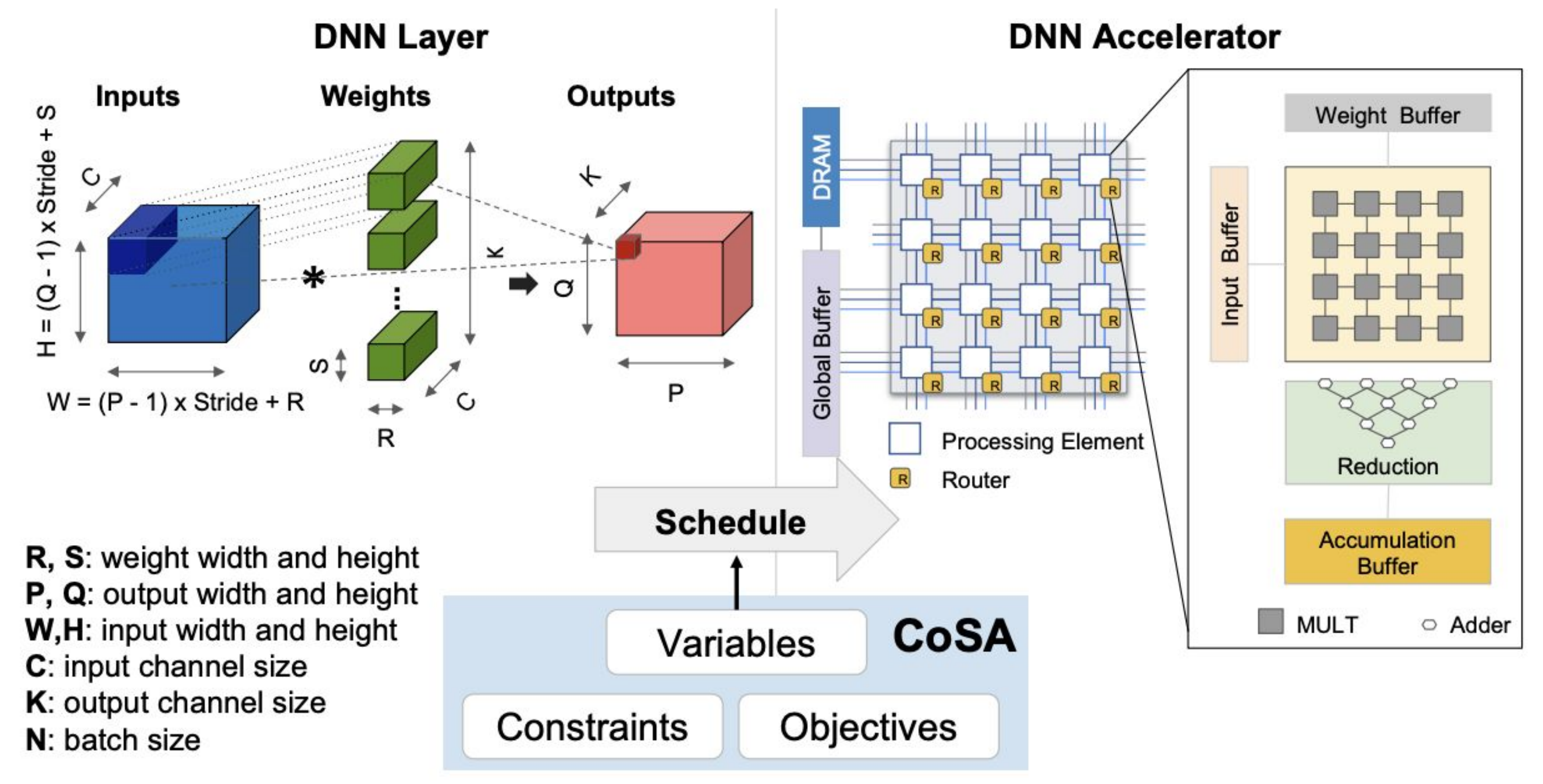
Predicting Performance of Deep Neural Network Schedules Across Accelerator Designs

