrained.

Objective. The GA-refinement fitness is the workload-equal-weighted mean iso-area energy savings of the candidate design over the best homogeneous design at the same area (from the sweep), normalized by the best peak TOPS/W observed:

$$\mathrm{fitness}(d) = \overline{\Delta E}_{\text{iso-area}}(d) \;+\; \alpha \cdot \frac{\mathrm{TOPS/W}(d)}{\max_{d'} \mathrm{TOPS/W}(d')}, \quad (8)$$

where $\overline{\Delta E}_{\text{iso-area}}(d)$ is the mean iso-area energy savings of candidate design d and α is a small positive tie-breaker. We report both *peak* TOPS/W (best-single workload) and *mean* TOPS/W (workload-equal-weighted); for general-purpose chip-level claims the mean is the right metric.

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

TABLE I
WORKLOAD SUITE. 14 BASE MODELS IN TEN ARCHITECTURAL FAMILIES; TRANSFORMER LLMS INCLUDE INT4/INT8 QUANTIZED VARIANTS.

Family	Model	Precision
CNN	ResNet-50 [1]	INT8
Vision Transformer	ViT-B/16 [2]	FP16, INT8
Dense LLM	LLaMA-7B [3]	FP16, INT8, INT4
Dense LLM	Spec. decode [39]	FP16
Mixture-of-experts LLM	Mixtral [31]	FP16, INT4
Hybrid attention/SSM LLM	Nemotron-H [29]	FP16, INT8, INT4
SSM	Mamba-370M [4]	FP16
SSM	Hyena-1.3B [5]	FP16
KAN	KAN [6]	FP16
SNN	SNN-VGG9 [7]	FP16
Multimodal	LAVISH [40]	FP16
Multimodal	LLaVA [41]	FP16
Multimodal	RT-2 [42]	FP16
GNN	GNN-GAT [43]	FP16

TABLE II
DESIGN-SPACE GRID. ASYMMETRIC-PRECISION MAC VARIANTS USE $W x A y$ NOTATION FOR x-BIT WEIGHTS AND y-BIT ACTIVATIONS.

Knob	Values
MAC array rows/columns	8, 16, 32, 64, 128
Per-tile SRAM capacity	64–4096 KB in 2× steps
Precision set	INT8; INT4+INT8; INT8+FP16; INT4+INT8+FP16
DRAM bandwidth	16, 32, 64, 128, 256, 512 GB/s
Tile counts	1–3 tile types, 1–8 instances per type
Sparsity mode	none, activation-sided, two-sided
MAC engine type	systolic, spatial, dot-product, CIM
Dataflow policy	WS, OS, RS
Interconnect	mesh, bus, ring, NoC
Double-buffering	off, on
Asymmetric-precision MAC	none, W4A8, W2A8, W4A16+W8A16
Pipeline depth	1, 4, 8, 16

IV. EXPERIMENTAL METHODOLOGY

A. Workload Suite

We evaluate TESSERA on a 20-workload suite: 14 base models spanning ten architectural families, plus six post-training-quantized INT4/INT8 variants of the widely deployed transformer LLMs (Table I). The suite was constructed to (i) exercise all 23 operator types in TESSERA's vocabulary, (ii) stress every execution path on the tile (MAC, DSP, Special-Function), (iii) span five orders of magnitude in arithmetic intensity, and (iv) cover the INT4/INT8 post-training quantization that production LLMs increasingly ship in.

B. Metrics

We report end-to-end single-batch inference latency, energy, and area as the primary metrics (lower is better), consistent with the Pareto objectives in Section III-E. Energy is the single-inference energy with per-module breakdown (compute, DRAM, SRAM, IRF, ORF, DSP, special, NoC) split into dynamic vs. static. Area is the total chip area (mm^2) with per-tile breakdown into MAC array, SRAM, DSP, special unit, and ports. Achieved TOPS, per-tile processing-element utilization, and the derived *peak* TOPS/W (best-single workload) and *mean* TOPS/W (workload-equal-weighted) are listed separately as supplementary metrics for which higher values are better. For chip-level claims we report the mean.

C. Experimental Setup

Technology node. All headline results target ASAP7 7 nm and use the default DC synthesis calibration described in Section III-D, paired with LPDDR5-6400 (40 pJ/byte). Big and Little tile types run in fixed clock domains at 1200 MHz and 500 MHz, respectively.

Architectures compared. At each area bracket we compare three families: *Homogeneous* (Homo, N identical FP16+INT8 tiles), *Big–Little* (Hetero-BL, mixed FP16+INT8 Big and INT4+INT8 Little tiles), and *Big–Little–Special-Function* (Hetero-BLS, adds Special-Function tiles for FFT/SNN-integrate/polynomial). The GA refinement (Section IV-E) returns Hetero-BLS as best at every area budget.

