

Saad Abbasi<sup>a</sup> (sabbasi@uwaterloo.ca), Alexander Wong<sup>a,b,c</sup>, Mohammad Javad Shafiee<sup>a,b,c</sup>

<sup>a</sup>Department of Systems Design Engineering, University of Waterloo, Canada <sup>b</sup>Waterloo Artificial Intelligence Institute, Canada, <sup>c</sup>DarwinAI Corp., Canada

## Introduction

### Background

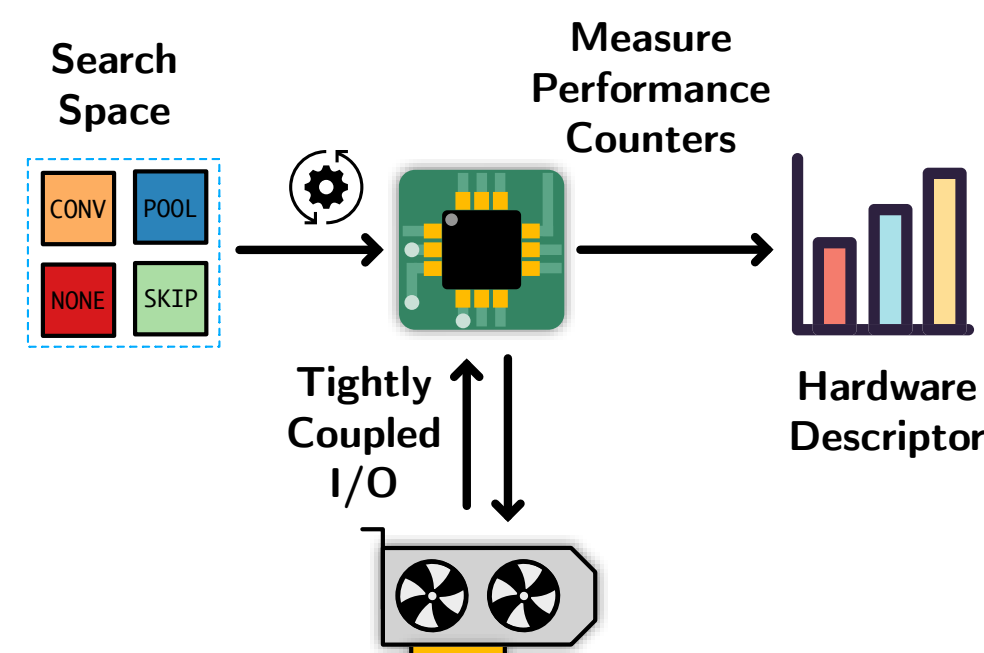
❑ **Neural Architecture Search (NAS)** is the process of automatically discovering optimal Deep Neural Network (DNN) architectures, alleviating much of the manual design process.

❑ NAS can use metrics such as latency or energy consumption to find optimal arch. However, **DNN latency is a function of its architecture and the underlying hardware [1].**

❑ Present techniques include using meta-learning to adapt latency regressors to new device. However, these techniques still require 10 samples from new hardware [2].

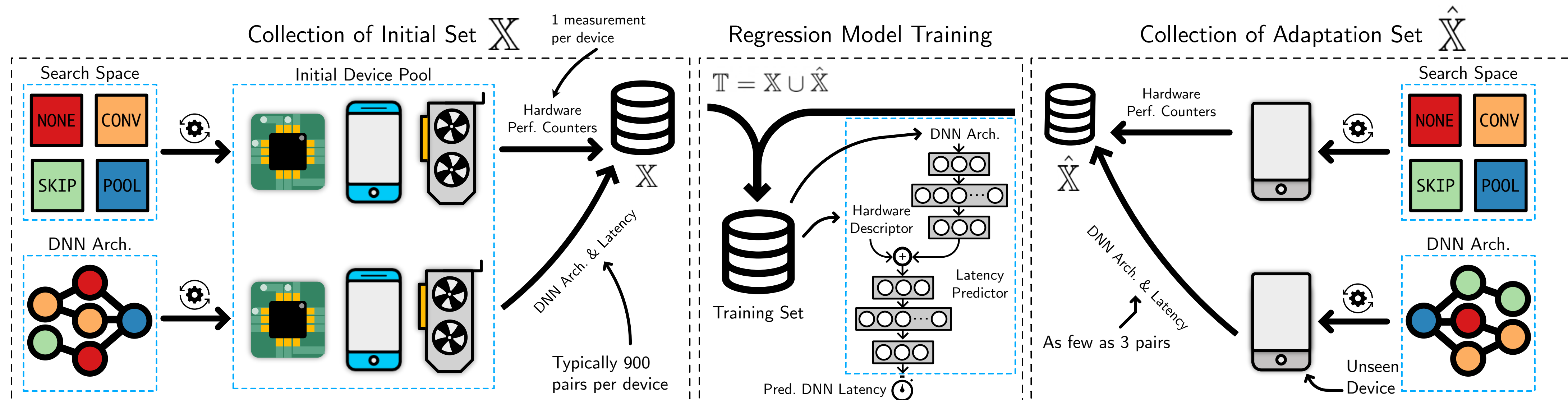
### MAPLE

❑ **MAPLE explicitly models the CPU through hardware performance counters.** Counters such as # cpu-cycles, # instructions & cache accesses/misses yield an informative hardware descriptor [3].



