DFA-DRIVE: A Cross-layer Delay Fault Analysis and Optimization Framework for Robust Multi-Task Driving Perception

Nrusinga Charan Gantayat[1], Philip Jacobson[2], Matthew Marinella[1], Ben Feinberg[2], Jeff Zhang[1]
[1]Arizona State University, [2]Sandia National Laboratories
{ngantaya, m, jeffzhang}@asu.edu, {pljacob, bfeinbe}@sandia.gov

Abstract

Autonomous driving systems increasingly rely on deep neural network (DNN) based multi-task perception models for reliable, real time scene understanding. At nanoscale technology nodes, these workloads are highly susceptible to timing errors arising from temperature fluctuations, voltage droop, and device aging. Among these, temperature poses a critical challenge. Prolonged high thermal stress exacerbates delay faults, degrading perception accuracy and endangering safety critical operation.

We present **DFA-DRIVE**, a cross-layer *Delay Fault Analysis Framework for Autonomous Driving* that bridges circuit-level timing analysis with system level resilience evaluation. DFA-DRIVE quantifies how temperature induced timing failures propagate through object detection, drivable area segmentation, and lane line segmentation, exposing task level reliability bottlenecks.

Building on DFA-DRIVE, we introduce **DFA-OPT**, an adaptive hardware mapping algorithm that dynamically reassigns systolic array (SA) resources based on DNN layer and application level thermal sensitivity. Targeting the automotive reliability envelopes of **AEC-Q100 Grade 0 (–40 °C to 150 °C)** and **Grade 1 (–40 °C to 125 °C)**, DFA-OPT restores near baseline accuracy of small, high reliable SA (e.g., 4×4) even when large SA (e.g., 256×256) experience accuracy drops of upto 4% at 150 °C, achieving comparable accuracy with fewer computation cycles.

1 Introduction

Autonomous systems are rapidly proliferating across robotics, transportation, and intelligent infrastructure, driving demand for highly reliable real time perception. Autonomous driving represents one of the most safety critical applications, requiring accurate, high quality scene understanding with strong precision and recall guarantees. To meet these requirements, modern vehicles rely on increasingly complex multi-task deep neural networks (DNNs) that jointly perform object detection, drivable area segmentation, and lane line segmentation. Representative models such as YOLOP [23] integrate these tasks into a unified architecture, providing both semantic and geometric cues essential for downstream planning and control.

Perception models involve massive MAC operations and are therefore deployed on high-throughput DNN accelerator hardware such as systolic arrays (SA) [2, 9, 15]. However, these accelerators are fabricated in advanced semiconductor nodes where timing margins are extremely tight. Even small timing induced perturbations can corrupt intermediate computations, and such perception errors can cascade into unsafe control decisions [13].

Compounding this challenge, automotive grade embedded platforms must sustain reliable operation under extreme thermal and lifetime stress [22]. During prolonged outdoor and high load driving conditions, junction temperatures routinely rise to 125°C and may reach 150°C, approaching **AEC-Q100 Grade 1** and **Grade 0** limits [5]. At nanoscale technology nodes, such elevated temperatures significantly degrade carrier mobility, increase threshold voltage variation, and exacerbate interconnect delay [3, 20]. These temperature induced effects manifest as delay faults in logic pipelines, increasing the likelihood of timing violations in key computational units such as multiply accumulate (MAC) cells.

These timing induced faults are particularly challenging because they manifest as *transient* errors appearing sporadically depending on data patterns, input timing, and local temperature variation. As a result, the resulting silent data corruptions (SDCs) [17] rarely exhibit deterministic signatures and are extremely difficult to detect or isolate through conventional monitoring. Their impact propagates across the hardware software boundary: low level MAC corruptions alter intermediate feature maps, which in turn perturb high level perception outputs, creating a deep impedance mismatch across the cross-layer stack.

Studying these faults in a principled way typically requires gate-level simulation with SDF annotated timing delays to capture temperature dependent behavior. However, executing full multi-task perception models at gate level is prohibitively slow, often requiring hours or days per inference rendering exhaustive characterization infeasible for realistic workloads and thermal conditions.

Purely software based evaluation is insufficient. Existing fault-injection frameworks typically rely on random bit flips [14, 16] or simplified statistical models [1, 18], which fail to reflect the temperature dependent timing behavior of real hardware. Consequently, they cannot model how delay faults emerge at the MAC level, propagate through convolutional layers, or differentially impact tasks in multi head perception models. Moreover, no prior work enables *fast software replay* driven by delay fault outcomes derived directly from hardware timing analysis. This lack of hardware aligned replay creates a critical visibility gap: software cannot estimate true layer wise error probabilities nor quantify how thermal timing faults distort intermediate feature maps and degrade detection, drivable area segmentation, and lane line estimation accuracy.