Comparison constraints. Comparisons run under three constraints: *iso-area* (area matched; report energy/latency), *iso-latency* (latency matched; report energy/area), and *iso-energy* (energy matched; report area/latency).

D. Simulator Validation Setup

The simulator is validated along three axes (Section V-A): (i) per-module cycle and energy agreement against Synopsys DC at ASAP7 7 nm; (ii) end-to-end agreement against NVIDIA NVDLA at two design points spanning 32× in MAC density (`nv_small` 8×8 INT8 64 KB CBUF; `nv_full` 32×64 INT8+FP16 512 KB CBUF); and (iii) a system-level RTL gating study (109 SystemVerilog files, homogeneous vs. iso-area heterogeneous, both synthesized at ASAP7 and run at 1 GHz with the heterogeneous design's INT4+INT8 tile power-gated when idle) that grounds the analytical power-gating model.

E. DSE Setup

The DSE pipeline runs two stages, each repeated across multiple random seeds for confidence-interval estimation.

Design-space grid. Table II lists the 12 DSE knobs and the values each can take. The winning designs of Section V-C differ mainly in tile-type composition, supported precision set, Little-tile count, and DRAM bandwidth.

Sweep. Stratified random sampling over this 12-knob design space, with three random seeds. Each seed evaluates ∼980 K configurations across area brackets $\{50, 100, 200, 400, 800\}$ mm^2 and architecture families {Homo, Hetero-BL, Hetero-BLS}, yielding ∼2.94 M total samples. The sweep is the source of the per-workload best-of-DSE numbers reported in Section V-B.

GA refinement. Per-area-budget GA (population 200, 100 generations, tournament selection of size 5, 80% crossover, 20% mutation, 10% elitism) seeded from the top 50 individuals returned by the sweep at that budget. Five parallel GA instances cover the area sweep $\{50, 100, 200, 400, 800\}$ mm^2. Fitness is the workload-equal-weighted mean energy savings of the candidate heterogeneous design over the best homogeneous design at the same area (found in the sweep), normalized to the

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

TABLE III
TESSERA VS. NVDLA ON INT8 64×64×64 GEMM, TWO DESIGN POINTS. *Ratio* COLUMNS REPORT TESSERA/NVDLA.

	nv_small (64 MAC)			**nv_full** (2048 MAC)		
Metric	NVDLA	TESSERA	Ratio	NVDLA	TESSERA	Ratio
Peak TOPS	0.064	0.064	1.00×	2.048	2.048	1.00×
Latency (μs)	5.12	5.52	1.08×	1.15	1.60	1.39×
Energy (nJ)	567.7	803.1	1.41×	567.7	677.2	1.19×
Area (mm^2)	0.40	0.71	1.77×	3.31	4.96	1.50×
TOPS/W	0.58	0.44	0.76×	4.16	4.85	1.17×

best peak TOPS/W observed. The DSE engine also supports BO, compared against the sweep and the GA in Section V-C. **Reporting convention.** Sweep numbers are reported as *cross-seed mean ± stdev (95% confidence interval)* over the three seeds. Per-workload best-of-DSE numbers report the mean and stdev of the single best sampled design per workload across the three seeds. GA-refinement results are reported as the best individual returned by the GA at each area budget.

V. RESULTS

The results are organized in three parts. Section V-A validates the simulator at three scales (per-module against ASAP7 synthesis, end-to-end against NVDLA on a GEMM cross-check, and against synthesized NVDLA-large on every workload it can execute) and reports the system-level RTL gating study. Section V-B presents the multi-seed DSE per-workload best designs. Section V-C reports the GA refinement, the best general-purpose heterogeneous architecture per area budget, and the three-group workload taxonomy that emerges.

A. Simulator Validation

1) Per-Module Calibration: The simulator is calibrated against Synopsys DC synthesis at ASAP7 7 nm for every RTL module that backs an analytical model, which is what grounds the absolute energy and area numbers reported in this paper. SRAM area is taken from CACTI 7.0 [27] rather than the standard-cell register array.

