Temperature-Aware Optimization of Monolithic 3D Deep Neural Network Accelerators

Prachi Shukla  
prachis@bu.edu  
Boston University

Sean S. Nemtzow  
nemtzow@bu.edu  
Boston University

Vasilis F. Pavlidis  
avasileios.pavlidis@manchester.ac.uk  
The University of Manchester

Aly K. Coskun  
acoskun@bu.edu  
Boston University

ABSTRACT

We propose an automated method to facilitate the design of energy-efficient Mono3D DNN accelerators with safe on-chip temperatures for mobile systems. We introduce an optimizer to investigate the effect of different aspect ratios and footprint specifications of the chip, and select energy-efficient accelerators under user-specified thermal and performance constraints. We also demonstrate that using our optimizer, we can reduce energy consumption by 1.6% and area by 2× with a maximum of 9.5% increase in latency compared to a Mono3D DNN accelerator optimized only for performance.

ACM Reference Format:

1 INTRODUCTION

Deep Neural Networks (DNNs) are extremely popular for numerous machine learning applications, such as image classification or object detection [1]. There is an increasing demand for DNNs in mobile systems, such as IoT devices, autonomous drones, tablets, etc. To satisfy the performance demands of these devices, DNN accelerators are actively being developed [2]. However, the high energy demand of DNNs (due to their heavy computation and data movement) is a major design issue. In addition, mobile systems have tight area and power/thermal budgets (e.g., due to the absence of heat sinks and fans) add to the constraints associated with designing energy-efficient DNN accelerators.

A systolic DNN accelerator comprises a two dimensional (2D) array of simple processing elements (PEs), with on-chip scratchpad memories for input feature map (IFMAP), filter weights (Filter), and output feature map (OFMAP), as shown in Fig. 1 [3]. Each PE consists of a Multiply-and-Accumulate (MAC) unit along with internal registers to store the inputs and partial sums. In a systolic architecture, data flows into the array from the PEs along the top and left edges (Fig. 1) and passes onto the neighboring PEs every clock cycle. Straightforward design and high compute density make systolic arrays a popular choice in mobile systems [2].

With technology scaling slowing down, improving performance under energy, power, and thermal constraints is increasingly more challenging. Monolithic 3D (Mono3D) is a three-dimensional (3D) integration technology that can overcome 2D scaling bottlenecks by achieving small chip footprint, dense integration, wire length savings, power savings, and high bandwidth [4]. These properties make Mono3D attractive for designing DNN accelerators in mobile systems. However, 3D architectures have significant thermal challenges due to high power densities and vertical thermal resistance [5]. In addition, Mono3D systems have thin device layers, resulting in limited lateral heat flow and high inter-tier thermal coupling (unlike through silicon via based 3D stacking), thus exacerbating thermal problems in mobile systems [6]. Consequently, temperature becomes an indispensable part of the methodologies and tools used to architect Mono3D systems.

This work focuses on designing energy-efficient architectures based on systolic arrays fabricated with Mono3D technologies, enabling DNN inference tasks in mobile systems. The primary contributions of this paper are as follows:

• We develop an automated method to investigate the performance, power, temperature, and energy trends in Mono3D DNN accelerators for a wide range of DNNs commonly used for mobile inference tasks.

• We integrate a DNN performance model and Mono3D power and thermal models to construct a comprehensive optimization flow. We also provide validation for our Mono3D thermal model.

• Compared to a Mono3D DNN accelerator that is only performance optimized, our optimizer reports up to 2× and 1.6× savings in chip footprint and energy, respectively, at the expense of a 9.5% increase in latency, while also satisfying the thermal budget.
The work proposes an automated method to perform a comprehensive partitioning in gate- and transistor-level partition [11–13], we focus on a systematic array-based partitioning with non-volatile memories [2], or designing dataflow mechanisms of connectivity through the Multi-Functional Interconnects (MIVs) [4]. The thin tiers and MIVs can overcome 2D scaling limitations and provide greater interconnect density, wire length reduction, power savings, and density integration than traditional TSV-based 3D ICs. There are three types of partitions possible in 2D accelerators without considering temperature. Another work achieves DNN energy efficiency and latency improvement by stacking memory-on-logic using through-silicon vias (TSVs) [10].

Fig. 3: Flow diagram of the optimization process.

