Towards a Benchmarking Suite for Kernel Tuners

Jacob O. Tørring, Ben van Werkhoven, Filip Petrovič,
Floris-Jan Willemsen, Jiří Filipovič, Anne C. Elster
Norwegian University of Science and Technology (NTNU), Norway
Netherlands eScience Center, The Netherlands
Masaryk University, Czech Republic

Norwegian University of Science and Technology

Outline

- Introduction
- Related work
- BAT 2.0
- Results
- Conclusion and Future work

Introduction

- Growing complexity in HPC systems
- Challenges in code optimization for heterogeneous systems
- Autotuning for code efficiency across generations of systems
- Need for benchmarks to compare different GPU kernel tuners

Contributions

- Benchmark suite for kernel tuners
- Problem interface to easily integrate new tuners, benchmarks and consistent reporting of results.
- Analysis of autotuning benchmarks based on five metrics
- A comparative study of autotuners is left for future work

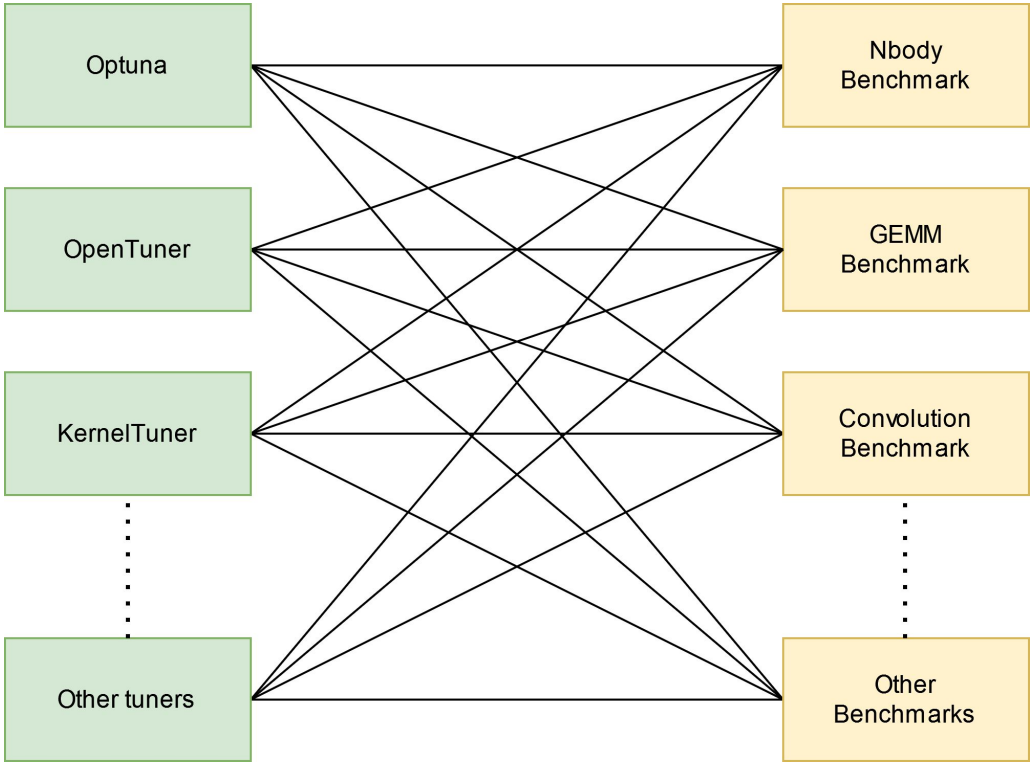
Related Work

- Rodinia, SHOC, PolyBenchGPU
- KTT Benchmark Suite
- Nevergrad, etc. black-box optimization problems.
- BAT 1.0

