UNIVERSITY OF CAMBRIDGE

Ansor: Generating High-Performance Tensor Programs for Deep Learning

R244: Large-Scale Data Processing and Optimisation

Felix Jonathan Rocke

Lianmin Zheng, Chengfan Jia, Minmin Sun, Zhao Wu, Cody Hao Yu, Ameer Haj-Ali, Yida Wang, Jun Yang, Danyang Zhuo, Koushik Sen, Joseph E. Gonzalez, and Ion Stoica

Problem Statement

Improving the performance of deep learning models requires Hardware-specific optimisations

Automatic Code-Generation (ML Compiler, e.g. AutoTVM)

+ Less to no engineering effort necessary to adapt to different/new hardware

- Performance is not always as good as with manual optimisations

Manual Optimisation (Operator Library, e.g. CuDNN)

- Significant engineering effort necessary to adapt to different/new hardware

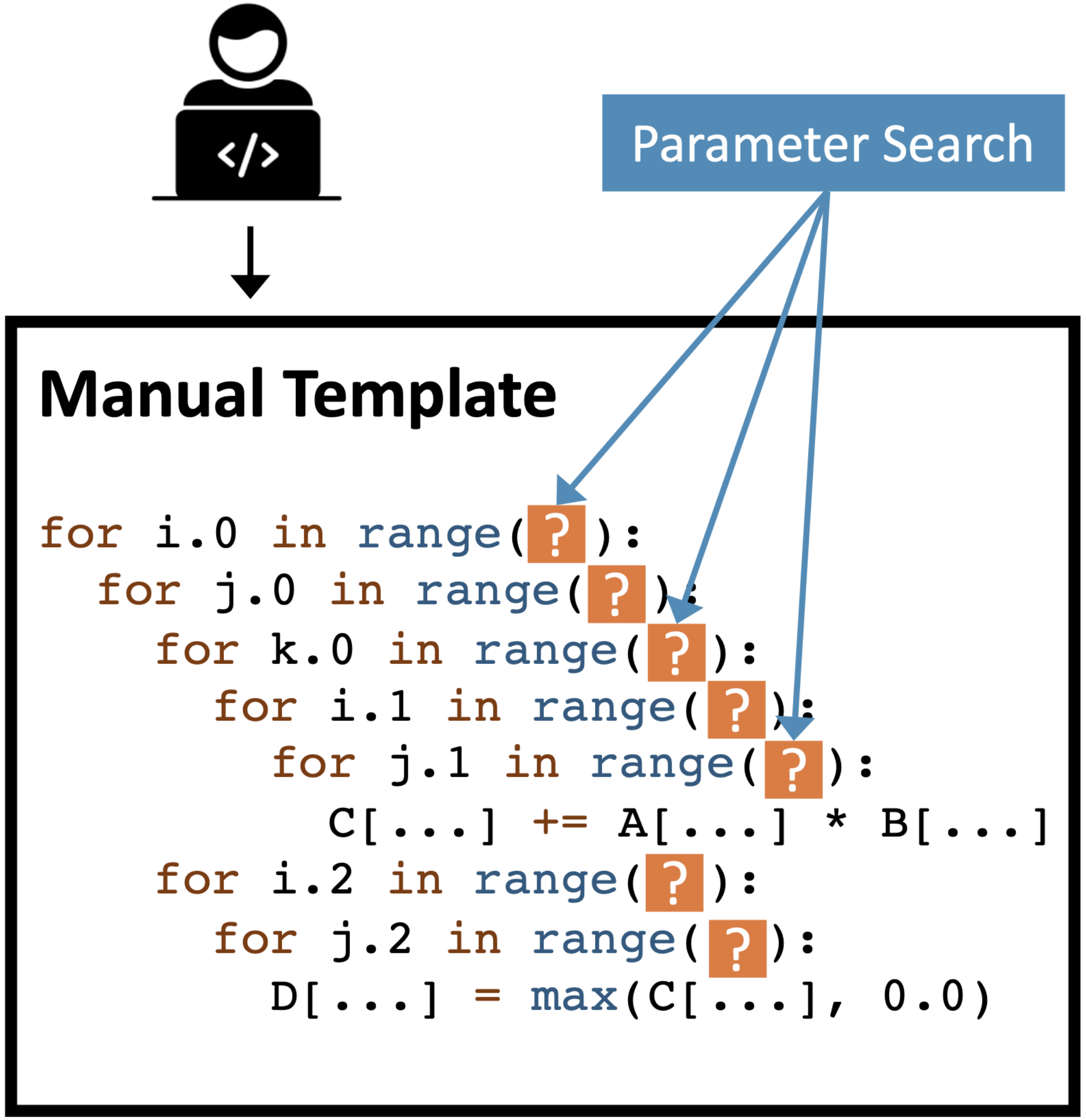
+ Performance is often better than automatic generation