Charles Hong (charleshong@berkeley.edu)   Advisor: Sophia Shao

Motivation/Prior Work

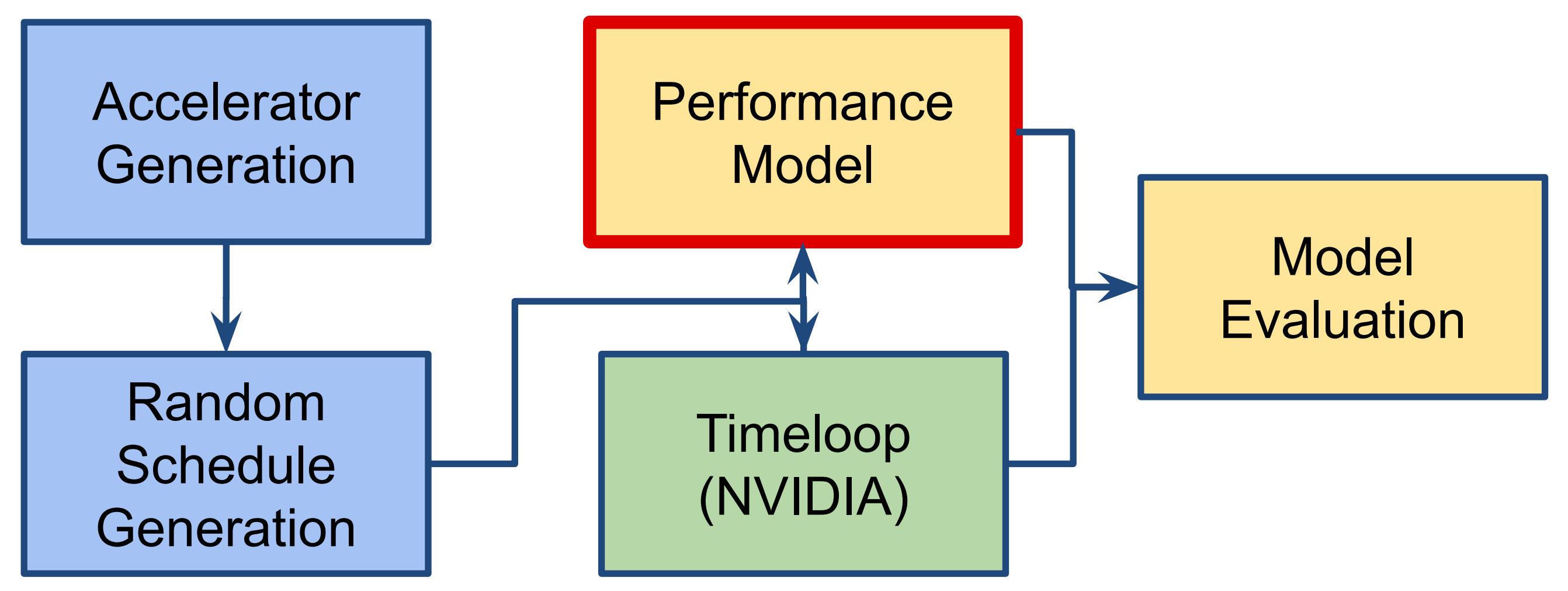


From "CoSA: Scheduling by Constrained Optimization for Spatial Accelerators"

- **The scheduling problem**
  - Tiling factors and loop permutations affect the performance of a neural network layer on a spatial accelerator
  - CoSA is a prior work that uses mixed integer linear programming (MILP) to generate a schedule, given an accelerator and CNN layer
- **HW/SW co-design**
  - Co-optimizing hardware adds an additional layer of complexity
  - HW simulation takes large amounts of time
  - MILP constraints can't be formulated like they are in the scheduling problem, because hardware can't be adjusted per layer
  - In this work, we explore data-driven, machine learning-based approaches and evaluate whether they can accurately and generalizably predict neural network accelerator performance