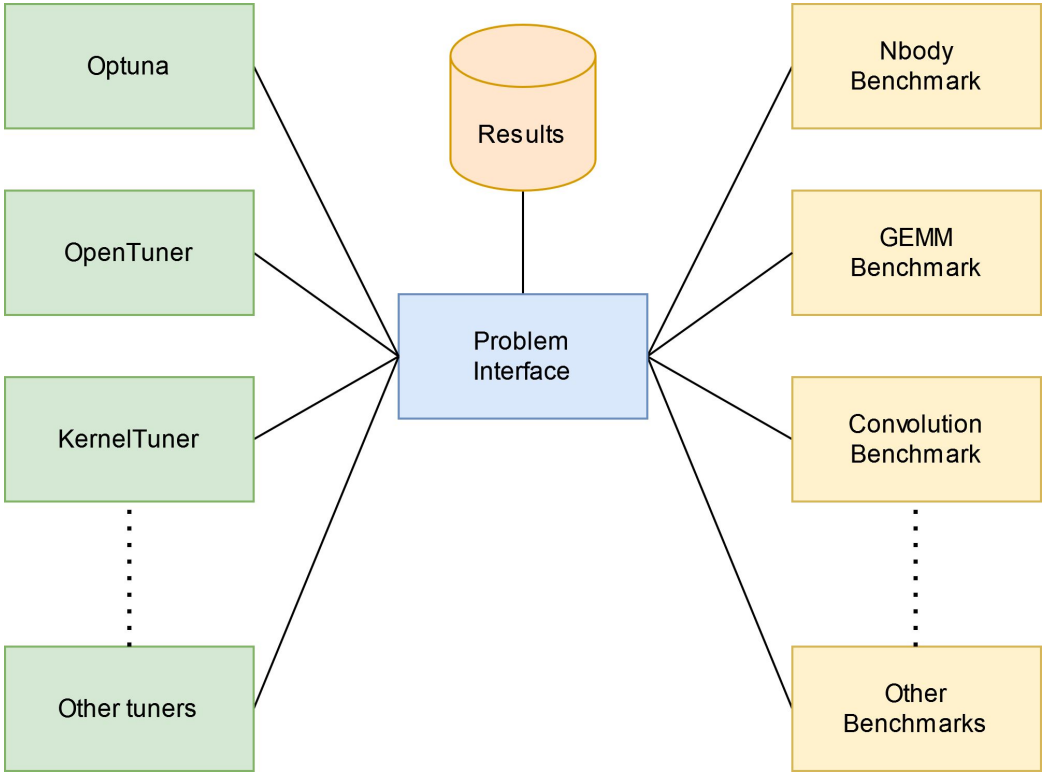
Comparison

	Large search spaces	Impactful parameters	Built for Multiple tuners	Consistent interface
PolyGPU	No	Yes?	N/A	N/A
KTT	Yes	Yes?	No	N/A
BAT 1.0	Yes	No	Yes	No
BAT 2.0	Yes	Yes	Yes	Yes