We propose DFA-DRIVE (Fig. 2), a cross-layer framework that models and mitigates temperature induced delay faults in multi task perception workloads. DFA-DRIVE couples gate level timing analysis with software level replay and system level optimization

DAC '26, Long Beach, CA, USA

ACM ISBN 979-8-4007-2254-7/2026/07
https://doi.org/10.1145/3770743.3804192

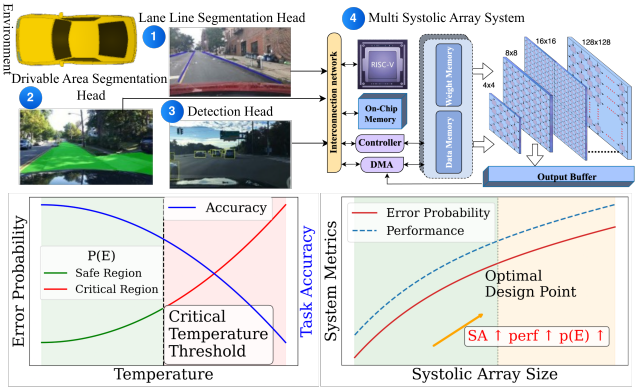

Figure 1: Illustration of temperature induced timing error vulnerability in autonomous driving perception. We study the safety and resilience of all the three tasks.

to capture, propagate, and compensate for thermal timing errors across the full computing stack. To enable fast and hardware-aware software replay, DFA-DRIVE integrates a lightweight software simulator with an HNSW based predictor (Fig. 2, ⑩) that performs vector search over an indexed database of timing logs, retrieving nearest neighbor hardware outcomes to estimate layer wise error probabilities with high fidelity.

Our analysis reveals: while larger SAs deliver higher throughput, they also exhibit substantially greater vulnerability under elevated temperature stress (Fig: 1). For weight-stationary SAs, this trend arises from two factors. First, as array size increases, the accumulator often becomes the critical timing path. Second, the probability of timing failures in MAC units increases with array dimension (N); a single timing error in an upper MAC can propagate downward through the column, compounding and amplifying accumulated partial-sum errors across subsequent units. This heightened susceptibility leads to severe degradation across drivable area segmentation, object detection, and lane line segmentation tasks. This underscores a critical insight: although scaling to larger arrays is a common approach to improve performance, reliability under real world thermal conditions is often overlooked.

To address this gap, DFA-DRIVE introduces DFA-OPT, an optimization algorithm that reallocates each DNN layer to the most reliable SA size under delay uncertainty, jointly balancing execution cycles and task accuracy. Although designed primarily for thermal delay fault characterization, the methodology naturally extends to voltage droops and process variation induced delay shifts.

This paper makes the following contributions:

- **End-to-end thermal fault characterization.** We design DFA-DRIVE, an integrated flow that couples a SystemVerilog-based weight-stationary systolic array with a Python co-simulation environment. The framework performs gate-level, timing-annotated analysis to capture MAC-level errors under different temperatures, and propagates these to task-level accuracy outcomes for multi-task perception models.
- **Fast, realistic fault injection.** We develop a software-level injector aligned with hardware-derived error profiles, overcoming the prohibitive cost of repeated gate-level simulation. Using a Hierarchical Navigable Small World (HNSW) [12] index constructed from MAC-level timing logs, the injector preserves both error rates and corrupted value distributions, enabling faithful replay of delay faults at software speed.
- **Temperature-aware workload mapping.** We propose an algorithm that assigns each DNN layer to an appropriate systolic array size under a targeted temperature. By balancing execution cycles against per-layer error vulnerability, it identifies configurations that maximize performance while bounding accuracy degradation.
- **Comprehensive evaluation on YOLOP.** We evaluate DFA-DRIVE on YOLOP, a representative multi-task driving model. Our results present accuracy-versus-temperature curves across tasks, layer-wise vulnerability maps, and accuracy recovery enabled by adaptive systolic array mapping. At 150°C (Grade 0), DFA-OPT improves drivable area segmentation by +2.1%, lane line by +3.2%, and detection by +5.5%, while reducing execution cycles by up to 92% compared to smaller, less efficient arrays.

2 Background and Related Work

2.1 Perception-Centric Autonomous Systems

Autonomous driving workloads rely on perception driven DNNs to interpret multimodal sensor inputs such as cameras, LiDAR, and radar for real time scene understanding. Given an input x, a DNN F_θ produces task outputs $\hat{y} = \{\hat{y}_{\text{det}}, \hat{y}_{\text{da}}, \hat{y}_{\text{lane}}\} = F_\theta(x)$, where $\hat{y}_{\text{det}}$ denotes object detections, $\hat{y}_{\text{da}}$ drivable area segmentation, and $\hat{y}_{\text{lane}}$ lane line predictions. Modern multi task perception models such as YOLOP [23], BEVFusion [11], and BEVDet [8] adopt shared visual encoders with lightweight, task specific decoders to efficiently handle the diverse perception objectives in autonomous driving.

Multi task learning enables these related tasks to share intermediate representations, improving feature richness while reducing computational redundancy. Architectures like YOLOP [23] demonstrate that jointly training detection, drivable area segmentation, and lane line estimation can lead to mutually beneficial feature sharing and more coherent scene interpretation. The simplicity of a shared encoder decoder structure also allows efficient end to end training without the need for complex cross task message passing or iterative optimization strategies often found in earlier multi task models. Although we evaluate YOLOP as a representative workload, the underlying methodology is broadly applicable to perception driven autonomous computing platforms.