❑ The **hardware descriptor characterizes the search space on the target hardware** and enables the adaptation of the latency regressor with **as few as 3 samples**. This approach **works for GPUs** as well by taking leveraging the tightly-coupled between CPU and GPU.

## Methods

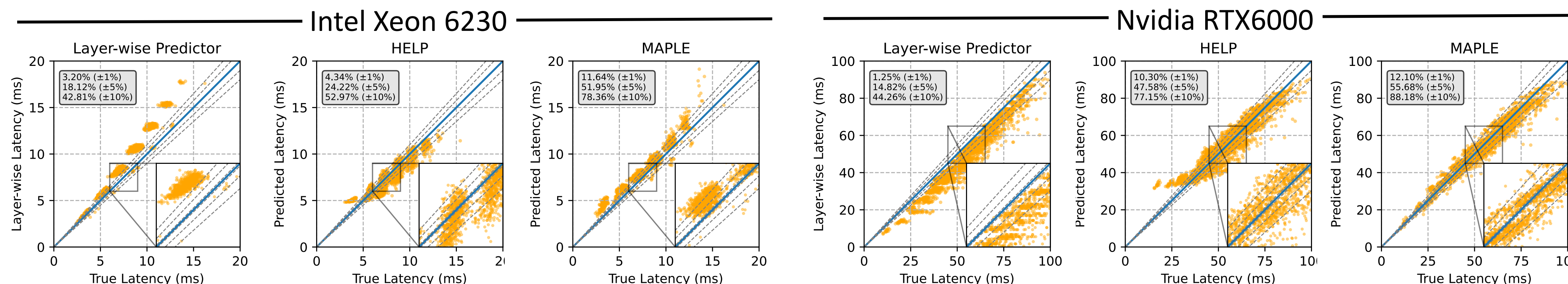


MAPLE Overview: (1) Collect latency and hardware descriptors from a set of devices. This forms the Initial Set. (2) Collecting 3 architecture latencies and hardware descriptors from the target hardware. (3) Mix in the adaptation samples with the initial set and train a regression model.

## Results

Method	# Samples	Unseen CPU			Unseen GPU			Mean
		i5-7600k	i9-9920k	Xeon 6230	GTX1070	RTX2080	RTX6000	
HELP	10	0.93	0.93	0.77	0.95	0.75	0.53	0.81
MAPLE	3	0.85	0.94	0.88	0.90	0.75	0.78	0.85
MAPLE	10	0.99	0.96	0.92	0.99	0.95	0.83	0.94

Comparison of few-shot adaptation efficacy between HELP and MAPLE. HELP was adapted to the unseen devices by collecting 10 additional samples as suggested by the authors. The efficacy of MAPLE is demonstrated by mixing 3 as well as 10 samples. The reported metric is  $\pm 10\%$  error-bound accuracy.



Visual comparison between layer-wise predictor, HELP and MAPLE on a Xeon 6230 and RTX6000. Reported metrics are  $\pm 1\%$ ,  $\pm 5\%$  and  $\pm 10\%$  error-bound accuracy. We note that the layer-wise predictor fails to capture the trend of architecture latency. This illustrates that simply summing the operator latency fails to capture the intricacies of running a full DNN architecture. Employing HELP reduces the error significantly, with most architectures falling within  $\pm 10\%$  of the actual latency. In contrast, MAPLE is able to outperform HELP and exhibits an improvement of nearly 10%, while utilizing only three samples for model adaptation.

## Conclusions

❑ We introduced a novel **hardware descriptor based on device performance events** to characterize new hardware rapidly.

❑ Despite using as few as three samples, MAPLE performs (on average) **4% better on the test hardware. When the adaptation samples are equivalent to HELP (ten), the performance gap is 13%** - despite using a simpler adaptation scheme.

❑ We take advantage of the **tight-coupling between a GPU and CPU to predict performance on GPU with CPU-based counters**, yielding a versatile solution.

❑ Our latency predictor is able to generalize to new hardware with just 3 measurements and a simple characterization of the hardware.

## References

- [1] Xiaoliang Dai et al. "Chamnet: Towards efficient network design through platform-aware model adapta-386tion". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, 387pp. 11398–11407.
- [2] Hayeon Lee et al. "HELP: Hardware-Adaptive Efficient Latency Predictor for NAS via Meta-Learning" (2021)
- [3] Wiplove Mathur and Jeanine Cook. "Toward accurate performance evaluation using hardware counters". 345In: ITEA Modeling and Simulation Workshop. 2003, pp. 23–32