Template Guided Search

1. Experts create hardware-specific tensor code templates

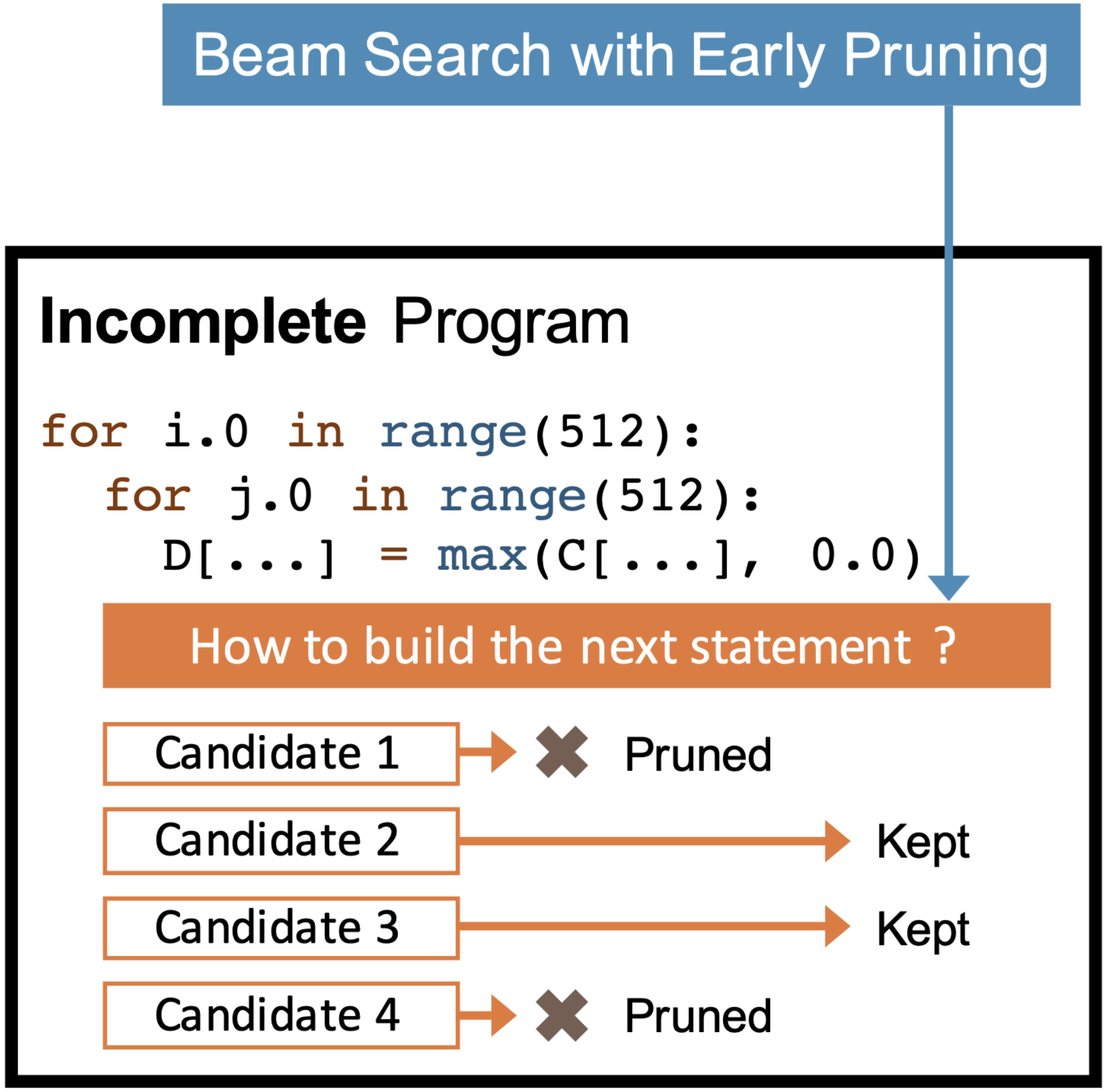
   Expert knowledge required in:
   - Hardware architecture
   - Optimisation techniques

