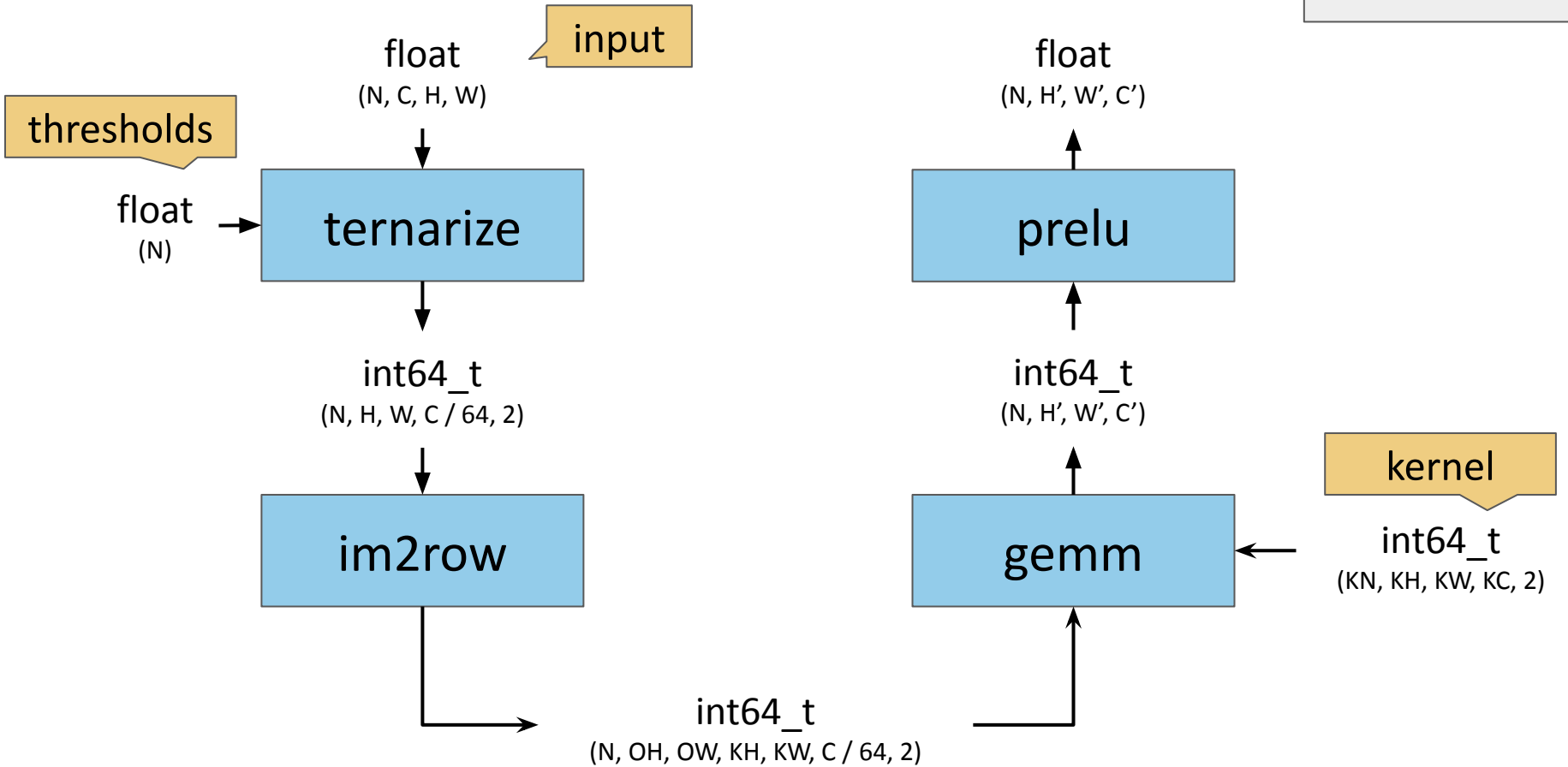
Bitwise Convolution for Ternary Neural Networks

Felix Möller, Daniel Nezamabadi, Rudy Peterson, Luca Tagliavini
Supervisor: Shien Zhu