2) End-to-End vs. NVDLA: We validate end-to-end accuracy against the open NVIDIA NVDLA [25] at two design points spanning 32× in MAC density: `nv_small` (8×8 INT8 systolic, 64 KB CBUF) and `nv_full` (32×64 INT8+FP16, 512 KB CBUF). Both are exercised on an INT8 64×64×64 GEMM that fully fits in each design's on-chip buffer (Table III). NVDLA is the natural—and effectively the only—external reference for this cross-check: it is the sole openly available production NPU that ships synthesizable RTL *together* with a published per-module area and energy breakdown, so it is the one design an analytical model can be checked against end-to-end at module granularity. This mirrors established practice for analytical DSE simulators, each of which is validated against a single such reference (e.g., Timeloop [20] against the Eyeriss accelerator), simply because no second design publishes RTL and reference numbers at comparable fidelity. The cross-check is nonetheless stronger than a single comparison: NVDLA

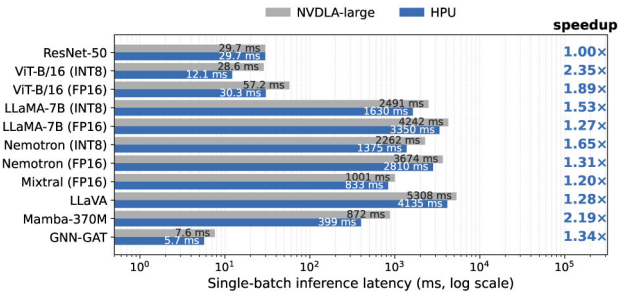


Fig. 5. Single-batch inference latency of the GA-refined HPU (Hetero-BLS, ∼100 mm^2) versus synthesized NVDLA-large on every NVDLA-supported workload, with per-workload speedup at right.

is only the *external* axis of the three-axis validation in this section—grounded bottom-up by per-module ASAP7 synthesis (Section V-A) and topped by a 109-file system-level RTL gating study—and even that axis spans two design points across 32× in MAC density, so agreement reflects scaling behavior rather than a single tuned operating point.

Table III reports each metric and the TESSERA-over-NVDLA ratio at both scales. *Peak TOPS* matches by construction. *Latency* ratios (1.08× small, 1.39× full) are on par with the fidelity that comparable analytical simulators (Timeloop [20], SCALE-Sim [21]) report against synthesized references. *Energy* sits at 1.41× small and shrinks to 1.19× full, a per-inference overhead that amortizes with scale. *Area* ratios (1.77×, 1.50×) reflect a constant bias: NVDLA's reported area covers only its convolution MAC array and CBUF; once TESSERA strips its load/store ports and PPM accounting (the `nvdla_cmac` + CBUF subset), the area matches the synthesized 3.238 mm^2 **within 2%** (3.308 mm^2). *TOPS/W* follows from energy and area. For TESSERA's *differential* use (comparing architectures under identical modeling assumptions), a constant bias cancels out, and the monotone tightening across the 32× MAC-density span between `nv_small` and `nv_full` confirms scaling correctness.

3) System-Level RTL Gating Study: At iso-area and 1 GHz, the synthesized heterogeneous chip (5×5 FP16+INT8 tile + 4×4 INT4+INT8 tile, the INT4+INT8 tile power-gated when idle) draws **93.6% less power** than the synthesized homogeneous baseline (2× 4×4 dual-datapath tiles with the FP16 path clock-gated) while providing **28.1% more MACs** (41 vs. 32) in **8.3% less area** (302,931 vs. 330,320 μm^2). At 500 MHz the same heterogeneous chip saves 41.9% area and 70.7% power. The 93.6% power-reduction figure agrees within 6% of TESSERA's analytical 95% leakage-elimination model, providing direct silicon validation of the gating model. The study reports post-synthesis cell area without full place-and-route and without a Special-Function tile.

4) End-to-End vs. NVDLA-large on the Workload Suite: Beyond the GEMM-level NVDLA validation, we compare a representative GA-refined general-purpose HPU (Hetero-BLS, ∼100 mm^2 ASAP7) against the synthesized NVDLA-large (2048-MAC INT8+FP16 systolic, 512 KB CBUF) on

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

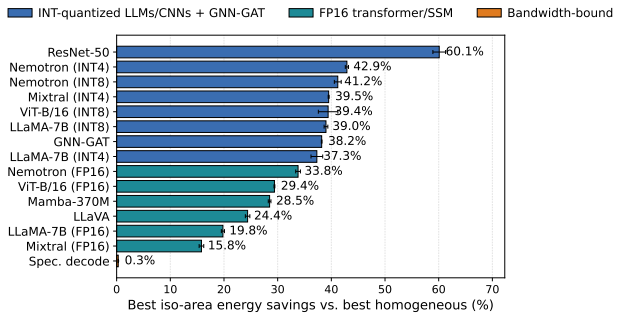


Fig. 6. Per-workload best iso-area energy savings of the DSE-selected heterogeneous design against the iso-knob homogeneous baseline; mean ± stdev across 3 random-sampling seeds.