2.2 Temperature-Induced Delay Uncertainty

At nanoscale process nodes, **temperature variation** becomes a dominant source of delay uncertainty, affecting carrier mobility, threshold voltage, and wire resistance [10, 19]. Automotive grade systems must operate reliably across thermal ranges from **AEC-Q100 Grade 1** (−40°C to 125°C) to **Grade 0** (−40°C to 150°C) [5] where elevated junction temperatures exacerbate setup time violations and timing jitter in logic pipelines.

We model these effects as an exogenous variation state δ_T. A timing error at MAC i in layer ℓ occurs when

$$d_i(\delta_T, \text{context}) > T_{\text{clk}}(n_\ell),$$

where d_i represents the temperature dependent path delay and $T_{\text{clk}}(n_\ell)$ is the clock period for array size n_ℓ. Architectural analysis

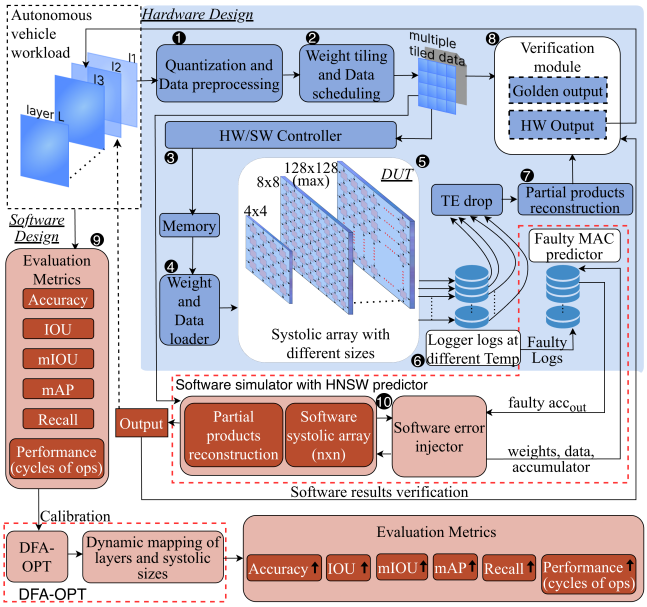

Figure 2: Overview of the proposed DFA-DRIVE framework.

of small delay faults has been explored previously via the DelayAVF framework [6]. In DFA-DRIVE, such timing violations are detected through delay annotated gate level simulations at different temperatures: if a MAC output fails to settle before the clock edge, the incorrect value is propagated down the column and also logged using a Logger (Fig. 2, ❻).

2.3 Timing Speculation and Thermal Resilience in Accelerators

Recent system level resilience frameworks have explored the behavioral impact of hardware faults in autonomous systems. MAVFI [7] analyzes how silent data corruptions (SDCs) affect mission level metrics in UAVs and employs anomaly based detection and recovery to enhance robustness. BERRY [21] investigates low voltage induced bit errors in reinforcement learning based UAV controllers and demonstrates robust learning strategies that maintain mission performance under aggressive voltage scaling. Although these works focus on UAV autonomy rather than DNN inference accelerators, they underscore the importance of linking low level hardware faults to high level behavioral degradation in safety critical systems.

In contrast, our objective is to map DNN layers to SA sizes under delay uncertainty, guided by per MAC error probabilities obtained through our HNSW based predictor. The goal is not merely fault detection, but to jointly optimize performance (cycles) and reliability (accuracy impact) for autonomous perception tasks.

3 DFA-DRIVE Framework

We introduce the DFA-DRIVE framework (Fig. 2) in this section.

3.1 SPICE Based Temperature–Delay Modeling

To accurately quantify the influence of temperature on circuit timing, we perform transistor-level **HSPICE** simulations using the ASAP7 predictive 7 nm PDK [4]. While modern FinFET devices

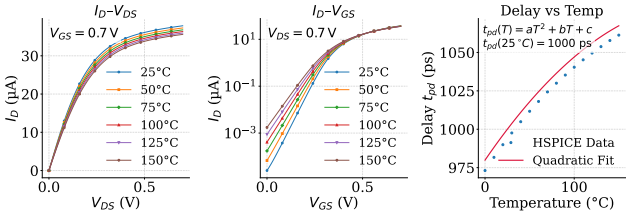

Figure 3: MOSFET output and transfer characteristics across temperature, and calibrated delay temperature model derived from HSPICE using ASAP7 PDK.

exhibit increased leakage, mobility degradation, and threshold voltage drift above 125–150°C, we intentionally extend the sweep to 150°C for worst case, exploratory characterization of temperature delay sensitivity. This enables a more robust evaluation of timing behavior in safety critical automotive environments.

We construct a calibrated ring oscillator surrogate using ASAP7 transistors and tune it such that its propagation delay is exactly 1000 ps at 25°C, matching the nominal timing budget assumed in our framework. The oscillator is then simulated from 0°C to 150°C in HSPICE, and the resulting delays are fitted to a quadratic model $d(T) = aT^2 + bT + c_{\text{cal}}$, with calibrated parameters $(a, b, c_{\text{cal}}) = (1.55{\times}10^{-3}, 0.84, 972.0)$ ensuring $d(25°\text{C}) = 1000$ ps. A timing error is flagged whenever the temperature dependent delay exceeds the available clock period: $d(T) > T_{\text{clk}}$.