Advanced Systems Lab - Final Presentation, ETH Zurich
June 7th, 2024

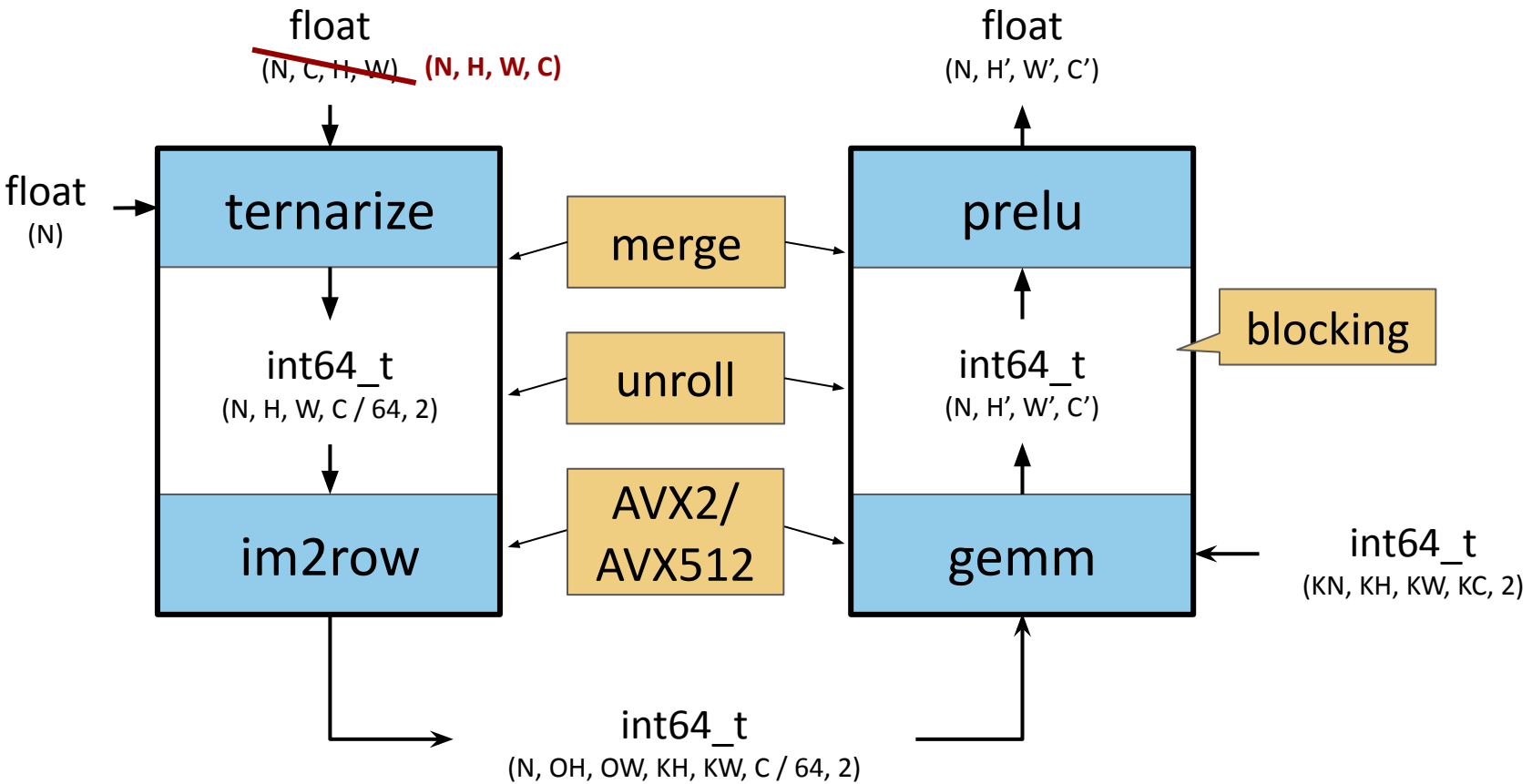
Overview: Algorithm

std::vector



Overview: Our Work

Tensor ~~std::vector~~



Using Tensors

```
class Tensor3D {
  T *data;
  const size_t dim1;
  const size_t dim2;
  const size_t dim3;
}
```

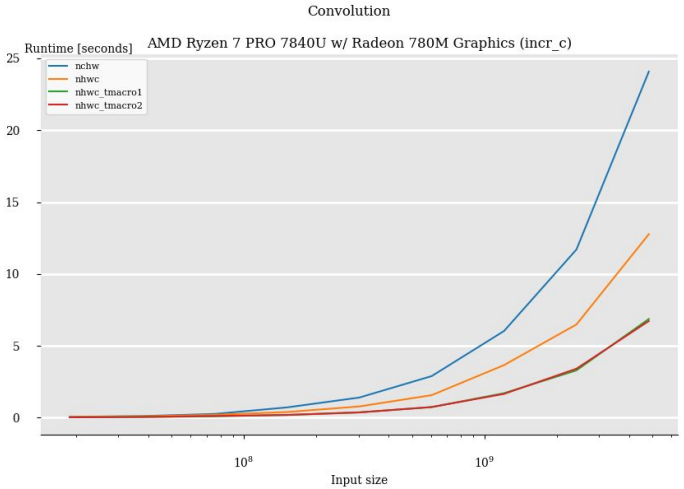
nhwc_tmacro1

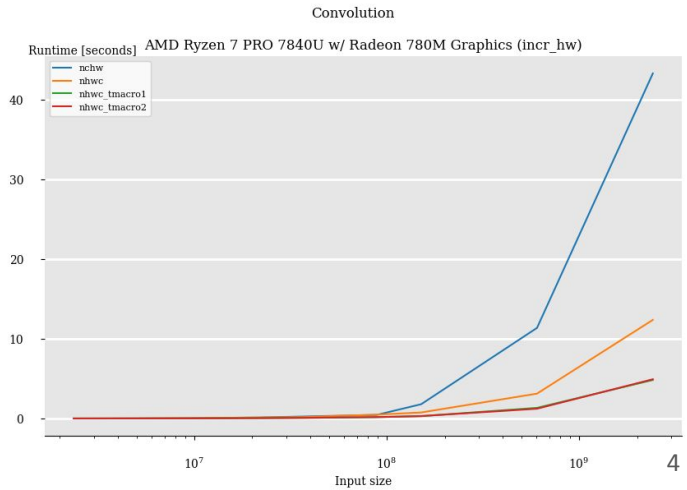
```
#define tensor3d_set(data, dim2, dim3, value, i, j, k)          \
  ((data)[((i) * ((dim2) * (dim3))) + ((j) * (dim3)) + (k)] = (value))
```

nhwc_tmacro2

```
#define tensor3d_set(data, dim2, dim3, value, i, j, k)          \
  ((data)[((i) * (dim2) + (j)) * (dim3) + (k)] = (value))
```