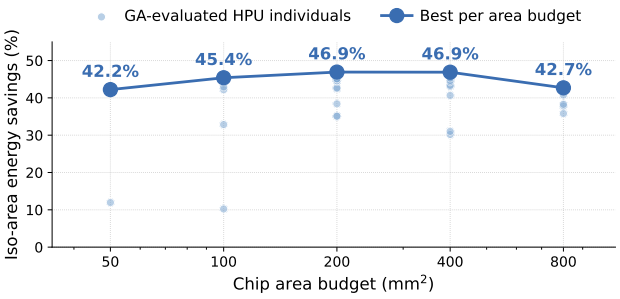


Fig. 7. GA-refined mean iso-area energy savings vs. chip-area budget; the per-budget best traces an inverted-U peaking in the 100–400 mm^2 band.

every workload NVDLA-large can execute (Fig. 5). Two findings stand out. First, the HPU reaches latency parity on ResNet-50 INT8 (NVDLA's design target) and is faster on every other NVDLA-supported workload, with 1.5–2.4× **wins on INT8/SSM and compute-bound ViT workloads** and smaller 1.2–1.3× wins on the FP16 dense-LLM decodes, where attention's FP16-only operators serialize on the single Big tile. Second, the HPU draws 1.1–2.0× more energy per inference than NVDLA-large; the energy–latency trade is the architect's to make and the GA refinement surfaces it as a Pareto choice (Section V-C). The four workloads NVDLA-large cannot execute, namely three INT4 LLMs (LLaMA-7B, Nemotron, Mixtral) and RT-2's multimodal operators, are omitted from Fig. 5; on them the HPU's INT4-native Little tile reaches ∼0.70 TOPS/W (RT-2 at 0.27 TOPS/W), unlocking deployment of GPTQ/AWQ-style post-training-quantized LLMs [44], [45] that the homogeneous baseline cannot serve.

B. DSE Across Hardware Knobs

We ran a multi-seed sweep over the 12-knob design space (3 seeds, ∼2.94 M samples; Section IV-E). Each sampled heterogeneous design is scored by its workload-equal-weighted mean iso-area energy savings against the iso-knob homogeneous baseline (Eq. 8). For each workload we record the best sampled design and report the cross-seed mean/stdev (Fig. 6).

ResNet-50 at $+\mathbf{60.10 \pm 1.18}\%$ is the headline; all per-workload stdevs sit below 1.82%. Each bar's winner is workload-specific: the optimal Big/Little/Special-Function composition is determined by the target workload's precision and arithmetic intensity, so the design that wins on ResNet-50 loses on INT4 LLMs and vice versa. A clear ordering emerges and lines up with the three-group taxonomy (Section V-C): INT4/INT8 quantized workloads (plus GNN-GAT) cluster at the top at 37–60%; FP16 transformer/SSM workloads recover 16–34%; speculative decoding is bandwidth-bound (0.28%). Quantifying the Special-Function tile's benefit on the five non-MAC workloads (KAN, SNN-VGG9, Hyena, LAVISH, RT-2) requires an asymmetric per-Special-Function-tile precision knob that the current symmetric per-tile-precision DSE does not yet expose; this measurement is reported as future work (Section VII). The single-chip general-purpose winner is reported in Section V-C.

TABLE IV
SEARCH-METHOD COMPARISON AT THE 200 MM^2 BUDGET. THE GA AND BO SHARE THE OBJECTIVE, WARM START, AND EVALUATION BUDGET. RUNTIME IS SINGLE-WORKER EVALUATION TIME,EXCEPT THE SWEEP, WHICH RAN AS 24 PARALLEL JOBS. BO SEEDS SPAN 46.38 TO 47.05%.

Method	Best savings	Unique configs	Runtime
Sweep (200 mm^2 bracket)	18.95%	14.1 K	2 h × 24 jobs
Genetic algorithm	46.91%	3,253	65.0 h
Bayesian opt. (median seed)	46.84%	3,253	75.6 h

C. GA Refinement

The GA refinement runs a per-area-budget GA seeded from the top 50 individuals returned by the random sweep, scored by the workload-equal-weighted mean iso-area energy savings of Eq. 8, the same metric used to score random samples in Fig. 6. The five area instances evaluate **28,268 GA individuals** in total, with four of five budgets reaching the ten-generation no-improvement early-stop (the 50 mm^2 instance terminates against the 144 h wall after 440 individuals). Fig. 7 shows the per-area-budget best alongside the GA-evaluated individuals. *The GA returns Hetero-BLS as the optimal architecture at every area budget*: the Big+Little+Special-Function composition is robust across an order-of-magnitude span in chip area.