Hardware-Aware Performance Model

- We generate 2000+ accelerators with varying arithmetic and buffer resource amounts
- For each accelerator, we evaluate ten random schedules for each layer of ResNet-50
  - NVIDIA's open source Timeloop simulator is used as a reference
- Performance models predict layer runtime, given accelerator config and layer schedule



Performance Model

1. Linear Regression

- Serves as a baseline
- Interpretable: small number of weights can be manually inspected

2. Random Forest

- In this work: 1000 trees, max depth 10
- Medium interpretability
  - Can estimate feature importance through Mean Decrease Impurity method

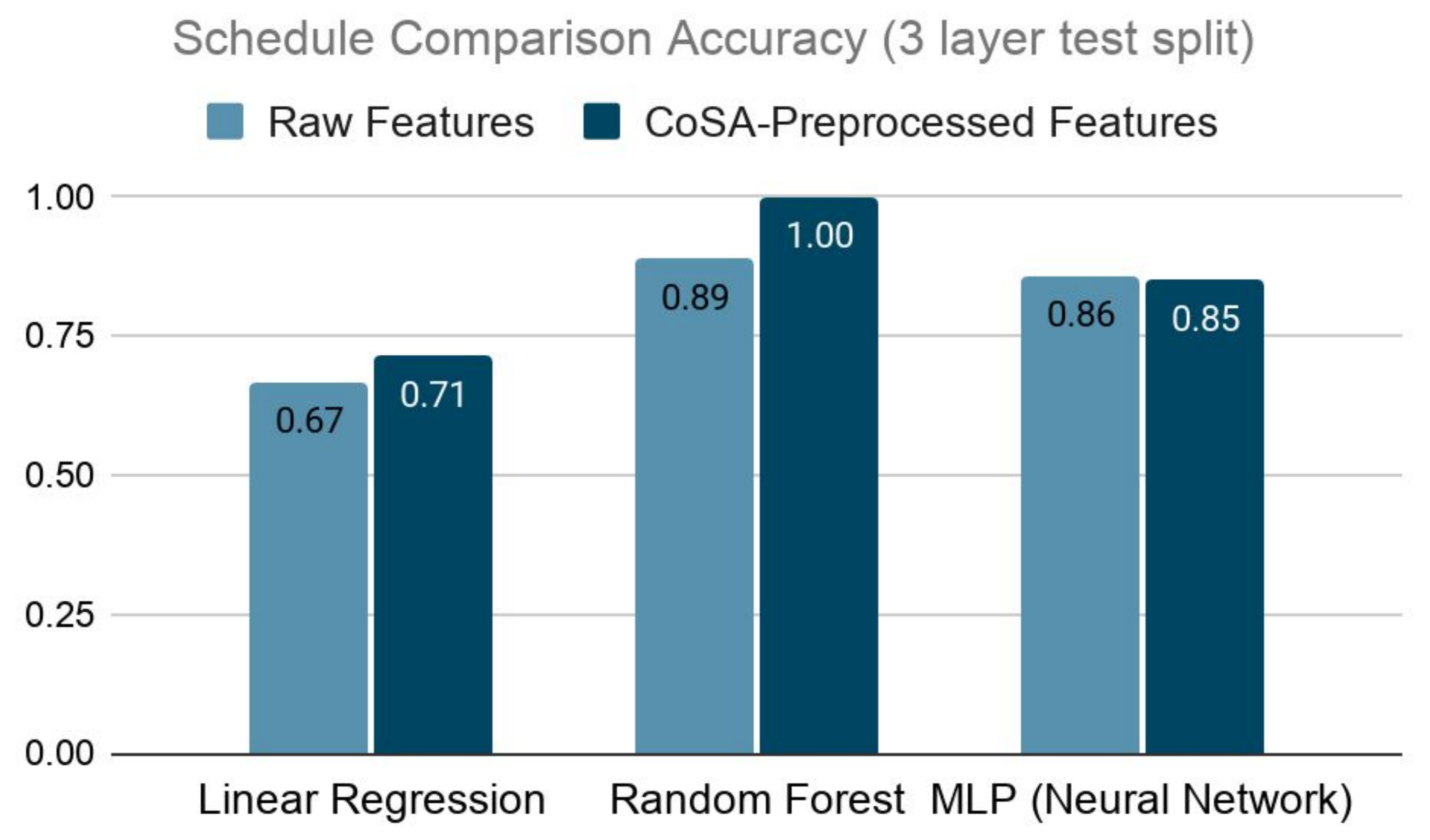
3. Multi-Layer Perceptron (neural network)