gcc with O3 and disabled vectorization



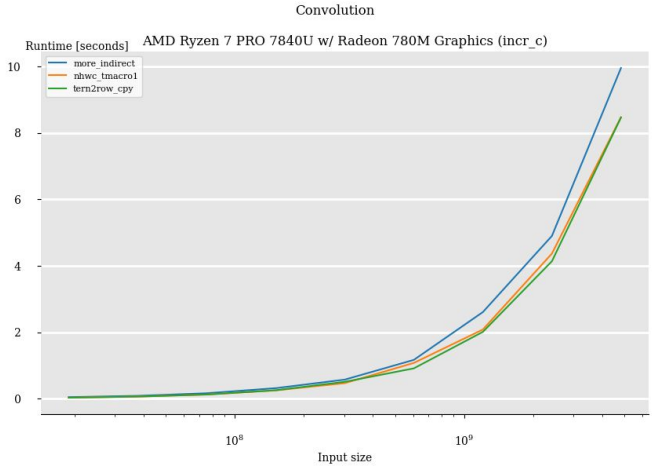


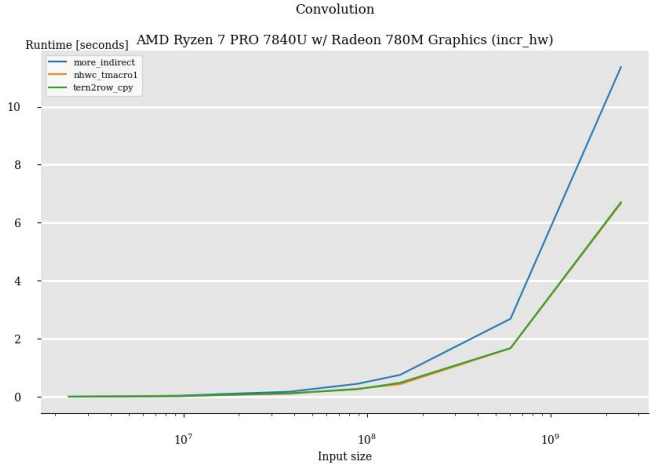
Merging im2row

Don’t merge im2row

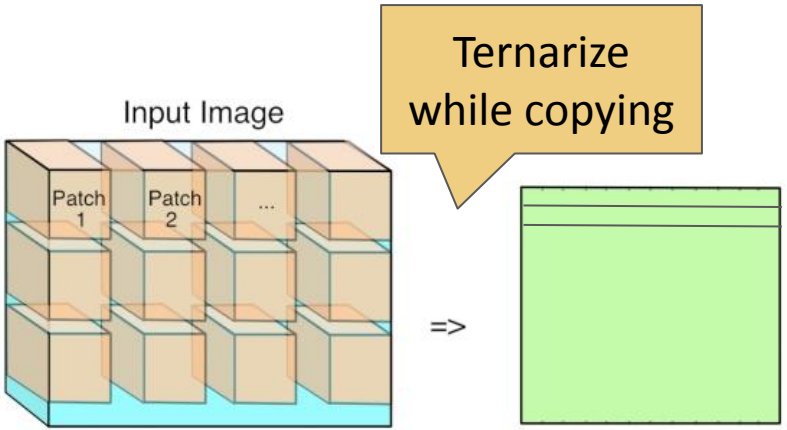
Merge ternarize and im2row

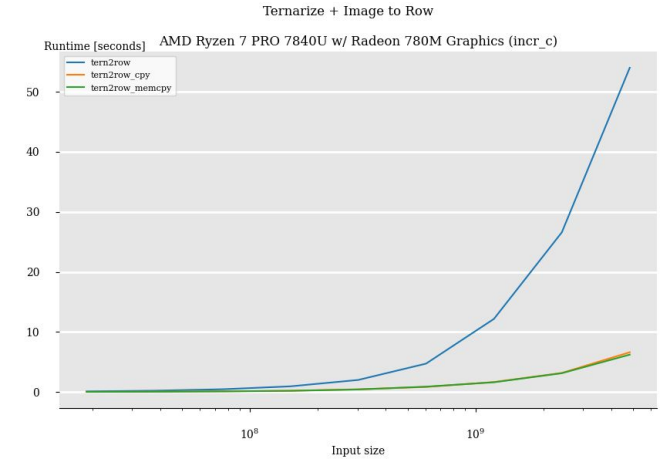
Merge im2row and gemm (indirect)





ternarize + im2row





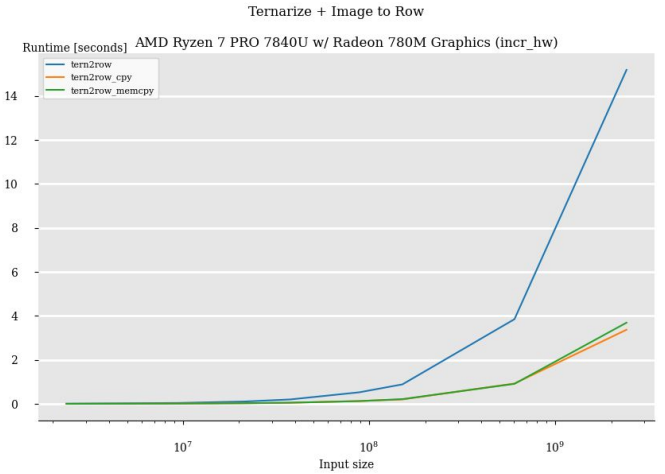
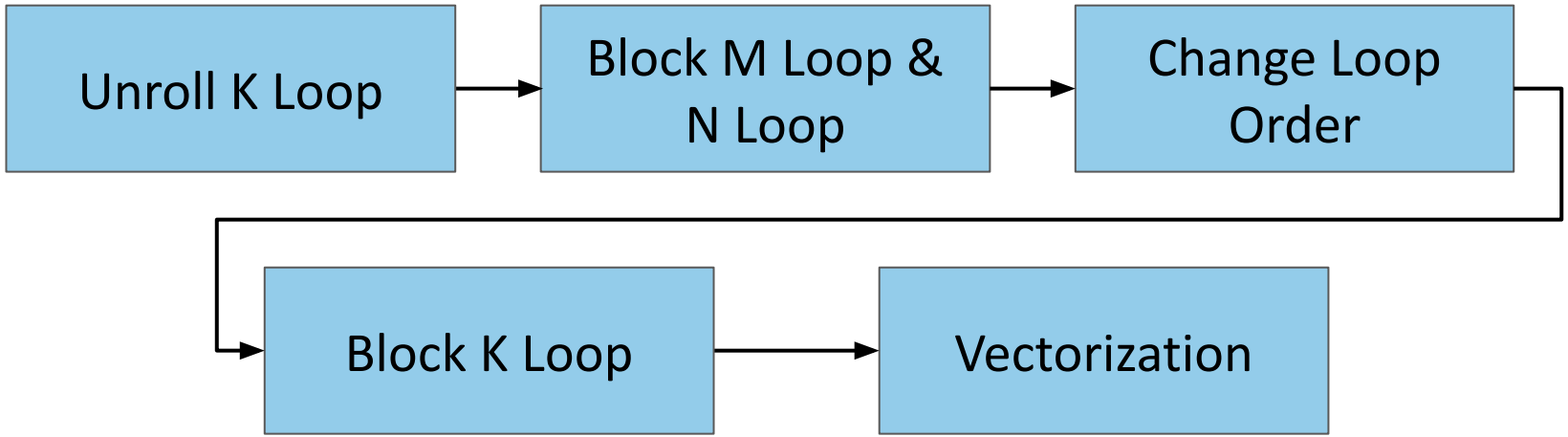
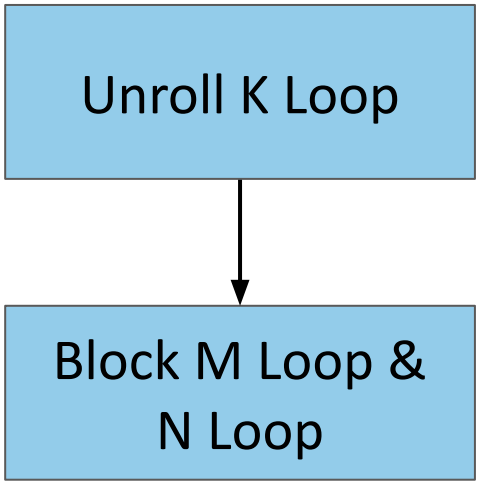