Best general-purpose chip. The 200 mm^2 Hetero-BLS winner reaches **+46.91% mean iso-area energy savings** and **2.44 mean TOPS/W** (peak 10.48), both workload-equal-weighted across the 20-workload suite. The inverted-U in Fig. 7 locates the sweet-spot in the 100–400 mm^2 band, where the three budgets land within 1.5 percentage points of each other ($+45.39/+46.91/+46.88\%$); the optimum is robust to area allocation in this band. The 400 mm^2 design hits the highest mean efficiency at 2.47 TOPS/W (peak 11.21). Smaller chips lack room for enough Little tiles, while larger chips amortize static area over fewer beneficial workloads (the 800 mm^2 regression to $+42.69\%$ comes from FP16-only ops that serialize on the few FP16-capable tiles).

Search-method comparison. Table IV compares the three search strategies at the 200 mm^2 budget. The sweep's best

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

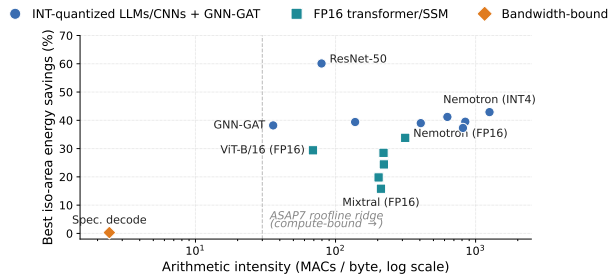


Fig. 8. Best iso-area energy savings vs. workload arithmetic intensity (MACs/byte) for the 15 MAC/DSP-dominant workloads; markers colored and shaped by the three-group taxonomy.

configuration reaches 18.95% under the objective of Eq. 8. Seeded identically from the sweep's top 50, the GA reaches 46.91% and BO 46.84% at the median of three seeds under the same budget of 3,253 unique evaluations. BO is the more sample-efficient of the two, passing 45% within 500 evaluations on every seed, about one seventh of the budget. Each method selects a different Hetero-BLS design of equivalent quality. The GA winner pairs one 64×8 Big tile with four Little tiles and a Special-Function tile, while the BO designs keep one Big and one Special-Function tile but use only one or two Little tiles at higher DRAM bandwidth. The plateau near 47% therefore reflects the design space rather than an artifact of one optimizer.

Workload-group taxonomy. The 15 MAC/DSP-dominant workloads fall into three groups (Fig. 8), distinguished by their best achievable iso-area savings and their position on the arithmetic-intensity axis. This grouping is a reading of the DSE results, not a design input. The design inputs are the tile types and per-tile knobs, which are motivated by the LNL operator measurements of Section II-B. *INT-quantized LLMs/CNNs and GNN-GAT (8)* reach 37–60% under per-workload tailoring, sitting at or past the ASAP7 roofline ridge (arithmetic intensity ≥ 30 MACs/byte) where the DSE picks a Big–Little mix and small INT4+INT8 Little tiles off-load quantizable (and, for GNN-GAT, INT8-compatible) layers. *FP16 transformer/SSM workloads (6)* recover 16–34% in the same compute-bound region by keeping FP16 attention on the Big tile while off-loading normalization, gating, and quantizable matmul fragments. *Bandwidth-bound (1)*, spec. decode at an arithmetic intensity of 2.4, recovers only 0.28% because no MAC sizing helps a memory-starved workload: the single point left of the roofline ridge in Fig. 8. The remaining five non-MAC workloads (KAN, SNN-VGG9, Hyena, LAVISH, RT-2) execute on the Special-Function tile that the GA-returned HPU instantiates; quantifying their savings against a homogeneous baseline requires the asymmetric per-Special-Function-tile precision DSE pass discussed in Section VII.