Benchmark suite structure 1.0

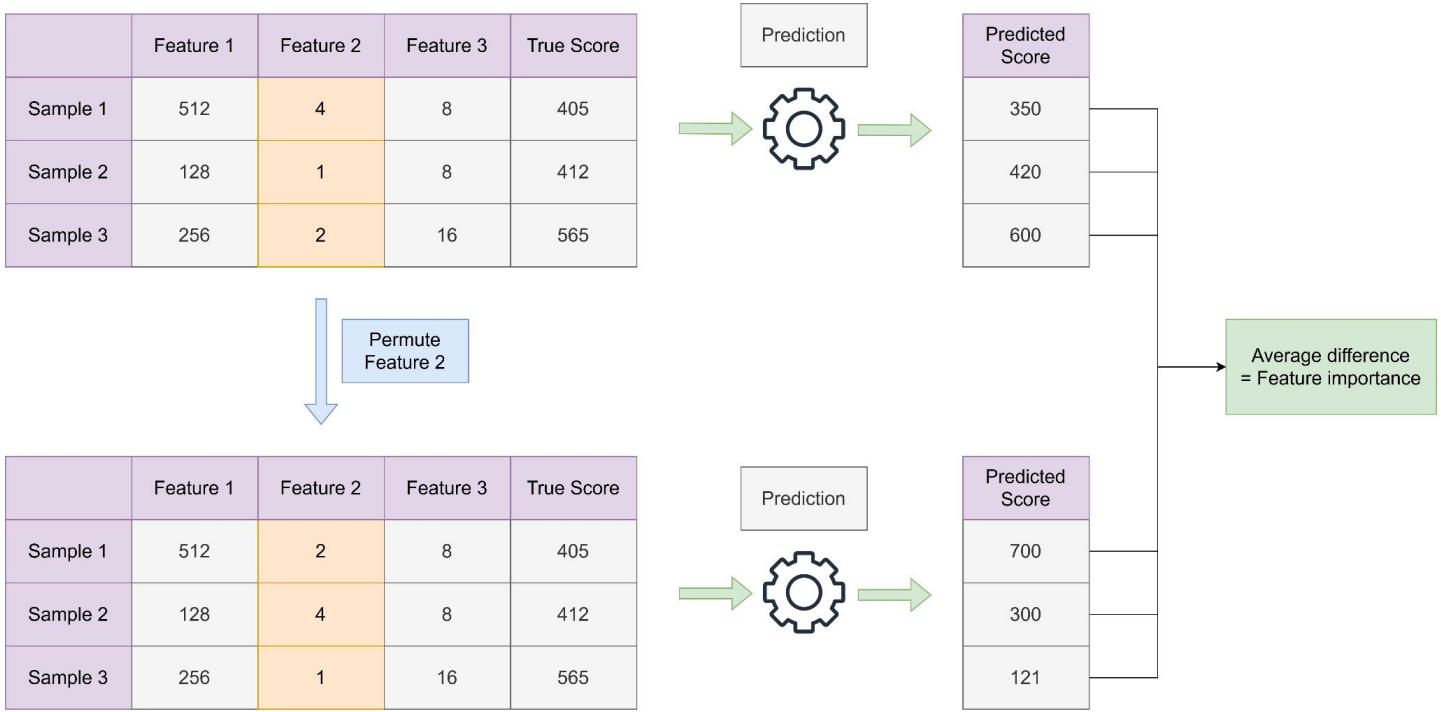


Benchmark suite structure 2.0



Metrics

Permutation Feature Importance

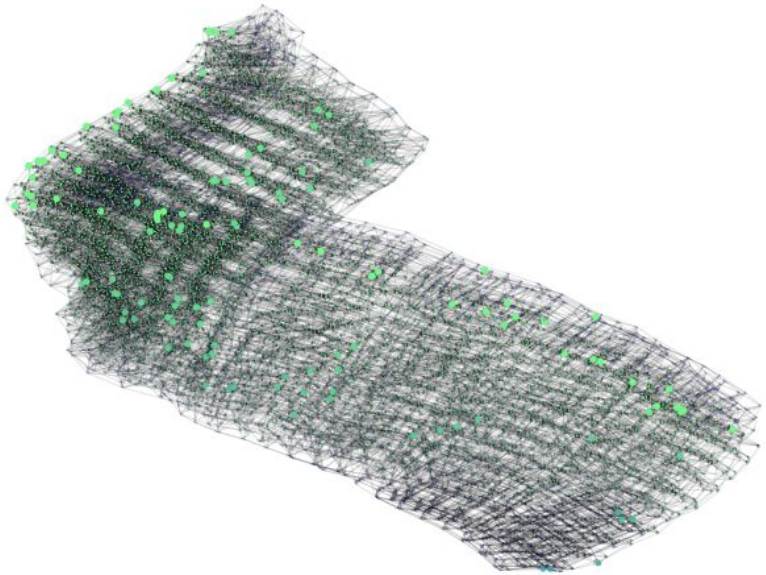


Feature importance

- Important for developing algorithms that can navigate the space of more and less important features.
- Understand the performance of search algorithms.
- Constrain the size of the search space

Proportion of Centrality

1. Fitness Flow Graphs
   a. Directed edge to better fitness
2. Reachability of each node
3. Local minima



R. Schoonhoven, B. van Werkhoven and K. J. Batenburg, "Benchmarking optimization algorithms for auto-tuning GPU kernels," in *IEEE Transactions on Evolutionary Computation*, 2022, doi: 10.1109/TEVC.2022.3210654.