This SPICE calibrated curve in Fig. 3 provides the physical foundation for temperature aware timing error injection in DFA-DRIVE. By mapping temperature directly to delay at the transistor level, it enables realistic timing error modeling across thermal conditions.

3.2 DFA-DRIVE Execution Flow

The DFA-DRIVE framework establishes a tightly coupled hardware software co simulation flow that links post synthesis timing evaluation with system level perception accuracy. As illustrated in Fig. 2, the execution begins with quantization and preprocessing (❶), where input activations are normalized and converted to 8-bit precision to match embedded inference constraints and reduce on chip memory footprint. These quantized tensors are then partitioned into spatial and channel-wise tiles (❷) according to the target systolic array dimensions, preparing the data for systolic scheduling and hardware mapping.

The hardware software controller (❸) orchestrates this dataflow by managing on chip buffers, scheduling layer execution, and synchronizing transactions between the software driver and the RTL accelerator. A dedicated SystemVerilog testbench acts as the weight and data loader (❹), injecting synchronized operands into the design under test (DUT) with cycle level determinism. The robust DUT (❺), capable of dynamically adjusting to systolic arrays ranging from 4×4 to 256×256, is synthesized using Synopsys Design Compiler and annotated with SDF. It is instrumented with a calibrated temperature delay model derived from HSPICE. During simulation, each MAC output is checked against the sampling edge; late arrivals are flagged as timing errors, effectively emulating high temperature operation from a single golden simulation.

A cycle accurate logger (❻) records every weight, activation, and partial sum, together with their delay annotations at different temperatures. These logs feed into the partial product reconstruction module (❼), which regenerates correct or corrupted outputs for arbitrary temperature points. The reconstructed results are compared with *PyTorch golden* references in the verification module (❽), quantifying degradation in accuracy, IoU, mIoU, and mAP as timing faults propagate through the systolic array. These logs also form the foundation for constructing a data driven software error model that replaces hardware level timing simulations.

3.3 Software Simulation and Error Modeling

To achieve hardware level fidelity at software speed, DFA-DRIVE integrates a Python based systolic-array simulator (Fig. 2, ❿) that mirrors the RTL dataflow and multiply accumulate (MAC) sequencing. Each software MAC operation receives quantized inputs (w, a) and produces partial sums that evolve cycle by cycle matching the timing granularity of the logged RTL simulation. On top of this engine, a data driven fault model is constructed to emulate temperature dependent timing errors.

HNSW Based Database Construction. Cycle level logs collected from hardware simulation are used to build a high dimensional feature database indexed by a **Hierarchical Navigable Small World (HNSW)** graph [12]. Each database entry corresponds to a unique MAC instance and is represented as

$$\mathbf{x}_i = [\, w_i,\ a_i,\ acc_{\text{in},i},\ a_i^{(t-1)},\ acc_i^{(t-1)} \,], \tag{1}$$

with metadata $\{T_i, \text{fault}_i, y_i^{\text{fault}}, y_i^{\text{clean}}\}$. Here, $\text{fault}_i \in \{0, 1\}$ denotes whether a MAC violates setup time at temperature T_i.

The HNSW index is constructed using two hyperparameters *graph degree M* and the construction parameter `efConstruct` which jointly define the density and connectivity of the multi layer small world graph. A higher M increases the number of bidirectional links per node, enhancing recall and search robustness at the cost of greater memory footprint (scaling approximately linearly with M). Similarly, a larger `efConstruct` expands the candidate pool during graph insertion, improving global neighbor ordering and search accuracy but increasing build time. Both parameters thus act as tunable trade offs between index quality, memory usage, and construction overhead. As recommended in [12], we performed a parameter sweep of M and `efConstruction` to identify the optimal configuration for our dataset, selecting the combination that yielded the highest recall without exceeding our memory budget.

Query Phase and Deterministic Fault Prediction. During software inference, each incoming MAC tensor from the systolic array simulator is expressed as a feature vector $\mathbf{x}_q$ at the target operating temperature T. The predictor searches the HNSW index for its nearest neighbor:

$$j^* = \arg\min_j \|\mathbf{x}_q - \mathbf{x}_j\|_2, \quad j \in \mathcal{D}_{\text{HNSW}}(T).$$

The retrieved neighbor j^* carries the observed label fault_{j^*} and the corresponding outputs $(y_{j^*}^{\text{fault}}, y_{j^*}^{\text{clean}})$. The software simulator then deterministically classifies and substitutes the MAC output as:

$$\tilde{S}_{ij}^{(t)} = \begin{cases} y_{j^*}^{\text{fault}}, & \text{if fault}_{j^*} = 1, \\ S_{ij}^{(t)}, & \text{if fault}_{j^*} = 0, \end{cases}$$

where $S_{ij}^{(t)}$ is the clean PyTorch computed output. This procedure enforces a one to one mapping between each runtime MAC operation and its most similar hardware observed instance, thereby preserving the temperature dependent error characteristics without requiring stochastic sampling or global $p(E, T)$ tables.