VI. Related Work

Heterogeneous NPU architectures. Prior work has explored heterogeneity along a single axis at a time. Herald [46] and SCAR [47] compose sub-accelerators that differ only in *dataflow* (weight- vs. output-stationary), while Stream [48] extends this paradigm with layer-fusion scheduling. Tiles in these designs share identical hardware structures, precisions, and functional units, leaving the structural efficiency of right-sized compute unexplored. DNPU [17] pairs a CNN core with an RNN core in a hard-wired two-core SoC, and the Renesas AI-MPU [49] integrates a CPU, reconfigurable processor, and AI accelerator on a 14 nm SoC; both demonstrate heterogeneous-core value but are tied to specific workload pairs and lack any DSE. Spantidi et al. [15] adopt a big.LITTLE NPU based on MAC precision alone, achieving $\sim$29% energy savings, but vary no other knob. CHARM 2.0 [50] composes field-programmable gate-array fabric and AI-Engine sub-accelerators on Xilinx Versal, and PICACHU [18] inserts a plug-in CGRA for nonlinear LLM operators next to an existing GEMM engine; both address complementary aspects of heterogeneity but do not provide a general-purpose heterogeneous-NPU substrate or simulator. Maleki et al. [16] sweep heterogeneous array sizes alone and report up to 36% energy reduction, again on a single axis. *No prior study jointly varies tile-type composition, precision, sparsity, dataflow, and special-function-unit presence*, leaving the cross-knob heterogeneity design space largely unexplored.

Simulation frameworks for NPUs. The widely used DSE simulators (Timeloop [20], SCALE-Sim [21], MAESTRO [22], and Voyager [23]) all model homogeneous substrates with a single tile type, lack dedicated models for non-MAC accelerators (FFT, SNN-integrate, polynomial), and have no notion of heterogeneity-aware mapping or DRAM bandwidth sharing across mixed tiles. As a result, they cannot evaluate the design points that recent operator-level workload studies on commercial NPUs (GraNNite [14], XAMBA [12], HKN [13]) identify as energy-leaving-on-the-table.

Operator-level orchestration on heterogeneous SoCs. A complementary line of work studies how to map individual operators to the right processing unit on a *fixed* heterogeneous SoC (e.g., CPU/GPU/NPU pipelines) [51], [52]. TESSERA inverts the question: given a workload mix, it produces the right heterogeneous chip composition in the first place.

VII. Limitations and Future Work

The current DSE does not independently vary precision by tile type, support within-layer heterogeneity, or use global mapper look-ahead. The model also omits floorplan and place-and-route effects, operates at operator-graph rather than kernel granularity, and models a single NPU rather than a multi-NPU SoC. These limitations define the scope of the reported 46.91% mean iso-area energy savings, which holds under the current design space and modeling assumptions. Each item below states a limitation and the follow-up that addresses it.

(1) Asymmetric per-tile-type precision DSE pass. The GA-returned HPU runs five non-MAC workloads (KAN, SNN-VGG9, Hyena, LAVISH, RT-2) natively on its Special-Function tile. Quantifying that tile's iso-area savings needs an asymmetric per-tile precision knob (e.g., an FP16 SNN

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

integrator alongside INT4 Big-tile MACs), absent from the current symmetric DSE—our highest-priority follow-up.

(2) Within-layer heterogeneity. TESSERA covers the across-models and across-layers scopes (Section II-C); group-level quantization within a matmul or convolution would add the third, within-layer scope.

(3) Mapper–architecture co-design. The greedy DAG mapper lacks global look-ahead, so on sparsely parallel graphs, tiles with rare op-type compatibility (e.g., the Special-Function tile during dense-MATMUL stretches) idle. A search-based or learned policy co-optimized with the tile-mix knobs would recover that slack.

(4) Physical-design feedback. The area model omits floorplan dead space, place-and-route (P&R) complexity, and power-density hotspots; the RTL gating study (Section V-A) bounds this mismatch to 8% on one two-tile composition, with no physical synthesis at the 7 nm target. Floorplan-aware modeling, full P&R, and advanced-node synthesis would tighten the iso-area comparisons and yield manufacturable designs.

(5) Kernel-fidelity simulation. The simulator operates at op-graph, not kernel, granularity; fusion savings are captured via metadata rather than full kernel fidelity. A kernel-level extension would tighten energy estimates for fusion-heavy workloads.

(6) SoC-scale heterogeneity across NPUs. TESSERA models heterogeneity within a single NPU. Commercial SoCs are beginning to integrate multiple distinct NPUs on one die—e.g., the dual-NPU design of MediaTek's Dimensity 9500, which pairs a compute engine with a CIM core [19]. Modeling such systems requires capturing inter-NPU communication and workload partitioning across NPUs, promoting TESSERA's chip-level orchestrator to an SoC-level scheduler.