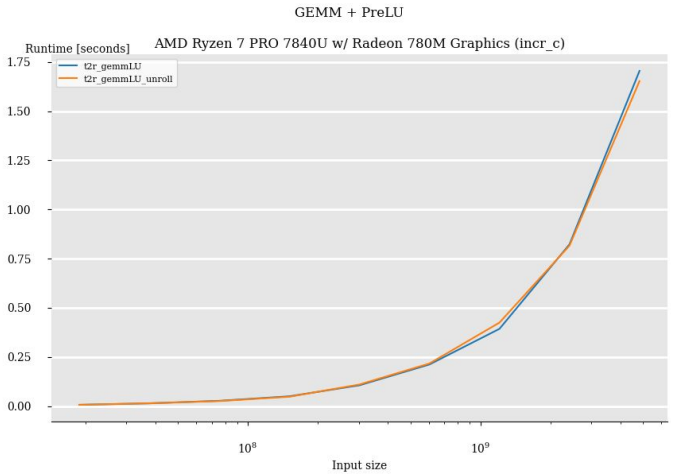
Image adapted from https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/convolution_layer/making_faster

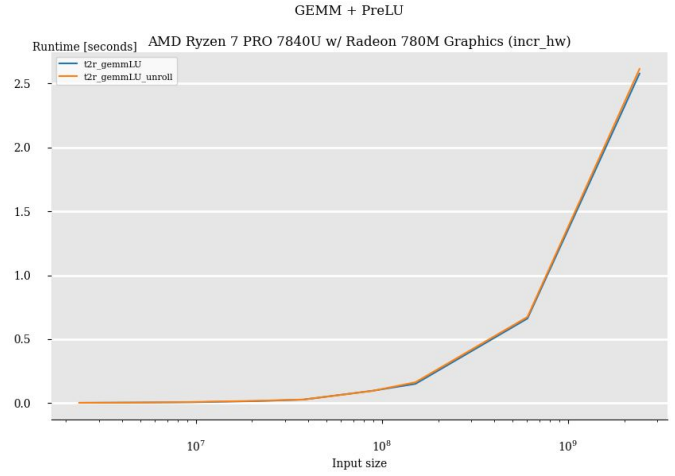
GEMM + PReLU - Optimization Plan



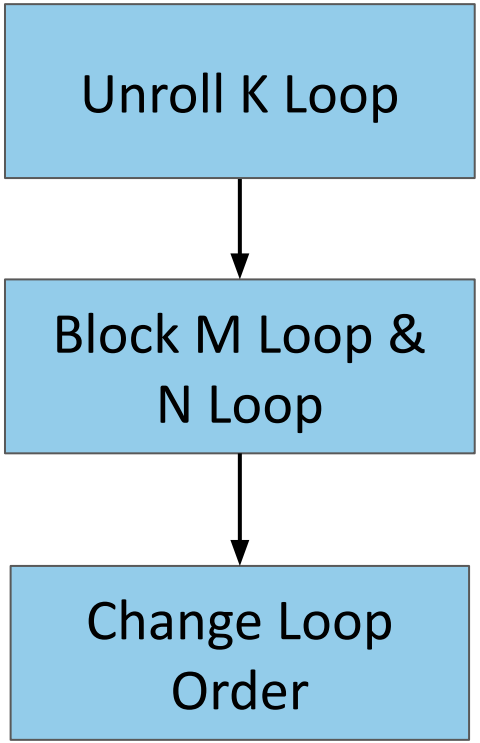
GEMM + PReLU - Unrolling over K & Blocking over M and N