2. Parameters are determined via an automatic search algorithm



Sequential Program Construction

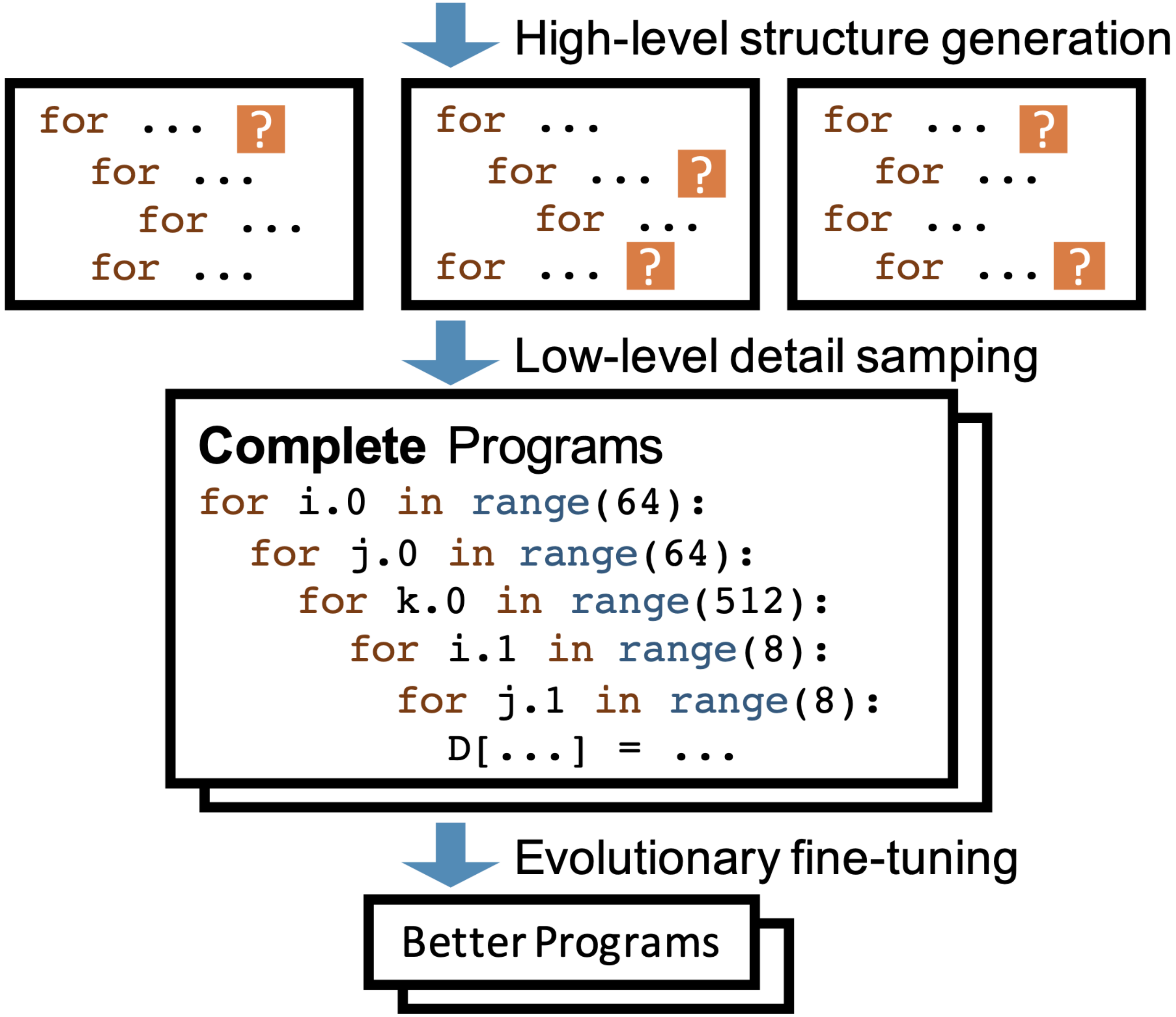
- Programs are sequentially constructed through a fixed sequence of decisions
- Uses unfolding rules for every node
- Only the top-k candidates are kept
- Cost function is used to evaluate incomplete programs
  - ➢ Low accuracy of cost function at the beginning of program creation
    - ➢ Candidate programs are pruned to early
    - ➢ Limited search space