VIII. Conclusion

TESSERA is the first workload-characterization and DSE framework that treats structural tile-type composition (Big, Little, and Special-Function (FFT/SNN/polynomial) tiles, each with a per-tile configurable precision set) as a joint design space, with an ASAP7-calibrated simulator cross-validated against NVDLA and a multi-seed sweep+GA pipeline. The GA-refined Big+Little+Special-Function HPU reaches **+46.91% mean iso-area energy savings** and **2.44 mean TOPS/W** on the 20-workload suite (peak $60.10 \pm 1.18\%$ on ResNet-50), is **1.5–2.4× faster** than NVDLA-large on INT8/SSM/ViT workloads, and a system-level RTL gating study confirms **93.6% lower power at 28% more MACs** in silicon. The contribution is the cross-knob platform that makes the Big+Little+Special-Function HPU discoverable in the first place.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[4] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *Conference on Language Modeling (COLM)*, 2024.

[5] M. Poli, S. Massaroli, E. Nguyen, D. Y. Fu, T. Dao, S. Baccus, Y. Bengio, S. Ermon, and C. Ré, "Hyena hierarchy: Towards larger convolutional language models," in *International Conference on Machine Learning (ICML)*, 2023.

[6] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov–Arnold networks," in *International Conference on Learning Representations (ICLR)*, 2025.

[7] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks," *IEEE Signal Processing Magazine*, 2019.

[8] Intel Corporation, "Intel Core Ultra mobile processors (series 2) product brief," 2024.

[9] Qualcomm Technologies, Inc., "Unlocking on-device generative AI with an NPU and heterogeneous computing," Qualcomm Technologies white paper, 2024.

[10] Advanced Micro Devices, Inc., "AMD XDNA architecture," 2025.

[11] MediaTek Inc., "MediaTek edge AI: The MediaTek NPU (APU)," 2025.

[12] A. Das, A. Raha, S. Kundu, S. K. Ghosh, D. Mathaikutty, and V. Raghunathan, "XAMBA: Enabling efficient state space models on resource-constrained neural processing units," *arXiv preprint arXiv:2502.06924*, 2025.

[13] A. Das, H. Kim, A. Raha, S. Kundu, S. K. Ghosh, D. Mathaikutty, and V. Raghunathan, "Towards efficient acceleration of Hyena and Kolmogorov–Arnold networks on NPUs," in *IEEE International Conference on Intelligent Computing and Systems at the Edge (ICEdge)*, 2025.

[14] A. Das, S. Kundu, A. Raha, S. Ghosh, D. Mathaikutty, and V. Raghunathan, "GraNNite: Enabling high-performance execution of graph neural networks on resource-constrained neural processing units," in *International Joint Conference on Neural Networks (IJCNN)*, 2025.

[15] O. Spantidi, G. Zervakis, S. Alsalamin, I. Roman-Ballesteros, J. Henkel, H. Amrouch, and I. Anagnostopoulos, "Targeting DNN inference via efficient utilization of heterogeneous precision DNN accelerators," *IEEE Transactions on Emerging Topics in Computing*, 2023.

[16] M. A. Maleki, M. Kamal, and A. Afzali-Kusha, "Heterogeneous multi-core array-based DNN accelerator," *arXiv preprint arXiv:2206.12605*, 2022.

[17] D. Shin, J. Lee, J. Lee, J. Lee, and H.-J. Yoo, "DNPU: An energy-efficient deep-learning processor with heterogeneous multi-core architecture," *IEEE Micro*, 2018.

[18] J. Qin, T. Xia, C. Tan, J. Zhang, and S. Q. Zhang, "PICACHU: Plug-in CGRA handling upcoming nonlinear operations in LLMs," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2025.

[19] TechInsights, "Dimensity 9500 debuts Arm C1 cores and dual-NPU," TechInsights blog, 2025.

[20] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. S. Emer, "Timeloop: A systematic approach to DNN accelerator evaluation," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2019.

[21] A. Samajdar, J. M. Joseph, Y. Zhu, P. N. Whatmough, M. Mattina, and T. Krishna, "A systematic methodology for characterizing scalability of DNN accelerators using SCALE-Sim," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2020.

[22] H. Kwon, P. Chatarasi, V. Sarkar, T. Krishna, M. Pellauer, and A. Parashar, "MAESTRO: A data-centric approach to understand reuse, performance, and hardware cost of DNN mappings," *IEEE Micro*, 2020.

[23] K. Prabhu, J. Yu, X. A. Pan, Z. Xie, A. Aleshire, Z. Chen, A. A. Ratnani, and P. Raina, "Voyager: An end-to-end framework for design-space exploration and generation of DNN accelerators," *arXiv preprint arXiv:2509.15205*, 2025.

[24] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "ASAP7: A 7-nm FinFET predictive process design kit," *Microelectronics Journal*, 2016.

[25] NVIDIA Corporation, "NVDLA: NVIDIA deep learning accelerator," 2017.

[26] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2014.

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.