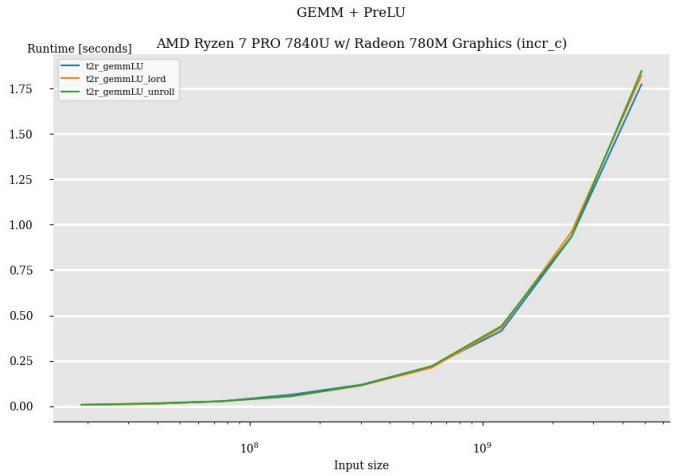


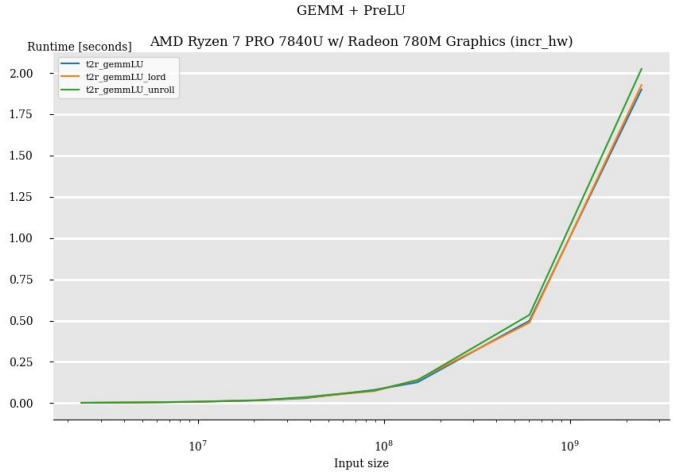




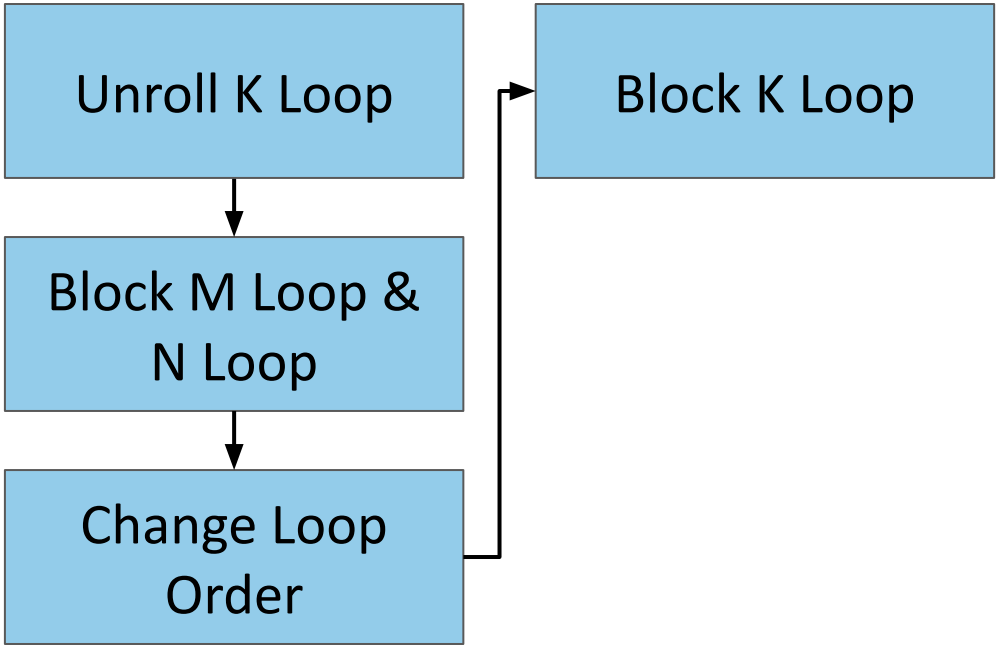
GEMM + PReLU - Changing the Loop Order

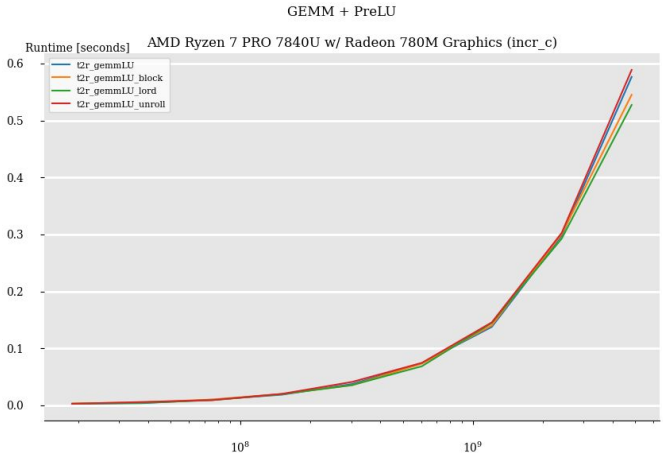


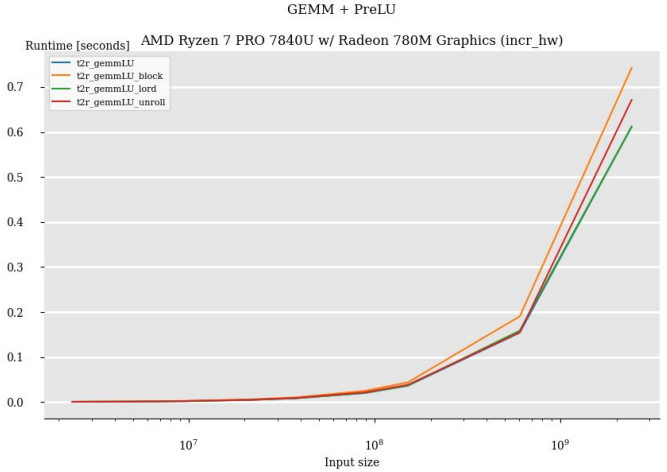