BAT 2.0 Benchmark Suite

- Representative GPU kernels from real-world applications
- Common problem interface for easy integration
- Evaluation based on five characteristics:
  - Convergence rate,
  - Local minima centrality,
  - Optimal speedup,
  - Permutation Feature Importance (PFI)
  - Performance portability

Benchmarks

- GEMM
  - Generalized dense matrix-matrix multiplication, from CLBlast
- N-body
  - Gravitational forces in astrophysics, from KTT benchmark suite.
- Hotspot
  - Thermal simulation benchmark from Rodinia benchmark suite.
- PnPoly
  - Point-in-poly, querying point clouds in geospatial databases
- Convolution
  - 2D convolution operation, eScience Center
- Expdist
  - Localization microscopy, eScience Center
- Dedispersion
  - Detection of single pulse astronomical transients, eScience Center

Benchmark	Cardinality
PnPoly	4 092
Nbody	9 408
Convolution	18 432
GEMM	82 944
Expdist	9 732 096
Hotspot	22 200 000
Dedisp	123 863 040

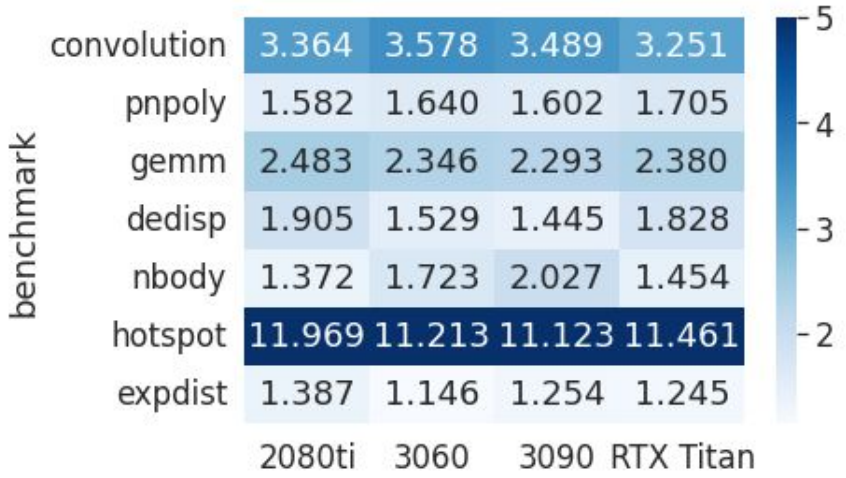
Common parameters:
Block size, Tile size, Loop unrolling, Shared memory

Experimental setup

- Hardware
  - RTX 2080Ti
  - RTX 3060
  - RTX 3090
  - RTX Titan
- Search space explored
  - Exhaustive: GEMM, PnPoly, Nbody, Convolution
  - Random search (10 000): Hotspot, Dedisp, Expdist

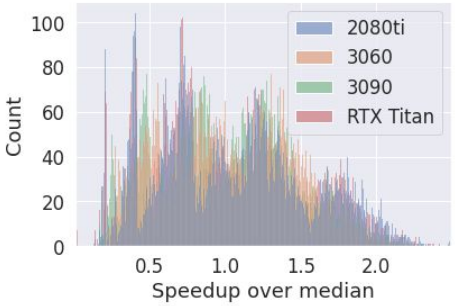
Results: Speedups over median

- Optimal speedup over median
- 1.15x - 11.97x
- Varies between benchmarks
- Generally consistent across architectures