[27] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "CACTI 6.0: A tool to model large caches," HP Laboratories, Tech. Rep., 2009.
[28] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," in *International Conference on Learning Representations (ICLR)*, 2022.
[29] NVIDIA, "Nemotron-H: A family of accurate and efficient hybrid Mamba–Transformer models," *arXiv preprint arXiv:2504.03624*, 2025.
[30] X. Dong, Y. Fu, S. Diao, W. Byeon, Z. Chen, A. S. Mahabaleshwarkar, S.-Y. Liu, M. Van Keirsbilck, M.-H. Chen, Y. Suhara, Y. Lin, J. Kautz, and P. Molchanov, "Hymba: A hybrid-head architecture for small language models," in *International Conference on Learning Representations (ICLR)*, 2025.
[31] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. Bou Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. Le Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.
[32] Intel Corporation, "OpenVINO toolkit," 2020.
[33] A. Raha, D. A. Mathaikutty, S. Kundu, and S. K. Ghosh, "FlexNPU: A dataflow-aware flexible deep learning accelerator for energy-efficient edge devices," *Frontiers in High Performance Computing*, 2025.
[34] P. Greenhalgh, "big.LITTLE processing with ARM Cortex-A15 and Cortex-A7," ARM white paper, 2011.
[35] ONNX Community, "Open neural network exchange (ONNX)," 2017.
[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
[37] A. Ramachandran, S. Kundu, A. Raha, S. Kundu, D. A. Mathaikutty, and T. Krishna, "Algorithm–hardware co-design of digital compute-in-memory architecture supporting flexible and temporal $N{:}M$ sparsity," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2026.
[38] A. Raha, S. Kundu, S. N. Sridhar, S. Kundu, S. K. Ghosh, A. Palla, A. Das, D. Crews, and D. A. Mathaikutty, "LLM-NPU: Towards efficient foundation model inference on low-power neural processing units," in *IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, 2025.
[39] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in *International Conference on Machine Learning (ICML)*, 2023.
[40] Y.-B. Lin, Y.-L. Sung, J. Lei, M. Bansal, and G. Bertasius, "Vision transformers are parameter-efficient audio-visual learners," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
[41] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
[42] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. Gonzalez Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. S. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. T. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning (CoRL)*, 2023.
[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.
[44] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "GPTQ: Accurate post-training quantization for generative pre-trained transformers," in *International Conference on Learning Representations (ICLR)*, 2023.
[45] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, "AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration," in *Proceedings of Machine Learning and Systems (MLSys)*, 2024.
[46] H. Kwon, L. Lai, M. Pellauer, T. Krishna, Y.-H. Chen, and V. Chandra, "Heterogeneous dataflow accelerators for multi-DNN workloads," in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2021.
[47] M. Odema, L. Chen, H. Kwon, and M. A. Al Faruque, "SCAR: Scheduling multi-model AI workloads on heterogeneous multi-chiplet module accelerators," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2024.
[48] A. Symons, L. Mei, S. Colleman, P. Houshmand, S. Karl, and M. Verhelst, "Stream: Design space exploration of layer-fused DNNs on heterogeneous dataflow accelerators," *IEEE Transactions on Computers*, 2025.
[49] K. Nose, T. Fujii, K. Togawa, S. Okumura, K. Mikami, D. Hayashi, T. Tanaka, and T. Toi, "A 23.9 TOPS/W @ 0.8 V, 130 TOPS AI accelerator with 16x performance-accelerable pruning in 14 nm heterogeneous embedded MPU for real-time robot applications," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2024.
[50] J. Zhuang, J. Lau, H. Ye, Z. Yang, S. Ji, J. Lo, K. Denolf, S. Neuendorffer, A. K. Jones, J. Hu, Y. Shi, D. Chen, J. Cong, and P. Zhou, "CHARM 2.0: Composing heterogeneous accelerators for deep learning on Versal ACAP architecture," *ACM Transactions on Reconfigurable Technology and Systems*, 2024.
[51] A. Das, Y. Agarwal, S. K. Ghosh, A. Raha, and V. Raghunathan, "Demo abstract: ECO: Low power context-aware multimodal AI on NPUs," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2025.
[52] B. Kundu, A. Rooj, S. Saha, A. Sarkar, A. Das, A. Raha, and M. K. Naskar, "KATANA: A fast, low-power mapping of Kalman filters onto edge NPUs for real-time tracking," *arXiv preprint arXiv:2606.14992*, 2026.

Authorized licensed use limited to: Purdue University. Downloaded on November 17,2026 at 19:59:38 UTC from IEEE Xplore. Restrictions apply.