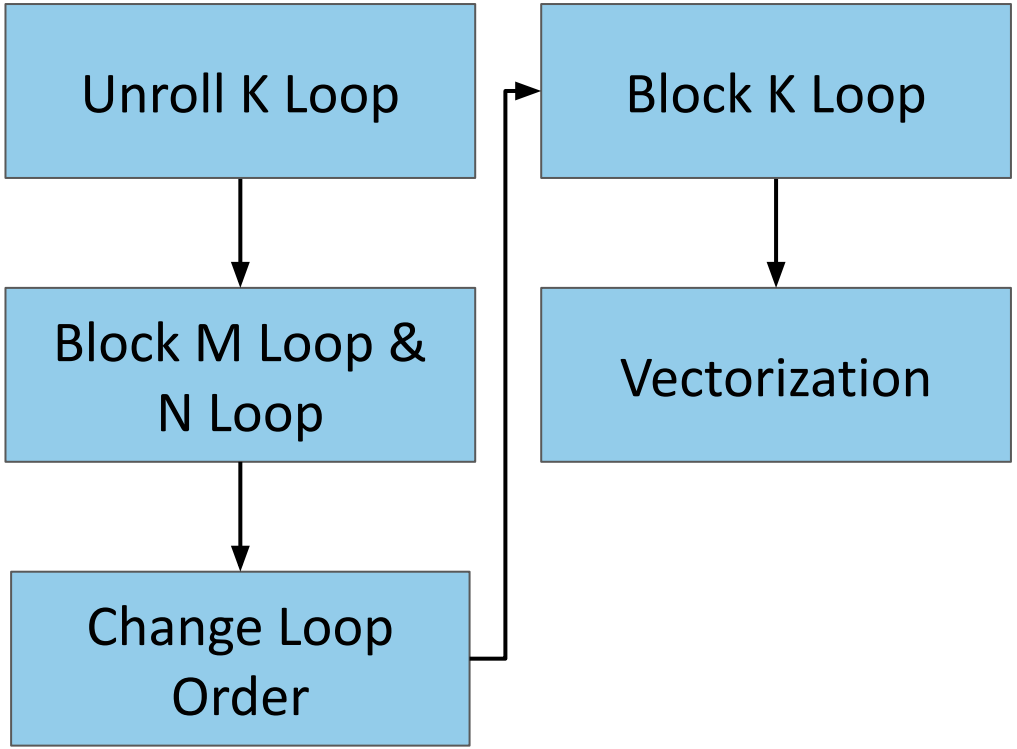
GEMM + PReLU - Blocking over K

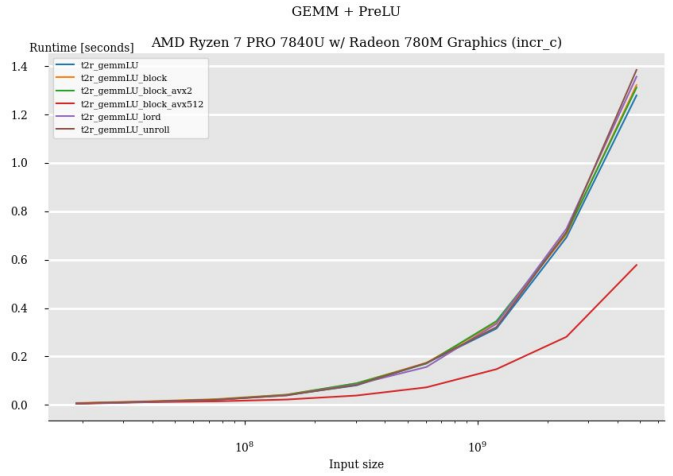


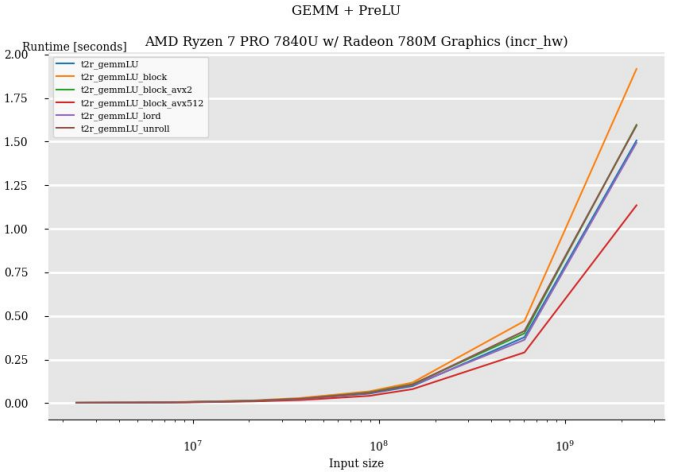




GEMM + PReLU - Vectorization with AVX2 and AVX512





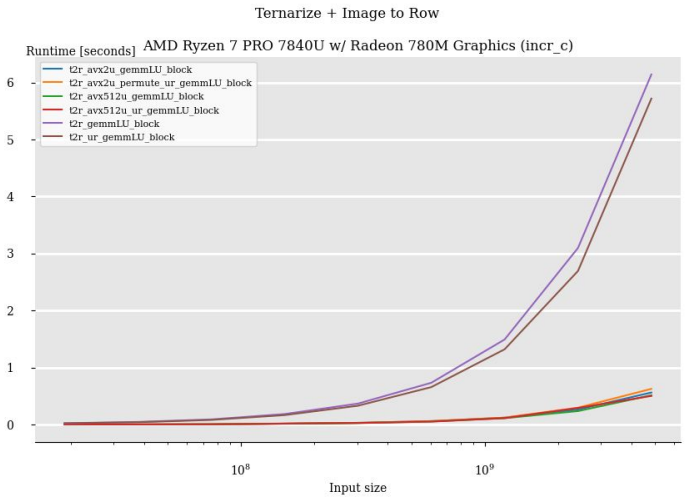


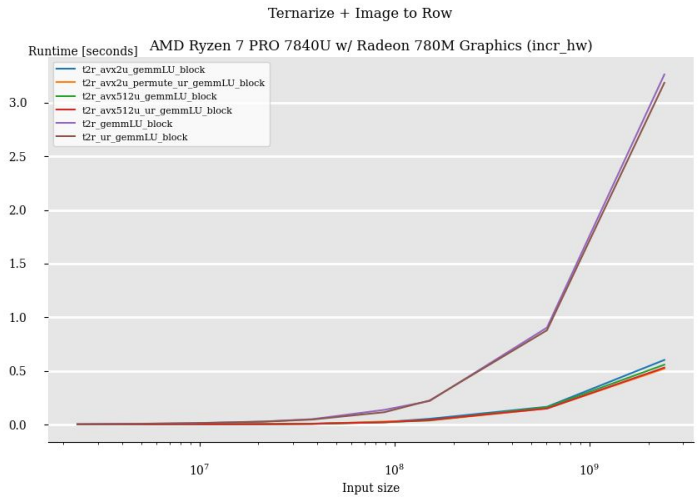