2 RELATED WORK

DNN accelerators. Recent works target energy efficiency in systolic accelerators by adjusting DRAM design parameters, such as supply voltage and access latency [7], replacing off-chip DRAM with non-volatile memories [2], or designing dataflow mechanisms to improve data re-use and reduce SRAM accesses [6]. Prior works have also focused on co-designing DNN models and their corresponding hardware accelerators (e.g., [9]). These works focus on 2D accelerators without considering temperature. Another work achieves DNN energy efficiency and latency improvement by stacking memory-on-logic using through-silicon vias (TSVs) [10].

Mono3D. Mono3D is an emerging 3D integration technology where multiple tiers (or device layers) are fabricated sequentially, separated by thin dielectrics, even though current Mono3D fabrication challenges limit the number of tiers to two [4]. The vertical connections between the tiers are achieved using nano-scale inter-tier vias (MIVs) [4]. The thin tiers and MIVs can overcome 2D scaling limitations and provide greater interconnect density, wire length reduction, power savings, and denser integration than traditional TSV-based 3D ICs. There are three types of partitions possible in Mono3D: block-, gate-, and transistor-level. While there are several works in gate- and transistor-level partition [11–13], we focus on a two-tier block-level partition in this paper, in which 2D IP blocks can be used in the design process of Mono3D. Block-level partitions have been shown to achieve up to 16%, 8%, and 51% improvement in power, performance, and footprint, respectively, over 2D [14].

Prior works have focused on designing DNN accelerators in Mono3D but have not considered thermal awareness [15–18]. Yu et al. design a block-level Mono3D architecture with an FPGA-based accelerator and several resistive RAM tiers to improve performance, power, and energy compared to a 2D baseline with an off-chip DRAM [15, 16]. Chang et al. implement accelerators (with MAC units and SRAMs) for two DNN models with different weight compression on a two-tier Mono3D system [17], and show up to 22.3% iso-performance power savings for block-level integration. Do et al. integrate a two-tier Mono3D scratchpad memory on a GPU and provide 46% performance improvement [18]. In contrast, our work proposes an automated method to perform a comprehensive architecture-level performance, power, and temperature analysis for various DNNs and underlying architecture parameters to determine the most energy-efficient accelerator while also satisfying the thermal and performance constraints.

Key Innovation. To the best of our knowledge, this is the first work that offers a temperature-aware analysis and optimization framework for Mono3D DNN accelerators. The proposed framework enables navigating performance versus temperature tradeoffs for Mono3D systolic DNN accelerators aiming mobile systems.

3 DESIGN OPTIMIZATION METHOD FOR MONO3D DNN ACCELERATORS

This section details our design optimization method for DNN accelerators in mobile systems. As shown in Fig. 3, our method takes a DNN topology (e.g., MobileNet [19]) and design constraints as inputs to a Mono3D optimizer that determines design parameters for the accelerators for the subsequent iterations and finally outputs a near-optimal accelerator with safe chip temperatures. This optimization flow starts with performance evaluation using SCALE-Sim, a cycle-accurate simulator for systolic DNN accelerators [20]. SCALE-Sim outputs, along with CACTI-6.5 (an SRAM simulator) [21] and Mono3D power models, are then used to generate power traces for the accelerator. We then use HotSpot v6.0 (which we configure to simulate Mono3D systems) to obtain steady state temperatures [22]. We also implement a feedback loop that updates the power traces with temperature-dependent leakage (resulting from inter-tier thermal coupling) for HotSpot, to obtain updated chip temperatures. This loop continues until the temperature converges.

3.1 Mono3D DNN Accelerator Design

Existing Mono3D technologies can typically support only two tiers due to the low fabrication temperature requirements for upper tiers [4]. Due to this, we have limited our design to two tiers (see Fig. 2 for a cross-sectional view). The number of metal layers, dielectric/device layer thickness, and material properties of the stack are taken from recent work [4, 12]. The systolic array has a higher power consumption than SRAMs and is placed on the tier closer to the heat spreader. The systolic array and SRAMs have a high degree of connectivity through the MIVs since there are many read/write accesses to the SRAMs throughout the computations in the systolic array. We assume a high logic density for the tier with the systolic array, with SRAMs of the appropriate size on the other tier. Any whitespace (as a result of area mismatch between the two tiers) always appears on the SRAM tier in our design. We place whitespaces along chip edges so that thermal analyses are not affected.