Software Hardware Equivalence. Because the Python systolic array follows identical dataflow and accumulation order as the RTL design, each substitution in $\tilde{S}_{ij}^{(t)}$ corresponds to a realistic timing violation at the same spatial temporal position. This tight correspondence enables large scale temperature sweeps and cross layer reliability studies at negligible runtime cost. Although the framework stores HNSW graphs across operating temperatures incurrs moderate memory overhead, the resulting vector search is extremely fast, offering microsecond level lookups that are orders of magnitude cheaper than rerunning gate level timing simulations. Users may further extend the framework by incorporating mitigation mechanisms such as TE-Drop [25] or spare MAC remapping directly atop the deterministic HNSW substitution flow.

3.4 Temperature-Aware DFA-OPT Algorithm

As depicted as the "DFA-OPT" block in Fig. 2, DFA-OPT produces an explicit per-layer mapping $s(\ell)$ that assigns each DNN layer ℓ to an systolic array size for a given operating temperature T. The objective is to minimize total execution cycles while satisfying a temperature-dependent reliability budget derived from DFA-DRIVE's fault characterization. Using DFA-DRIVE's hardware-aligned fault profiling, we observe that timing error probability varies significantly across SA sizes: larger arrays exhibit higher vulnerability due to increased error propagation and accumulation. As a result, DFA-OPT typically assigns: (1) large SA sizes to timing-robust layers (low error probability), and (2) smaller SA sizes to a small subset of timing-sensitive layers that dominate the overall reliability risk.

In practice, only a small fraction of layers are remapped to smaller SAs, while most layers remain on the largest SA to preserve throughput. This behavior is consistent across representative temperatures and workloads and reflects the fact that timing faults are highly non-uniform across layers.

Problem Formulation: Let $\mathcal{L}$ denote the set of network layers and $\mathcal{S}$ the available systolic array sizes (e.g., {4×4, 8×8, . . . , 256×256}). For each layer $\ell \in \mathcal{L}$ and array size $s \in \mathcal{S}$: $C[\ell, s]$ represents the total cycle count to execute layer ℓ on array s, estimated from the software scheduler and validated via RTL simulation. $P_e[\ell, s, T]$ denotes the timing error probability of that layer at temperature T, obtained from HNSW-based fault analysis using the SPICE-calibrated delay model. A mapping $s : \mathcal{L} \to \mathcal{S}$ assigns each layer to a specific array size, producing an overall execution cost:

$$C_{\text{total}}(s) = \sum_{\ell \in \mathcal{L}} C[\ell, s(\ell)].$$

To quantify reliability under thermal variation, we define the *Top-K thermal risk*:

$$R_{\text{top}K}(s, T) = \sum_{\ell \in \text{Top-}K \text{ by } P_e[\ell, s(\ell), T]} P_e[\ell, s(\ell), T],$$

which captures the cumulative fault probability of the K most vulnerable layers at temperature T. This metric reflects the practical

observation that perception accuracy is often dominated by a small subset of high sensitivity layers.

The optimization problem is then posed as:

$$\min_{s:\mathcal{L}\to\mathcal{S}} C_{\text{total}}(s) \quad \text{s.t.} \quad R_{\text{top}K}(s,T) \le B(T),$$

where $B(T)$ is a temperature dependent **reliability budget** the maximum allowable aggregated failure probability that still preserves system level accuracy. This budget is calibrated from empirical sweeps of layer to array mappings or task to array mapping at each temperature, identifying the fastest configuration whose accuracy degradation remains below a tolerance τ (2% in our case).

Algorithm 1 DFA-OPT: SA Assignment under Temperature Faults

Inputs: Layers $\mathcal{L}$, SA sizes $\mathcal{S}$, cycle costs $C[\ell, s]$, error probabilities $P_e[\ell, s, T]$, accuracy tolerance τ, Top-K.

Outputs: Mapping $s(\ell)$ minimizing total cycles with $R_{\text{top}K}(s,T) \le B(T)$.

1: **Phase A: Calibrate** $B(T)$**:**
2: Evaluate candidate mappings at temperature T; among those with accuracy drop $\le \tau$, select the fastest mapping $s^\star$.
3: Set $B(T) \leftarrow R_{\text{top}K}(s^\star, T)$.
4: **Phase B: Thermal Aware Optimization:**
5: Initialize $s(\ell) \leftarrow \max(\mathcal{S})$ for all $\ell \in \mathcal{L}$.
6: **while** $R_{\text{top}K}(s,T) > B(T)$ **do**
7: Identify Top-K risky layers.
8: **for** each candidate shrink $\ell : s \to s'$ **do**
9: Compute reliability reduction $\Delta R = R^{\text{before}}_{\text{top}K} - R^{\text{after}}_{\text{top}K}$.
10: Compute cycle increase $\Delta C = C[\ell, s'] - C[\ell, s]$.
11: Compute efficiency ratio $\rho = \Delta R / \Delta C$.
12: **end for**
13: Apply the move maximizing ρ (break ties by larger ΔR, then smaller ΔC).
14: **end while**

Algorithm Overview: Algorithm 1 summarizes the two-phase optimization process. **Phase A: Reliability-Budget Calibration (Lines 1–3).** At each temperature T, a small set of candidate mappings is evaluated. Among the configurations that satisfy the accuracy constraint (drop $\le \tau$), the fastest mapping $s^\star$ is selected, and its Top-K risk $R_{\text{top}K}(s^\star, T)$ defines the reliability budget $B(T)$.