Ansor Hierarchical Search

1. High and low-level structures are separated
2. Create search space of high-level tensor programs
3. Sample high-level programs uniformly from search space
4. Sample low-level features
5. Fine-tune low-level features

- ➢ No early pruning or limited options
  - ➢ Greater search space covered
    - ➢ Better programs



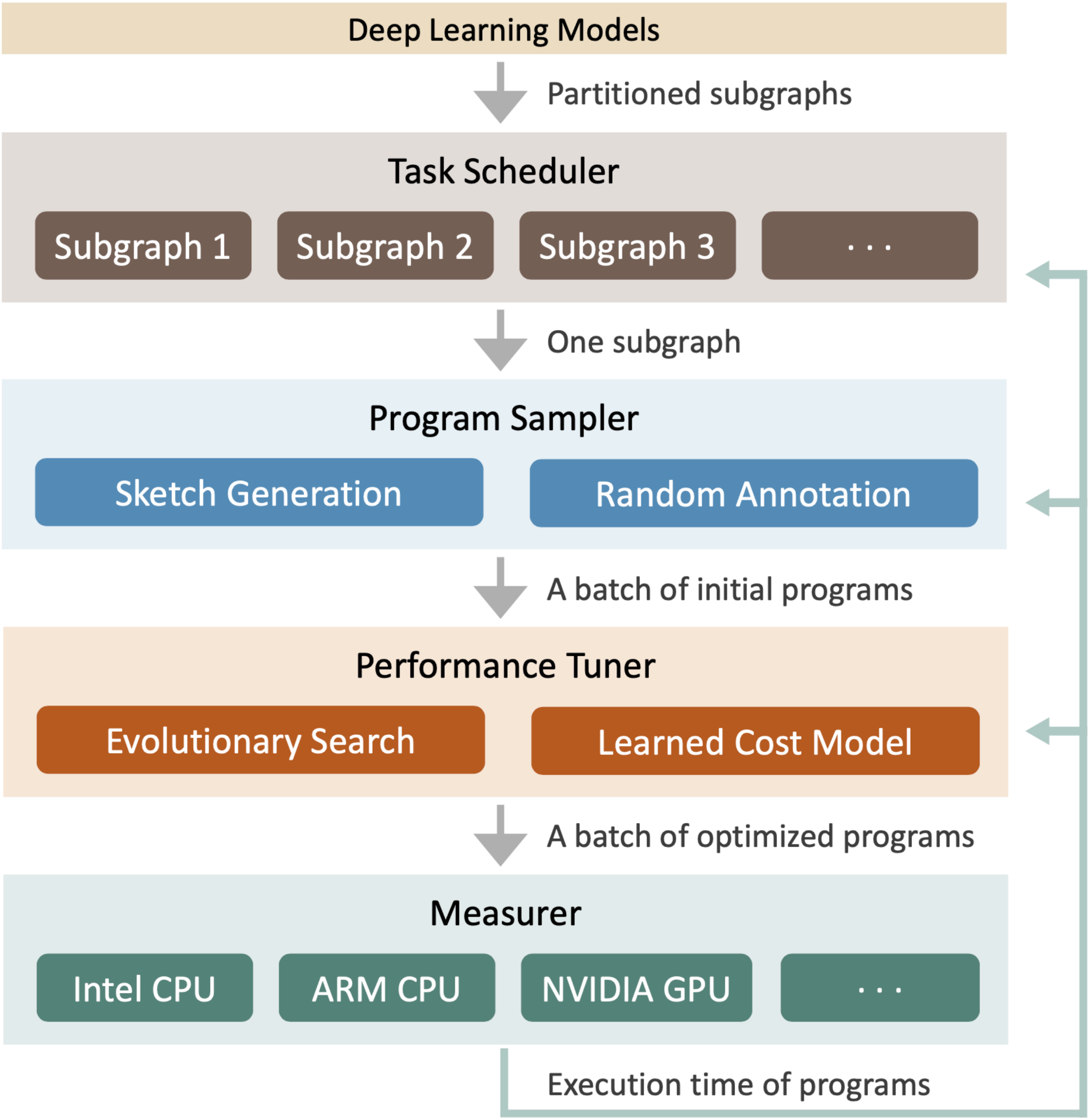
Ansor Overview

Split computational graph into subgraphs

Schedule subgraph for program sampling

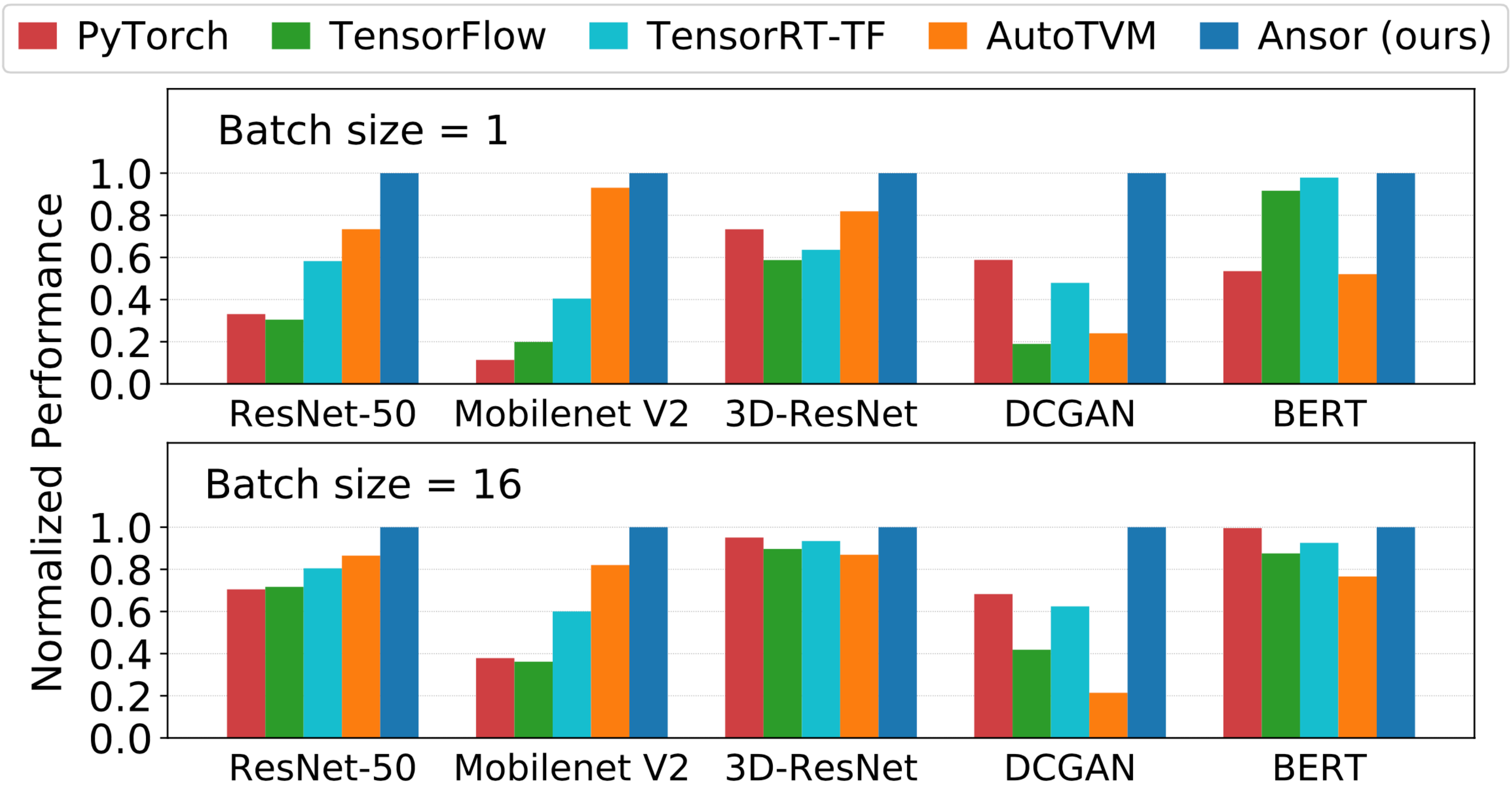
Uniformly select initial programs

Send programs to hardware for measurements



Provide feedback to the individual steps, e.g. updates to the learned cost model

GPU Benchmark Results



Results for Nvidia V100

Pros & Cons / Discussion

Cons:

- An ARMv8 CPU is used in the benchmarks
    - However, it is not specified if NEON is enabled, unlike for the x86 CPU, where it is stated that AVX-512 was used
- No multi-objective optimisations

Pros:

- Simplifies model optimisation since manual kernel or template development is replaced
- Better performance than hand-optimisation