Optimizing Ternarize + im2row

Loop Unrolling

AVX2

AVX512



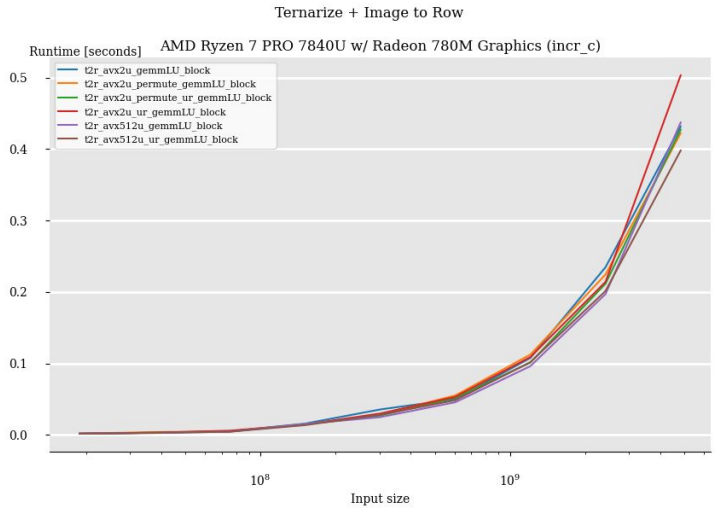


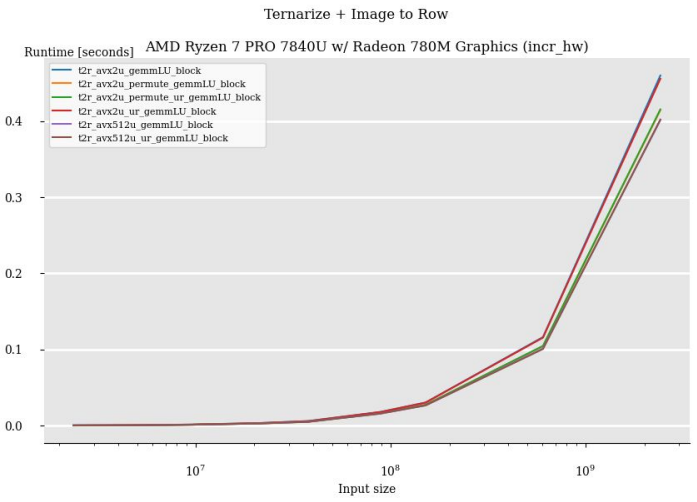
Optimizing Ternarize + im2row

Loop Unrolling

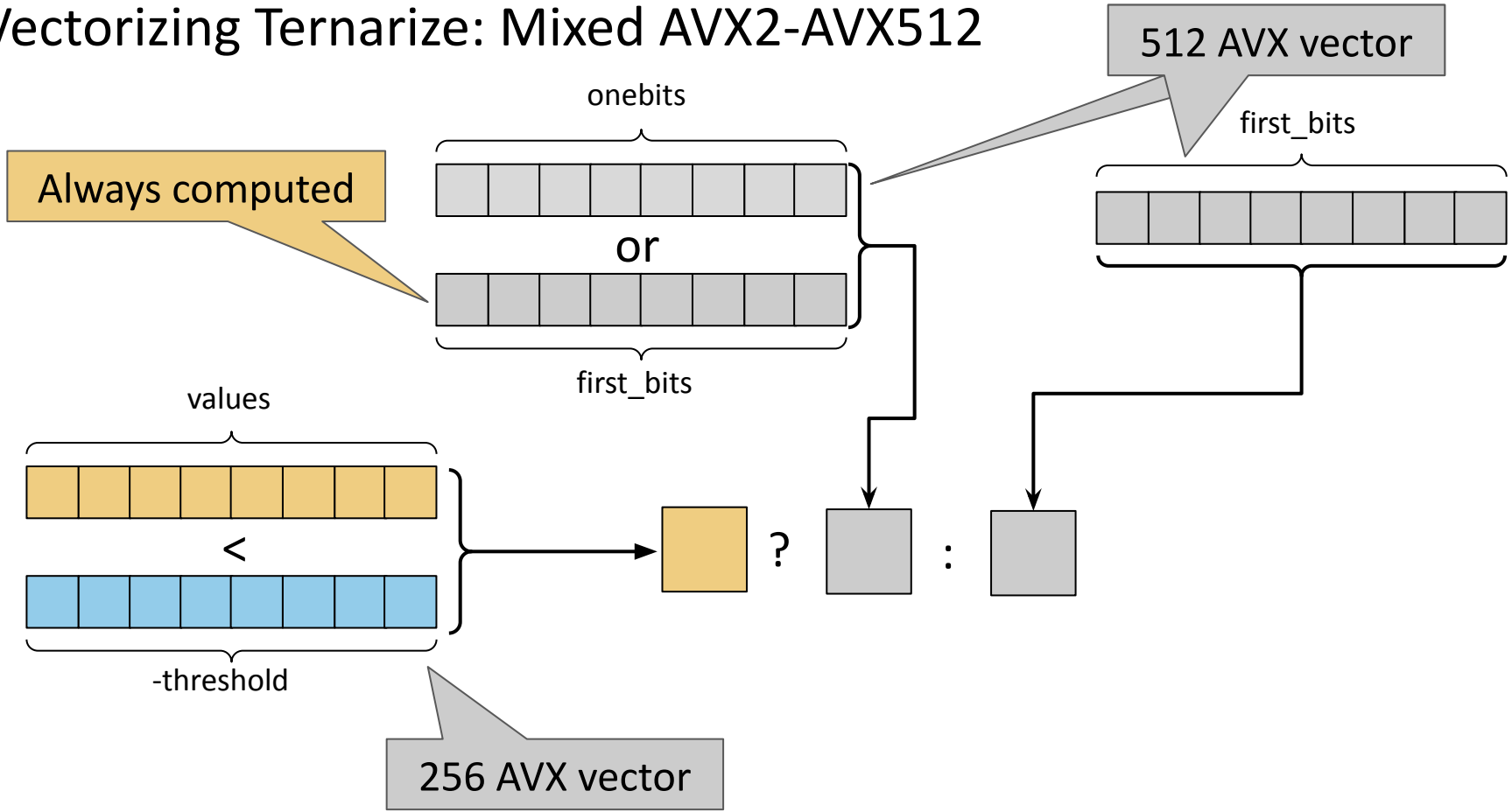
AVX2

AVX512