Phase B: Iterative Thermal Aware Assignment (Lines 4–14). All layers are initially assigned to the largest array size (Line 5), which minimizes cycle count but often exceeds $B(T)$. While the current mapping violates the budget (Line 6), DFA-OPT progressively reassigns risky layers to smaller arrays. For each candidate shrink (Lines 8–11), the algorithm computes:

$$\Delta R = R_{\text{before}} - R_{\text{after}}, \quad \Delta C = C_{\text{after}} - C_{\text{before}}, \quad \rho = \frac{\Delta R}{\Delta C},$$

where ρ quantifies the **reliability gain per unit cycle cost**. The move with the highest ρ is selected (Line 13), prioritizing the largest reduction in Top-K risk for the smallest throughput penalty. The process terminates once $R_{\text{top}K}(s,T) \le B(T)$ (Line 6), resulting in an optimized, temperature-aware configuration $s(\ell)$.

In summary, DFA-OPT adaptively allocates array resources per layer as a function of temperature, guided by timing error probabilities. This enables performance-reliability optimization across AEC-Q100 grades with minimal throughput loss.

4 Evaluation

We evaluate DFA-DRIVE on YOLOP [23], a multi task perception model that performs object detection, drivable area segmentation, and lane line segmentation.

Dataset. All experiments use the BDD100K dataset [24], which contains annotated urban driving scenes across diverse weather, lighting, and traffic conditions.

Metrics. Detection is evaluated using recall, whereas lane line & drivable area tasks use accuracy. At the hardware level, we measure the per MAC error probability $P(E,T)$, which represents the column wise distribution of MAC timing failures for a given SA size.

4.1 Validation of Software Error Injector and HNSW Predictor

Per-layer error probability consistency. Figure 4 shows that the predicted error probability $p(E)$ closely matches hardware measurements across layers and temperatures. Minor overestimation in a few layers provides conservative fault coverage. The model accurately captures both the magnitude and temperature dependence of timing errors.

Task-level fidelity. Using these predictions, the software injector reproduces hardware induced accuracy degradation across YOLOP tasks. As shown in Fig. 4, accuracy temperature trends for representative layers (31, 40, and 20) align closely with hardware injection, confirming that the HNSW based injector preserves system level fault behavior and enables scalable thermal reliability evaluation without rerunning post synthesis simulations.

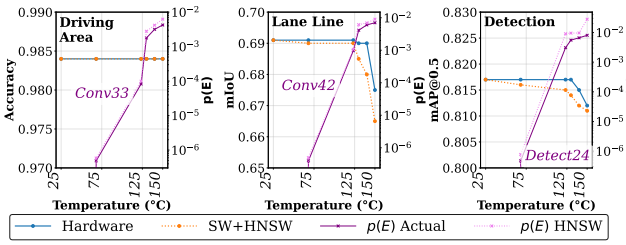

Figure 4: Hardware vs. software fault injection results under temperature variation.

4.2 Error Probability Across Systolic Array Sizes

We report task level median error probabilities $P(E,T)$ for systolic arrays ranging from 4×4 to 256×256 across temperature points. As shown in Fig. 5, error probability increases sharply with both array size and temperature: smaller arrays maintain negligible $P(E,T)$ even at 150°C, whereas larger arrays exhibit higher aggregate failure rates. Detection layers are most sensitive to this effect, while Lane line and Drivable area heads remain relatively stable until higher temperatures. These results highlight the fundamental throughput reliability trade off that motivates for DFA-OPT.

4.3 Error Probability Across DNN Layers

Fig. 6 shows layer-wise $P(E,T)$ for YOLOP with a 128×128 SA. Thermal vulnerability varies significantly across layers. This heterogeneity underscores the inefficiency of fixed SA mappings, motivating DFA-OPT to reassign high risk layers to smaller, more reliable arrays while using larger arrays for thermally robust ones.

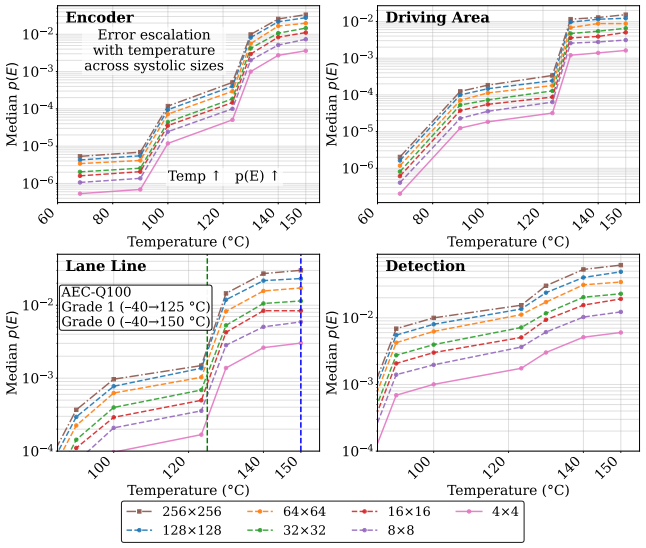

Figure 5: Median $P(E,T)$ across SA sizes (4×4–256×256) and temperature. Larger arrays exhibit higher fault rates under thermal stress.