3.2 Mono3D Optimizer

We construct a multi-start simulated annealing (MSA) based optimization workflow to sweep a sufficient portion of the design space of accelerators and select near-optimal energy-efficient Mono3D architectures for mobile systems. MSA is a search algorithm that accepts solutions that temporarily degrade the optimization goal to escape from local minima. MSA can launch multiple “starts” in parallel to increase the probability of finding the global minima.

Our optimizer takes a DNN topology and the following design constraints as inputs (Fig. 3): (i) chip footprint budget; (ii) bounds on chip aspect ratio; (iii) limits on systolic array size, (iv) maximum SRAM size, (v) maximum allowed whitespace (as a result of mismatch between the two tiers in the Mono3D chip), (vi) thermal budget (i.e., maximum allowed peak temperature, \( T_{\text{threshold}} \)), and (vii) maximum performance loss \( C_{\text{loss, max}} \) w.r.t. the fastest design that satisfies the design constraints (i)-(vii). The optimizer generates performance, power, and thermal traces for systematically selected
Temperature-Aware Optimization of Monolithic 3D
Deep Neural Network Accelerators

MONO3D accelerators, and converges to a near-optimal design for the user-specified optimization goal (e.g., minimizing energy latency) while satisfying performance and thermal constraints. For the systematic selection of accelerators, the optimizer uses the operating frequency, chip's aspect ratio, and combinations of systolic array and SRAMs as its control knobs. Since the array and SRAMs have to satisfy the whitespace constraint, we produce a list of all possible combinations offline ($\ell_{comb}$).

Algorithm 1 details our optimizer, which is inherently parallelizable because all the "starts" run in parallel (line 1). Each start is assigned an operating frequency and an aspect ratio range (AR), within which the optimizer determines a near-optimal solution by minimizing the objective function, $Obj$, which can be inference latency, chip power, energy or another energy efficiency metric. $T_{start}$, $T_{finish}$, and decay ($\delta$) are parameters of the optimizer that define the annealing temperatures and the rate of cooling. Each start begins by randomly choosing an initial accelerator ($S_i$) that satisfies the design constraints (i)-(vi) listed above (lines 3-7). We set $S_i$ as the current solution ($S_{curr}$) and initialize the latency ($C_{curr}$), smallest latency ($C_{best}$), peak temperature ($T_{peak,curr}$), and $Obj_{curr}$ with $S_i$'s parameters (lines 8-10). We then randomly perturb $S_{curr}$ by selecting a feasible design ($S_p$) from $\ell_{comb}$ (lines 11-15). If the DNN inference latency on the perturbed design ($C_p$) is smaller or within a user-specified performance degradation ($C_{loss,max}$) from $C_{best}$, this design (i.e., $S_p$) is 'accepted' for the next step. Otherwise, it is 'rejected' (lines 16-19). The 'accepted' $S_p$ is then thermally simulated for steady state analysis. If the peak chip temperature ($T_{peak,p}$) is greater than the thermal budget ($T_{threshold}$), $S_p$ is rejected. Otherwise, $S_p$ is checked for a lower $Obj_p$. If $Obj_p$ is smaller than $Obj_{curr}$, then $S_p$ is 'accepted'. Otherwise, it is 'accepted' with a certain probability (lines 20-30). The term $\Delta Obj$ is the difference between $Obj_p$ and $Obj_{curr}$, while $\Delta Obj_{tang}$ refers to the running average of $\Delta Obj$ for the accepted designs. The 'accepted' $S_p$ is then set as $S_{curr}$ for the next iteration (lines 31-35).

The algorithm terminates upon satisfying the conditions in lines 1 and 11. The accepted designs are stored in a file. Finally, the optimizer selects the best design among all the starts with the least $Obj$ while satisfying the performance and thermal constraints (line 39). If the user's objective is to design a single accelerator for multiple DNNs, then additional meta strategies could be integrated to the optimizer, e.g., selecting the most efficient design out of several optimized solutions for all target DNNs on average, or the design that yields the best results for the most frequently run DNNs.