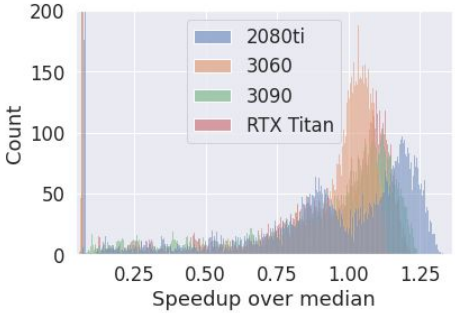


Results: Performance Distribution

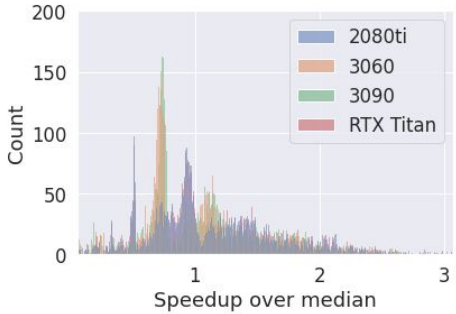
GEMM



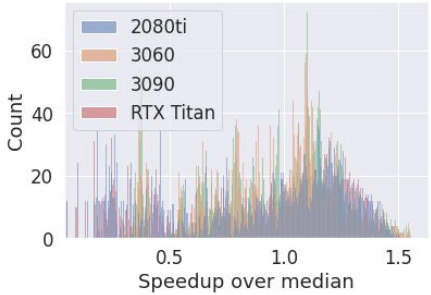
Expdist



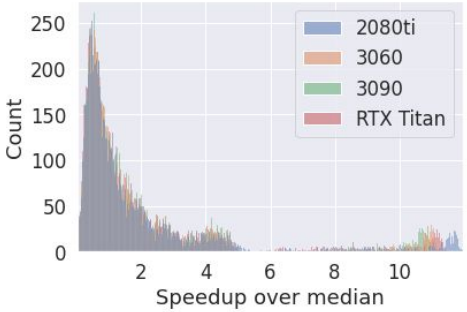
Convolution



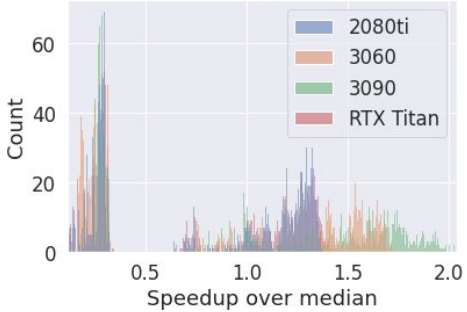
PnPoly



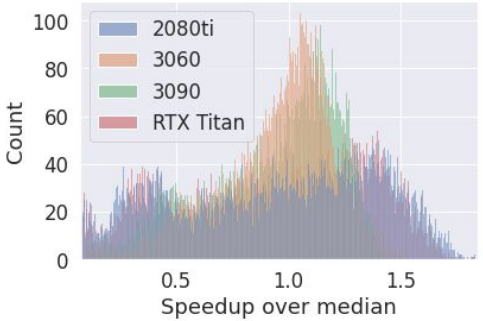
Hotspot



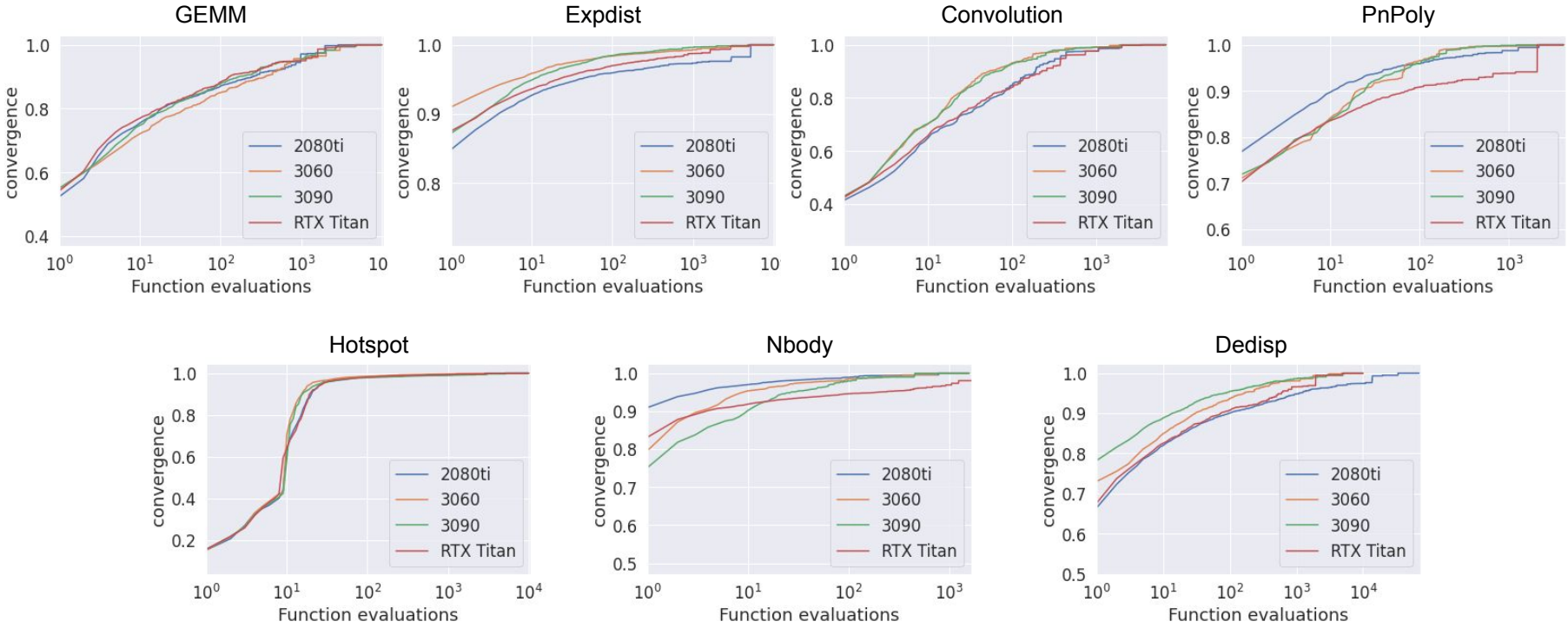
Nbody



Dedisp

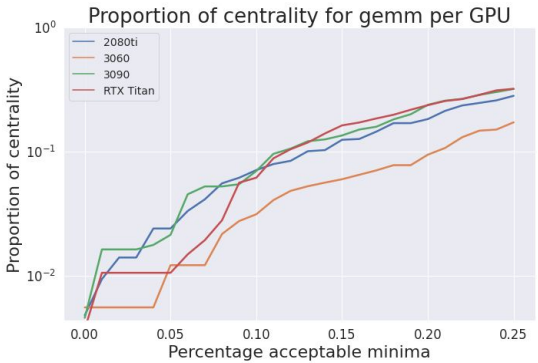


Results: Convergence

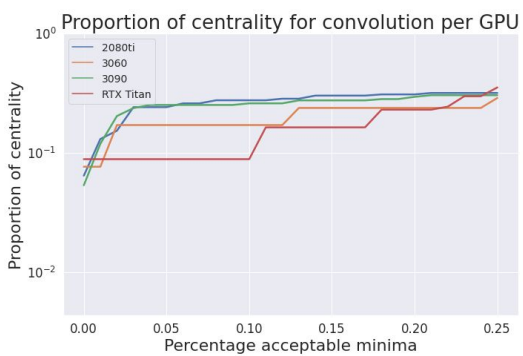