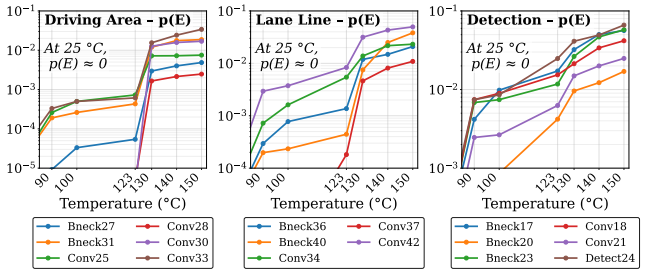

Figure 6: Per layer $P(E,T)$ for YOLOP using a 128×128 SA.

4.4 DFA-OPT Accuracy and Performance Optimization

We evaluate task level accuracy across the full temperature range focusing on the basline TE-Drop (static single SA mapping) vs. DFA-OPT (TE-Drop + dynamic multi-SA mapping). As shown in Fig. 7, smaller arrays (e.g., 4×4) remain thermally robust up to 123°C but incur significantly higher cycle counts (Fig. 8). In contrast, larger arrays deliver higher throughput but experience sharp accuracy degradation beyond high temperature corners. By dynamically resizing layers based on thermal sensitivity, DFA-OPT preserves baseline (4×4) accuracy within a 0–2% tolerance band (Fig. 7) while reducing overall computation time (Fig. 8), achieving an improved balance between throughput and reliability across automotive temperature grades.

5 Conclusion

We presented **DFA-DRIVE** for analyzing and mitigating timing induced reliability challenges in DNN accelerators. By combining hardware fault logs with software level error injection, we quantified per layer error probabilities and validated an HNSW based predictor for scalable replay of hardware induced errors. Building on these insights, we proposed **DFA-OPT**, an adaptive layer to array mapping algorithm that balances performance and reliability across

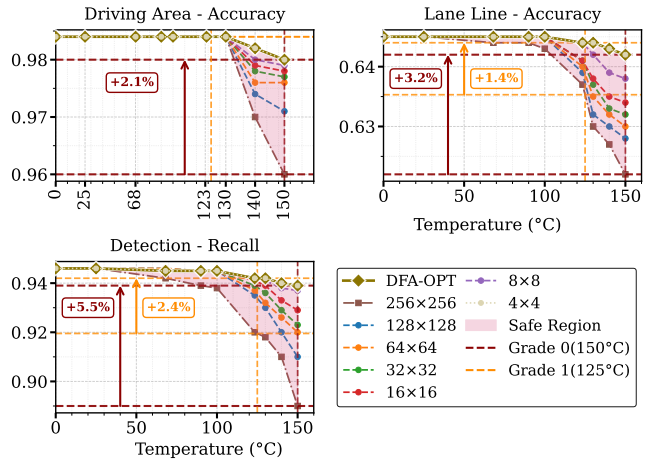

Figure 7: Accuracy degradation under temperature stress vs. DFA-OPT with TE-Drop. DFA-OPT preserves accuracy across tasks within a 0–2% tolerance up to 150°C.

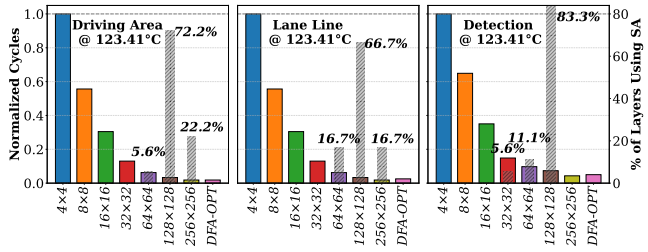

Figure 8: Comparison of execution cycles for different SA sizes and the dynamic SA size assignments selected by DFA-OPT.

temperature corners. At 150°C, DFA-OPT improves driving area segmentation accuracy by **+2.1%**, lane line segmentation by **+3.2%**, and object detection by **+5.5%** compared with high performance systolic arrays (larger SAs), while significantly reducing computation cycles upto **92%** relative to high reliability arrays (smaller SAs). Future work will extend this framework to GPU based fault modeling and explore timing error resilience in large language models (LLMs) to generalize DFA-DRIVE beyond vision workloads.

Acknowledgments

This article has been authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan https://www.energy.gov/downloads/doe-public-access-plan.

References

[1] Subho S Banerjee, James Cyriac, Saurabh Jha, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. 2019. Towards a bayesian approach for assessing fault tolerance of deep neural networks. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks–Supplemental Volume (DSN-S)*. IEEE, 25–26.