### 3.3 Performance Model

SCALE-Sim is a cycle-accurate simulator for systolic arrays that operate on 8-bit integer data. It takes the array and SRAM size, along with DRAM bandwidth as inputs, simulates a stall-free DNN inference, and outputs compute cycles, non-overlapping DRAM cycles, array utilization, SRAM accesses, and DRAM bandwidth to support stall-free inference. Compute cycles include cycles spent in data transfer between SRAMs and systolic array, along with DRAM cycles that overlap with the computation. We divide the compute cycles and non-overlapping cycles by chip and DRAM frequencies, respectively, to calculate the latency. Among the several dataflows

---

1. Annealing temperature is a unitless parameter in MSA that allows it to escape a local minima by accepting a design with a higher $Obj$ value. Rate of cooling is the rate at which the annealing temperature decays to achieve convergence.

---

**Algorithm 1: MSA-based Temperature-Aware Optimizer**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$T_{peak,i} \leftarrow T_{threshold} + 1$</td>
</tr>
<tr>
<td>2</td>
<td>while $T_{peak,i} &gt; T_{threshold}$ do</td>
</tr>
<tr>
<td>3</td>
<td>randomly select an accelerator ($S_i$) with AR and a frequency</td>
</tr>
<tr>
<td>4</td>
<td>if $S_i$ meets design constraints (i)-(v) in Sec. 3.2 then</td>
</tr>
<tr>
<td>5</td>
<td>generate performance traces, calculate inference latency $C_i$</td>
</tr>
<tr>
<td>6</td>
<td>generate power traces, estimate peak temperature ($T_{peak, i}$)</td>
</tr>
<tr>
<td>7</td>
<td>set current solution: $S_{curr} \leftarrow S_i$, $C_{curr} \leftarrow C_i$, $T_{peak,curr} \leftarrow T_{peak, i}$</td>
</tr>
<tr>
<td>8</td>
<td>calculate $Obj_{curr}$</td>
</tr>
<tr>
<td>9</td>
<td>initialize best performance, $C_{best} \leftarrow C_{curr}$</td>
</tr>
<tr>
<td>10</td>
<td>while $T \times T_{finish} &gt; T_{peak, i}$ do</td>
</tr>
<tr>
<td>11</td>
<td>while $N &gt; 0$ do</td>
</tr>
<tr>
<td>12</td>
<td>randomly select a design, $S_p$, from $\ell_{comb}$ with AR $</td>
</tr>
<tr>
<td>13</td>
<td>$N \leftarrow 1$</td>
</tr>
<tr>
<td>14</td>
<td>if $S_p$ meets design constraints (i)-(v) in Sec. 3.2 then</td>
</tr>
<tr>
<td>15</td>
<td>calculate $C_p$ and loss in performance $C_{loss} = C_{curr} - C_{curr}$</td>
</tr>
<tr>
<td>16</td>
<td>initialize status $\leftarrow$ 'Reject'</td>
</tr>
<tr>
<td>17</td>
<td>if $C_{loss} \leq C_{loss,max}$ then</td>
</tr>
<tr>
<td>18</td>
<td>status $\leftarrow$ 'Accept'</td>
</tr>
<tr>
<td>19</td>
<td>if status $\leftarrow$ 'Accept' then</td>
</tr>
<tr>
<td>20</td>
<td>status $\leftarrow$ 'Reject'</td>
</tr>
<tr>
<td>21</td>
<td>generate power traces and calculate $T_{peak, p}$</td>
</tr>
<tr>
<td>22</td>
<td>calculate $Obj_p$</td>
</tr>
<tr>
<td>23</td>
<td>if $T_{peak, p} \leq T_{threshold}$ then</td>
</tr>
<tr>
<td>24</td>
<td>$\Delta Obj_{tang} = abs(Obj_p - Obj_{curr})$</td>
</tr>
<tr>
<td>25</td>
<td>if $Obj_p \leq Obj_{curr}$ then</td>
</tr>
<tr>
<td>26</td>
<td>status $\leftarrow$ 'Accept'</td>
</tr>
<tr>
<td>27</td>
<td>else if $Obj_p &gt; Obj_{curr}$ then</td>
</tr>
<tr>
<td>28</td>
<td>if random(0,1) $&lt; \exp(-\frac{\Delta Obj}{\Delta Obj_{tang}})$ then</td>
</tr>
<tr>
<td>29</td>
<td>status $\leftarrow$ 'Accept'</td>
</tr>
<tr>
<td>30</td>
<td>if status $\leftarrow$ 'Accept' then</td>
</tr>
<tr>
<td>31</td>
<td>$S_{curr} \leftarrow S_p$, $C_{curr} \leftarrow C_p$, $Obj_{curr} \leftarrow Obj_p$</td>
</tr>
<tr>
<td>32</td>
<td>update $\Delta Obj_{tang}$</td>
</tr>
<tr>
<td>33</td>
<td>if $C_p &lt; C_{curr}$ then</td>
</tr>
<tr>
<td>34</td>
<td>$C_{best} \leftarrow C_{curr}$</td>
</tr>
<tr>
<td>35</td>
<td>Store $S_{curr}$, $C_{curr}$, $Obj_{curr}$, $T_{peak, curr}$ in a data structure</td>
</tr>
<tr>
<td>36</td>
<td>$T \leftarrow T \times \delta$</td>
</tr>
<tr>
<td>37</td>
<td>$multi_starts \leftarrow 1$</td>
</tr>
<tr>
<td>38</td>
<td>return $S_{best}$, $T_{peak} \leq T_{threshold}$ &amp; $C_{loss} \leq C_{loss,max}$</td>
</tr>
</tbody>
</table>