- Four hidden layers with sizes 64, 32, 16, 4
- Differentiable: may help if used to search HW design space

Model type	Interpretability	Model Capacity
Linear regression	High	Low
Random forest	Medium	High
Multi-layer perceptron	Low	High

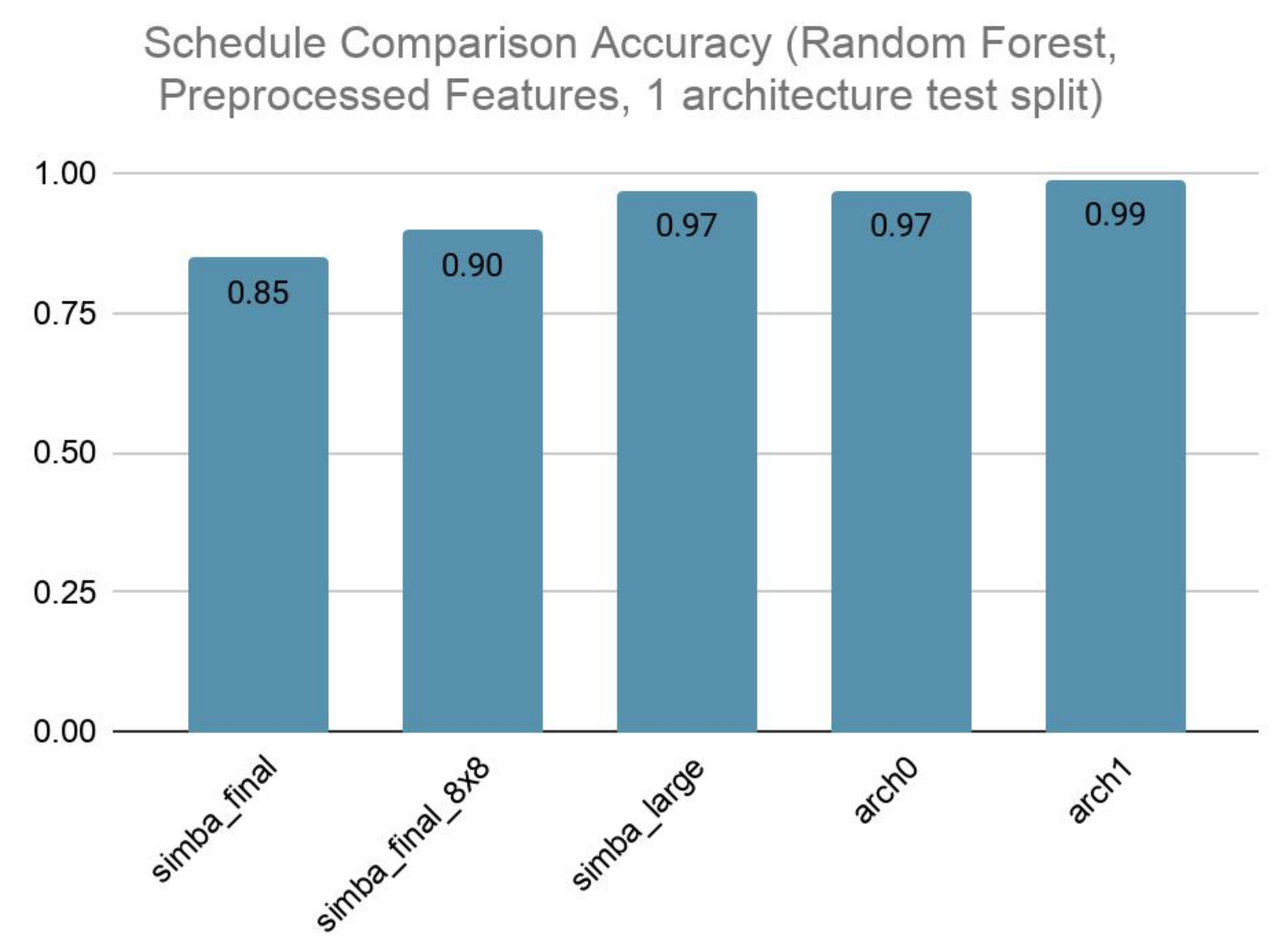
Schedule Runtime Prediction

- Models are evaluated based on their ability to predict the more performant option between two NN schedules for a new, unseen layer on previously seen hardware
- Among the three basic model types, random forests and multi-layer perceptrons perform similarly well
- The models, in particular our random forest model, perform better when augmented with an analytically preprocessed feature set that compresses the data and includes estimates for buffer traffic and resource utilization
- Schedule feature preprocessing improves prediction accuracy by 12% and reduces training time to ~0.5x, as each feature encodes more information