Results: Centrality

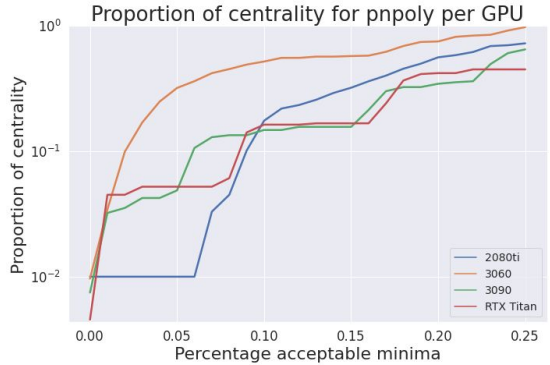
GEMM



Convolution

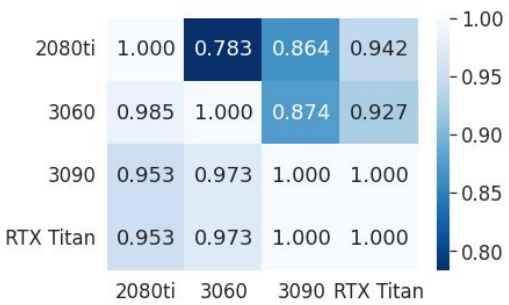


PnPoly

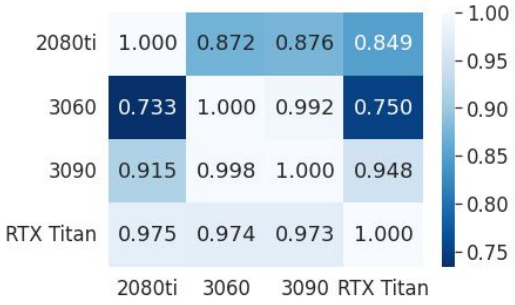


Results: Portability Performance

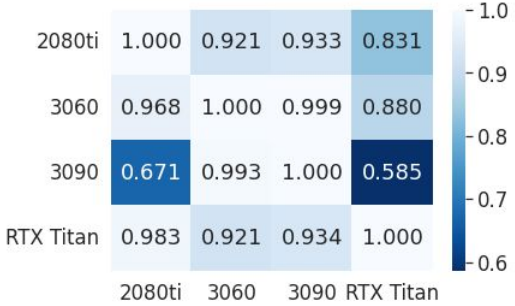
Nbody



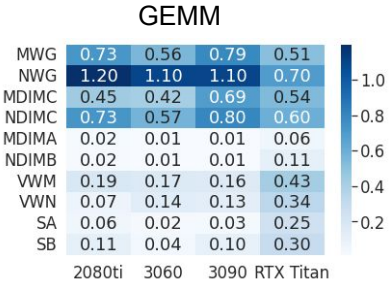
Convolution

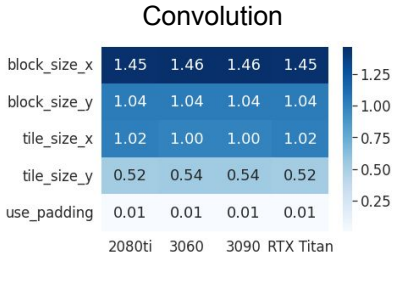


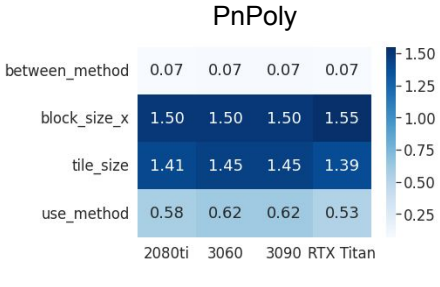
PnPoly

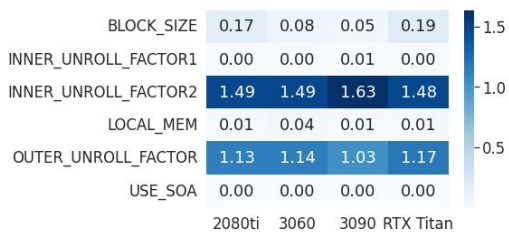


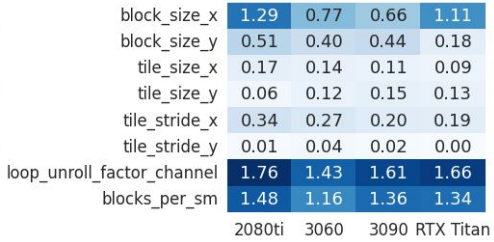
Results: Feature Importance

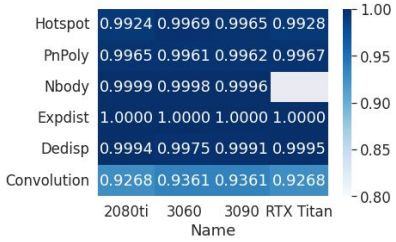












Conclusion

- GPU autotuning research needs a shared benchmark suite
  - Diverse benchmarks with large and interesting search spaces
  - Easy to implement new autotuners and benchmarks
- Analysis gives insights into autotuning characteristics of the benchmarks
- BAT 2.0 benchmark suite is useful for facilitating the study of optimization algorithms

Thank you for listening!

Contact information
Jacob O. Tørring
jacob.torring@ntnu.no