### 3.4 MONO3D Power Models

We use SCALE-Sim outputs to obtain the average dynamic power of the systolic array ($P_{SA, Dynamic}$) using Eqs. (1) and (2):

$$U_{avg} = \left( \sum_{i=1}^{N} U_i \right) / \left( \sum_{i=1}^{N} C_i \right), \quad (1)$$

$$P_{SA, Dynamic} = U_{avg} \times P_{MAC, Dynamic}, \quad (2)$$

where $N$ is the total number of convolutional layers in the DNN, $U_i$ and $C_i$ are the utilization and compute cycles, respectively, for the $i^{th}$ layer, and $P_{MAC, Dynamic}$ is the dynamic power for a MAC unit. We also integrate an exponential leakage model for MAC (see Sec. 4.1.1 for details on MAC's power model).
We build a compact thermal model (CTM) in HotSpot for the chip-with their temperature-dependent leakage, and rerun HotSpot. This
VGG11, ResNet50, MobileNet, and GoogLeNet, along with Faster
(ASPDAC ’21, January 18–21, 2021, Tokyo, Japan Prachi Shukla, Sean S. Nemtzow, Vasilis F. Pavlidis, Emre Salman, and Ayse K. Coskun
8-bit MAC unit at 250
4.1.1 SRAM/Systolic Array MAC model. We solve a second order heat diffusion equation [25]. We model var-
3.5 Mono3D Thermal Model
We build a compact thermal model (CTM) in HotSpot for the chip-
sto 1.3
{735, 600, 500} MHz.

<table>
<thead>
<tr>
<th>Table 1: Design space for DNN accelerators.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systolic array size</td>
</tr>
<tr>
<td>16×16 to 256×256</td>
</tr>
<tr>
<td>Each SRAM size</td>
</tr>
<tr>
<td>[32, 64, 128, 256, 512, 1024, 2048, 4096] KB</td>
</tr>
<tr>
<td>Aspect ratio of the chip</td>
</tr>
<tr>
<td>0.7 to 1.1</td>
</tr>
<tr>
<td>Frequencies</td>
</tr>
<tr>
<td>[735, 600, 500] MHz</td>
</tr>
</tbody>
</table>

We use the SRAM bandwidth (bytes per cycle) generated by SCALE-Sim to decide the number of banks in SRAM. We use CACTI to
calculate the SRAM dynamic power and leakage. To estimate SRAM leakage at a finer granularity than the 10 degree default
granularity of CACTI, we fit a linear model (a linear model can accurately estimate leakage across close temperatures [23]).

We deploy a generic interconnect power model, where the inter-
connects consume 15% of the total chip dynamic power because
(i) DNNs require large amounts of memory for inputs, weights,
and outputs, and (ii) there is frequent data movement between
the systolic array and SRAMs [24]. We then reduce the interconnect
power by 10%, which is equal to Mono3D iso-performance power
savings obtained from a recent work [14].

For energy efficiency, we use system energy (\(E_{sys}\), includes both
the chip and DRAM energy), energy-delay-area product (EDAP),
energy-delay\(^2\) product (ED2P), and energy-delay product (EDP).