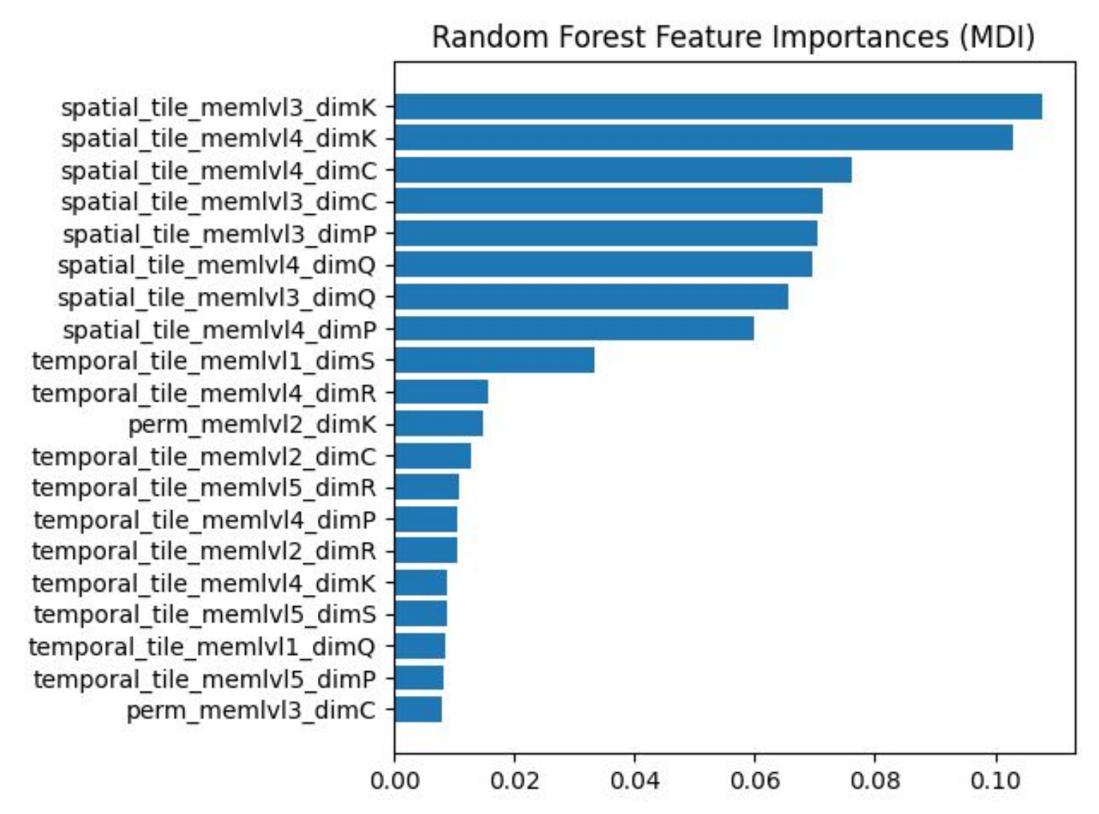
Hardware Performance Prediction

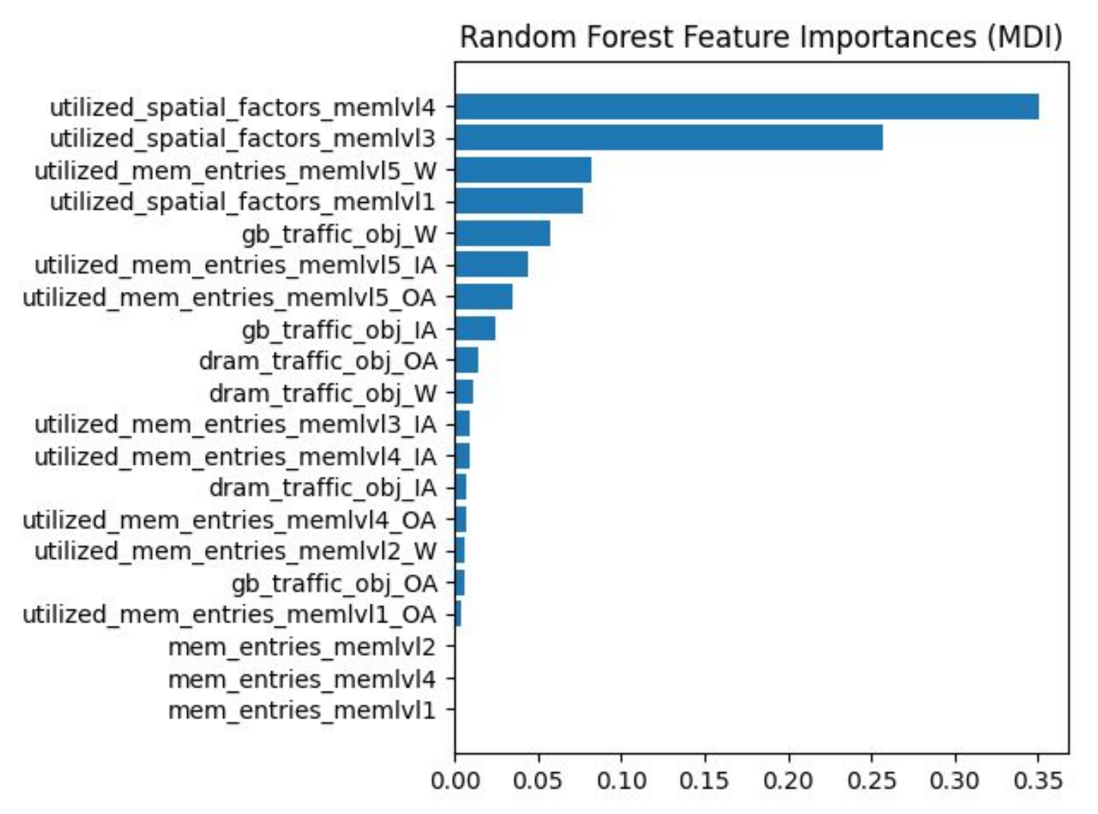
- In preliminary experiments, the random forest model generalizes surprisingly well to 1 unseen architectures when trained on 4 others



