Department of Computer Science & Engineering
University of Engineering and Technology

Scalable Neural Architectures for Heterogeneous Edge Devices

A Doctoral Dissertation Proposal

Candidate:
Alex J. Mercer
Department of CSE
alex.mercer@uet.edu

Advisory Committee:
Prof. Sarah Jenkins (*Advisor*)
Dr. Marcus Aurelius (*Co-advisor*)

August 4, 2026

1 Abstract

As deep neural networks (DNNs) become increasingly ubiquitous in real-time embedded systems, deploying them on resource-constrained edge devices remains a critical bottleneck. Heterogeneous edge platforms—comprising CPUs, GPUs, DSPs, and dedicated Neural Processing Units (NPUs)—offer high theoretical computational density but suffer from extreme programming complexity and fragmented execution. This proposal presents a framework for scalable neural architectures tailored to heterogeneous edge environments. By combining hardware-aware neural architecture search (NAS), specialized compilers, and dynamic runtime execution schedulers, the proposed framework enables optimal performance, low latency, and energy efficiency across diverse and evolving hardware backends.

2 Introduction & Motivation

Deploying state-of-the-art deep learning models to the edge is constrained by hardware diversity and strict resource budgets. Modern edge hardware is rarely homogeneous; instead, system-on-chip (SoC) architectures deploy multiple asymmetric compute engines to balance throughput and power. However, optimization workflows remain highly siloed. Developers must manually tune networks for specific target devices, leading to long deployment cycles and suboptimal resource utilization.

To address these challenges, this research proposes a co-design paradigm that unifies model architecture optimization and heterogeneous hardware scheduling. By automatically exploring the joint design space of neural network topologies and hardware compiler settings, we can systematically discover architectures that maximize operational intensity while minimizing latency and power consumption.

3 Research Questions

This dissertation proposal aims to address three fundamental research questions:

1. **RQ1: Hardware-Aware Estimation.** How can we accurately model heterogeneous execution latency and energy consumption across diverse edge hardware backends without exhaustive physical benchmarking?
2. **RQ2: Multi-Objective NAS.** What optimization algorithms can efficiently search the joint neural-compilation design space under tight hardware constraints?
3. **RQ3: Dynamic Runtime Scheduling.** How can dynamic runtime scheduling partition neural network execution graph blocks across concurrent heterogeneous accelerators to minimize end-to-end latency?

4 Related Work

Existing methodologies in hardware-aware neural architecture search (HW-NAS) primarily target single-device deployment, such as a mobile CPU or a micro-controller. Standard approaches utilize reinforcement learning or evolutionary algorithms to navigate search spaces. However, these methods do not scale well to heterogeneous systems where multiple accelerators execute model segments concurrently.

Compiler frameworks like Apache TVM and Halide have introduced automated tensor program optimization. While effective, they operate on fixed neural architectures. Our proposed method builds upon these compiler techniques but integrates them directly into the NAS loop, enabling the model's structural parameters (e.g., channel width, kernel size) to be co-optimized with compiler parameters (e.g., loop tiling, thread binding) as detailed in Goodfellow et al. [1] and Vaswani et al. [2].

5 Proposed Methodology

The proposed framework consists of three primary components:

- **Component 1: Multi-Backend Profiler.** A lightweight profiling tool that builds predictive models of layer-wise latency and energy using Gaussian Process Regression.
- **Component 2: Co-Design NAS Engine.** A differentiable search engine that optimizes both operator-level parameters (e.g., convolutional block choices) and compilation schedules.
- **Component 3: Heterogeneous Runtime Scheduler.** A runtime scheduler that partitions the optimized model graph dynamically across available CPU, GPU, and NPU cores based on current system load.

6 Preliminary Results

We implemented a proof-of-concept version of our profiler and NAS engine on an embedded SoC containing an ARM Cortex-A72 CPU and a Mali-G71 GPU. Preliminary results show that our co-design approach yields a $1.8\times$ speedup compared to standard single-device search baselines (like MobileNetV3) while maintaining equivalent classification accuracy on ImageNet. Figure 1 outlines the workflow of our proposed three-stage pipeline.

Figure 1: Overview of the proposed three-stage co-design and scheduling framework.

7 Timeline

The research tasks are structured into four main phases over the next two academic years, as summarized in Table 1.

Table 1: Proposed Research Timeline and Milestones

Phase	Research Milestone / Key Deliverables	Timeline	Status
Phase 1	Literature review, backend profiler design, and model target definition	Q3–Q4 2026	Completed
Phase 2	Differentiable search space formulation and co-design optimizer	Q1–Q2 2027	In Progress
Phase 3	Dynamic runtime partition scheduler development and evaluation	Q3–Q4 2027	Planned
Phase 4	Multi-platform edge deployment and dissertation writing	Q1–Q2 2028	Planned

8 References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[2] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 5998–6008.