4.1 Experimental setup
4.1.1 SRAM/Systolic Array MAC model. We synthesize a 65 nm
8-bit MAC unit at 250 MHz using the Synopsys Design Compiler
(DC) and scale it down to 22 nm technology node. The scaled down
area, dynamic power, and the frequency are 121 \(\mu m^2\) (length = 11
\(\mu m\), 0.25 mW, and 735 MHz, respectively. We also fit a temperature-
dependent exponential leakage model for a MAC unit using data
points (temperature, leakage) from our synthesized MAC model.
Furthermore, we model 22 nm SRAMs in CACTI-6.5 and the off-chip
DRAM is based on 8 Gb LPDDR2-800 x32 chips at 400 MHz, with
8.5 Gbps bandwidth and 200 pJ/byte energy consumption [29].

4.1.2 Constraints and Design Space. We set our chip footprint bud-
get to 8 mm\(^2\), desired systolic array size between 16×16 [2] and
256×256 (similar to Google’s Tensor Processing Unit), total allowed
SRAM size to 24 MB, thermal budget to 80\(^\circ\)C, and the maximum
whitespace allowed to 1% of the chip footprint. In addition to the
chip frequency of 735 MHz, we include 600 MHz and 500 MHz in
our search space. We set a constraint on maximum performance
loss of \(\leq 10\%\) w.r.t. the design with the lowest latency under the
given constraints. Note that this is a user-defined parameter and can
change as required. Each SRAM has 4 banks and provides a band-
width of 256 bytes per cycle to match with the maximum SRAM
bandwidth for the given systolic array bounds as output by SCALE-
Sim. Table 1 shows the total design space for DNN accelerators, i.e.,
24.6k (3 frequencies × 8.2k accelerators) design points.

4.2 Optimizer Evaluation
4.2.1 Setup and Running Times. We launch 6 starts for each fre-
quency and each start is assigned an aspect ratio range. Each start
has 6 annealing temperatures with 35 perturbations. We ensure
convergence by observing that the optimizer does not accept worse
designs as it approaches termination. For tuning the optimizer, we
vary \(\delta\) from 0.7 to 0.92 and \(T_{start}\) from 1.446 to 4.481 (values set
empirically, based on a set of known good results). These two para-
ters control the rate and probability, respectively, with which
MSA accepts worse solutions to escape the local minima and arrive
at a near-optimal solution. Also, our optimizer can work with a
larger range of frequencies and still select a near-optimal point (this
may require launching more starts in parallel).

SCALE-Sim and HotSpot take 10-60 and 5-45 mins, respectively,
depending on the chip footprint and DNN. HiC DNNs have a higher
number of MAC operations that lead to higher power densities and
peak temperatures (more active PEs), which increase tempera-
ture-dependent leakage. Thus, these DNNs require more iterations (4-5)
to converge in HotSpot. LoC DNNs require fewer iterations (2-3)
due to fewer MAC operations [28] and lower chip power. Long
simulation times are bottlenecks to perform an exhaustive search
in our large design space and demonstrate the need for an optimizer.

4.2.2 Correctness of the Optimizer. To demonstrate the correct-
ness of our optimizer, we select a smaller design space with one
frequency (735 MHz), 0.94 to 1 aspect ratio range, under the same
constraints listed in Sec. 4.1.2. In total, there are 1,196 valid config-
urations. We evaluate the optimizer with [10, 5, 3]% performance
constraints. We select 2 DNNs, Tiny-YOLO (LoC) and VGG11 (HiC),
and compare the optimizer’s choices to those determined by an
exhaustive search. The optimizer’s parameters \(\{T_{start}, T_{finish}, \delta\}\)
for Tiny-YOLO and VGG11 are set to [1.446, 0.738611, 0.8] and [1.446,
0.885963, 0.85], respectively. The 6 starts are assigned aspect ratio
ranges: \([0.94, 0.95]\), \([0.95, 0.96]\), and so on till \([0.99, 1]\). Across all the
objectives (performance, power, energy, EDP, ED2P, and EDAP),
the near-optimal designs selected by the optimizer and the global
optimal differ by \(\leq 2\%\) in \(Obj\) values, showing close agreement.
We next discuss the temperature-aware optimization results for VGG19. Fig. 4b shows a 5% performance tradeoff w.r.t. the smallest latency accelerator has higher utilization (with same SRAM size), which leads to better performance but higher dynamic power and temperature in the systolic array tier. The inter-tier thermal coupling in Mono3D further increases the static power by 4% (despite the same SRAM size), eventually leading to a 3°C higher peak temperature. On average, HiC DNNs tradeoff 2% performance to operate under safe temperatures (Table 2).