[2] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. 2016. Eyeriss: a spatial architecture for energy-efficient dataflow for convolutional neural networks. *ACM SIGARCH computer architecture news*, 44, 3, 367–379.

[3] Zehua Chen, Hei Wong, Yan Han, Shurong Dong, and BL Yang. 2014. Temperature dependences of threshold voltage and drain-induced barrier lowering in 60 nm gate length mos transistors. *Microelectronics Reliability*, 54, 6-7, 1109–1114.

[4] Lawrence T Clark, Vinay Vashishtha, Lucian Shifren, Aditya Gujja, Saurabh Sinha, Brian Cline, Chandarasekaran Ramamurthy, and Greg Yeric. 2016. Asap7: a 7-nm finfet predictive process design kit. *Microelectronics Journal*, 53, 105–115.

[5] Automotive Electronics Council. 2023. Failure Mechanism Based Stress Test Qualification for Integrated Circuits. Tech. rep. AEC-Q100 Rev-J. See page 3 for operating temperature grades (Grade 1 and Grade 0). Automotive Electronics Council. http://www.aecouncil.com/Documents/AEC_Q100_Rev_J_Base_Document.pdf.

[6] Peter W Deutsch, Vincent Quentin Ulitzsch, Sudhanva Gurumurthi, Vilas Sridharan, Joel S Emer, and Mengjia Yan. 2024. Delayavf: calculating architectural vulnerability factors for delay faults. In *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 231–245.

[7] Yu-Shun Hsiao, Zishen Wan, Tianyu Jia, Radhika Ghosal, Abdulrahman Mahmoud, Arijit Raychowdhury, David Brooks, Gu-Yeon Wei, and Vijay Janapa Reddi. 2023. Mavfi: an end-to-end fault analysis framework with anomaly detection and recovery for micro aerial vehicles. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1–6.

[8] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. 2021. Bevdet: high-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*.

[9] Norman P Jouppi et al. 2017. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, 1–12.

[10] Xiaochun Li, Jialing Tong, and Junfa Mao. 2010. Temperature-dependent device behavior in advanced cmos technologies. In *2010 International Symposium on Signals, Systems and Electronics*. Vol. 2. IEEE, 1–4.

[11] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. 2022. Bevfusion: multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*.

[12] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42, 4, 824–836.

[13] Khan Muhammad, Amin Ullah, Jaime Lloret, Javier Del Ser, and Victor Hugo C De Albuquerque. 2020. Deep learning for safe autonomous driving: current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 22, 7, 4316–4336.

[14] Cheng Qian, Ming Zhang, Yuanping Nie, Shuaibing Lu, and Huayang Cao. 2023. A survey of bit-flip attacks on deep neural network and corresponding defense methods. *Electronics*, 12, 4, 853.

[15] Tejas Raja. 2024. Systolic array data flows for efficient matrix multiplication in deep neural networks. *arXiv preprint arXiv:2410.22595*.

[16] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. 2019. Bit-flip attack: crushing neural network with progressive bit search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1211–1220.

[17] Elvis Rojas, Diego Pérez, and Esteban Meneses. 2022. Exploring the effects of silent data corruption in distributed deep learning training. In *2022 IEEE 34th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. IEEE, 21–30.

[18] Annachiara Ruospo et al. 2023. Assessing convolutional neural networks reliability through statistical fault injections. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1–6.

[19] Sachin S Sapatnekar. 2011. Overcoming variations in nanometer-scale technologies. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 1, 1, 5–18.

[20] Smruti R Sarangi, Brian Greskamp, Radu Teodorescu, Jun Nakano, Abhishek Tiwari, and Josep Torrellas. 2008. Varius: a model of process variation and resulting timing errors for microarchitects. *IEEE Transactions on Semiconductor Manufacturing*, 21, 1, 3–13.

[21] Zishen Wan, Nandhini Chandramoorthy, Karthik Swaminathan, Pin-Yu Chen, Vijay Janapa Reddi, and Arijit Raychowdhury. 2023. Berry: bit error robustness for energy-efficient reinforcement learning-based autonomous systems. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[22] Wei-Chun Wang, Shida Zhang, Sudarshan Sharma, Minah Lee, and Saibal Mukhopadhyay. 2024. Measurement of aging effect on an analog computing-in-memory macro in 28nm cmos. In *2024 IEEE International Reliability Physics Symposium (IRPS)*. IEEE, 1–4.

[23] Dong Wu, Man-Wen Liao, Wei-Tian Zhang, Xing-Gang Wang, Xiang Bai, Wen-Qing Cheng, and Wen-Yu Liu. 2022. Yolop: you only look once for panoptic driving perception. *Machine Intelligence Research*, 19, 6, 550–562.

[24] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. Bdd100k: a diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

[25] Jeff Zhang, Kartheek Rangineni, Zahra Ghodsi, and Siddharth Garg. 2018. Thundervolt: enabling aggressive voltage underscaling and timing error resilience for energy efficient deep learning accelerators. In *Proceedings of the 55th Annual Design Automation Conference*, 1–6.