Random Forest Feature Importance

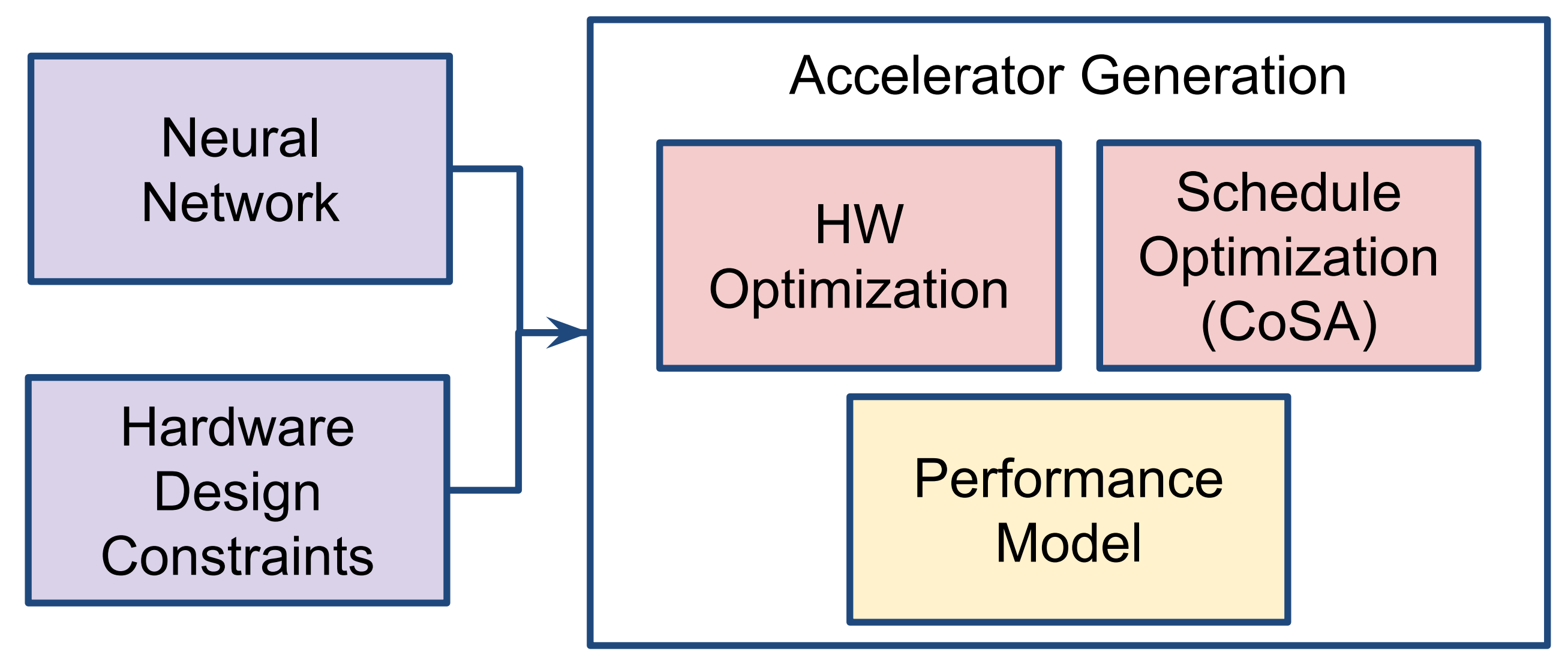
- Random forest feature importance provides insight into which features are most predictive
- Schedules implicitly encode information about hardware constraints, so hardware information appears underutilized
- MDI analysis of simba_final architecture provides evidence that input buffer and global buffer size (levels 3 and 4 below) may be bottlenecks





Future Work

- Trial different hardware optimization methods
- Improve NN performance, to take advantage of its differentiability
- Optimize for energy or EDP rather than just cycle count


Neural Network
Hardware Design Constraints
Accelerator Generation
HW Optimization
Schedule Optimization (CoSA)
Performance Model