### Optimization Results

We next discuss the temperature-aware optimization results for various objective functions. The 6 starts are assigned aspect ratio ranges: [0.7, 0.8], [0.8, 0.9], and so on till [1.2, 1.3].

#### 4.3.1 Performance

Fig. 4 shows performance versus temperature results for all the designs that our optimizer evaluates before converging to near-optimal solutions for ResNet50 and VGG19 when minimizing latency. The dashed lines are the user-defined performance and thermal constraints. The optimizer selects a 198×184 systolic array with a 4160 MB SRAM at 735 MHz for ResNet50 (Fig. 4a). The figure also shows a few points with slightly worse performance but higher temperature within the performance constraint. Those points have a slightly larger footprint (1%) with more active PEs, which results in higher power and peak temperatures. LoC DNNs have adequate thermal headroom to run on big systolic arrays at 735 MHz without sacrificing performance (see Table 2).

In contrast, HiC DNNs have a higher array utilization (due to more MAC operations) and lead to more thermal violations (due to higher chip power) compared to the LoC DNNs (e.g., VGG19 in Fig. 4b). The optimizer selects 170×214 with 4160 KB SRAM for VGG19. Fig. 4b shows a 5% performance tradeoff w.r.t. the smallest latency accelerator to obey the tight thermal budget for VGG19. The smallest latency accelerator has higher utilization (with same SRAM size), which leads to better performance but higher dynamic power and temperature in the systolic array tier. The inter-tier thermal coupling in Mono3D further increases the static power by 4% (despite the same SRAM size), eventually leading to a 3°C higher peak temperature. On average, HiC DNNs tradeoff 2% performance to operate under safe temperatures (Table 2).

#### 4.3.2 Power

Fig. 5 shows performance, power, and temperature tradeoffs for ResNet50 and VGG19. We see at low total chip power (< 1 W), peak temperatures can be high (80°C for ResNet50 and 82°C for VGG19). Here, the DNNs are running on smaller chip accelerators violate the performance constraint and thus, are not selected by the optimizer. Similarly, the optimizer selects 735 MHz designs for the other LoC DNNs (Table 2).

For VGG19, the optimizer selects a 180×202 systolic array with 2080 KB SRAM at 600 MHz (see Fig. 5b). At 735 MHz, the most power-efficient design under the user-specified constraints is almost of the same size as the selected design (≈0.99x) with a similar array utilization and same SRAM size. The higher dynamic power (due to faster PEs) causes higher temperature in the systolic array tier, which further increases the static power by 9% due to inter-tier thermal coupling (despite the same size of the SRAM), eventually resulting in a 7°C higher peak temperature. Similar, a 600 MHz...
We propose a design optimization method that yields near-optimal HiC DNNs (e.g., VGG19) with more PEs running in parallel and thus making up for the performance loss w.r.t. 735 MHz designs. On average, our optimizer achieves 1.2x energy and 1.1x area savings, with a performance loss of 5.3% across all the DNNs. Finally, selections made by the optimizer for minimizing EDAP achieve up to 2x chip footprint and 1.6x $E_{\text{sys}}$ savings, by sacrificing up to 9.7% latency (average: 1.2x, 1.4x, 5.5%, respectively).

5 CONCLUSION

We propose a design optimization method that yields near-optimal efficient DNN accelerators based on Mono3D under user-specified thermal and performance constraints. Based on tradeoff analysis, several conclusions can be drawn: (i) HiC DNNs with higher dynamic power result in higher temperature, which further increases leakage due to inter-tier thermal coupling, eventually resulting in thermal violations. As a result, HiC DNNs have to tradeoff performance to operate under safe temperatures. (ii) Although we can add more SRAM and PEs (i.e., larger systolic array) to utilize the two tiers in a given chip footprint, power efficiency can drop (even at lower frequencies) due to (a) higher dynamic power (more active PEs) and (b) higher SRAM static power, as a result of both SRAM size and inter-tier thermal coupling in Mono3D across all DNNs. (iii) HiC DNNs (e.g., VGG19) with more PEs running in parallel can benefit from running at lower frequency, along with Mono3D power savings, thereby achieving higher energy efficiency.

ACKNOWLEDGMENTS

This work is partially funded by National Science Foundation under grants CCF 1910075/1909027.

REFERENCES