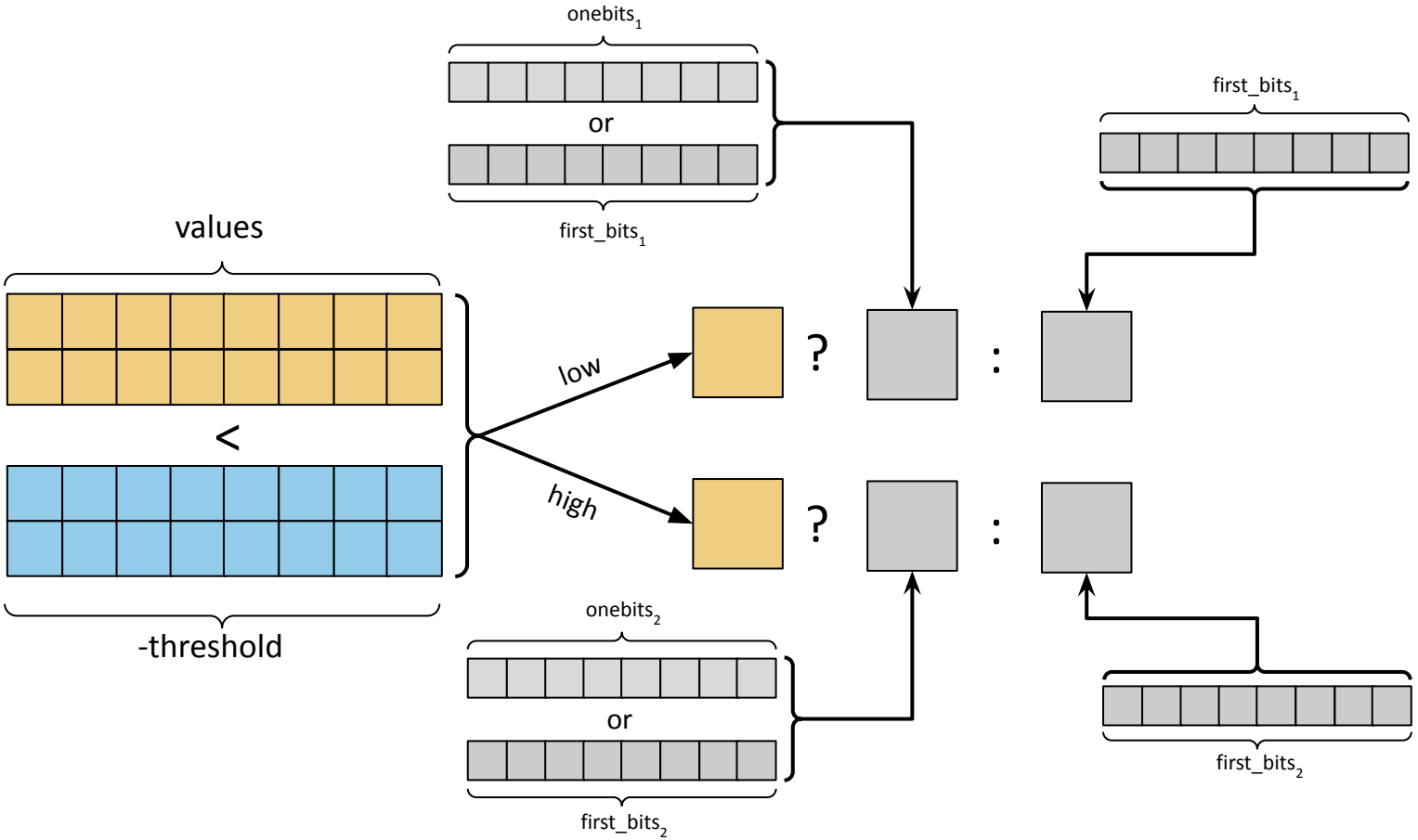




Vectorizing Ternarize: Mixed AVX2-AVX512

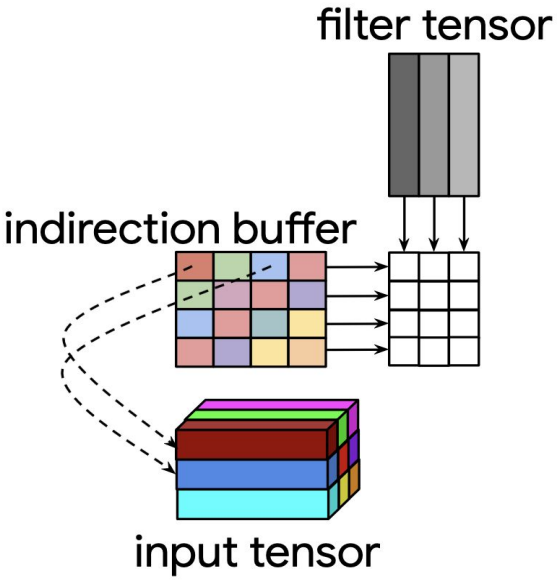


Vectorizing Ternarize: AVX512 only

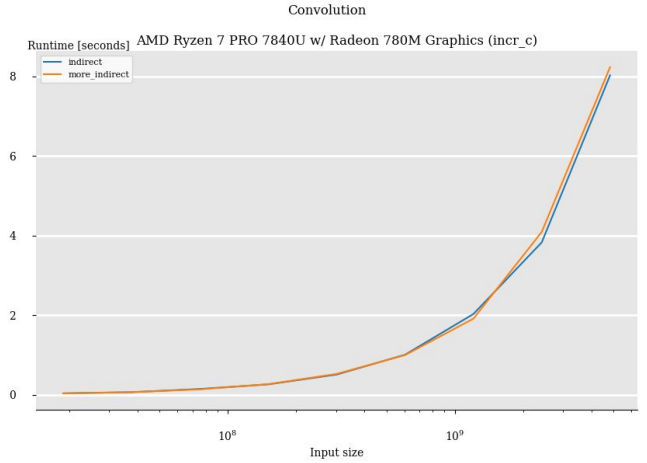


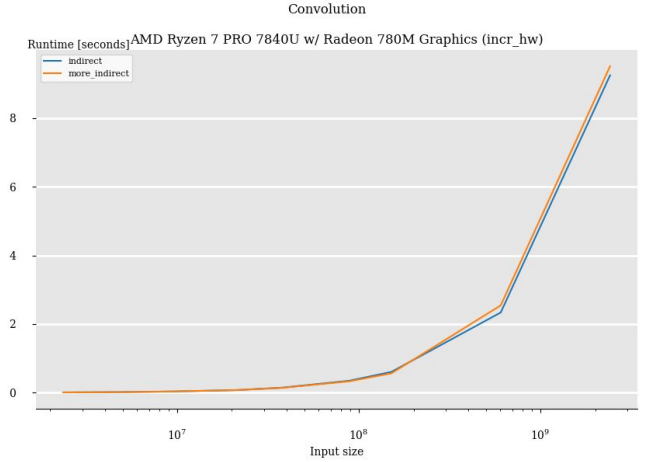
im2row + GEMM

Indirect Convolution

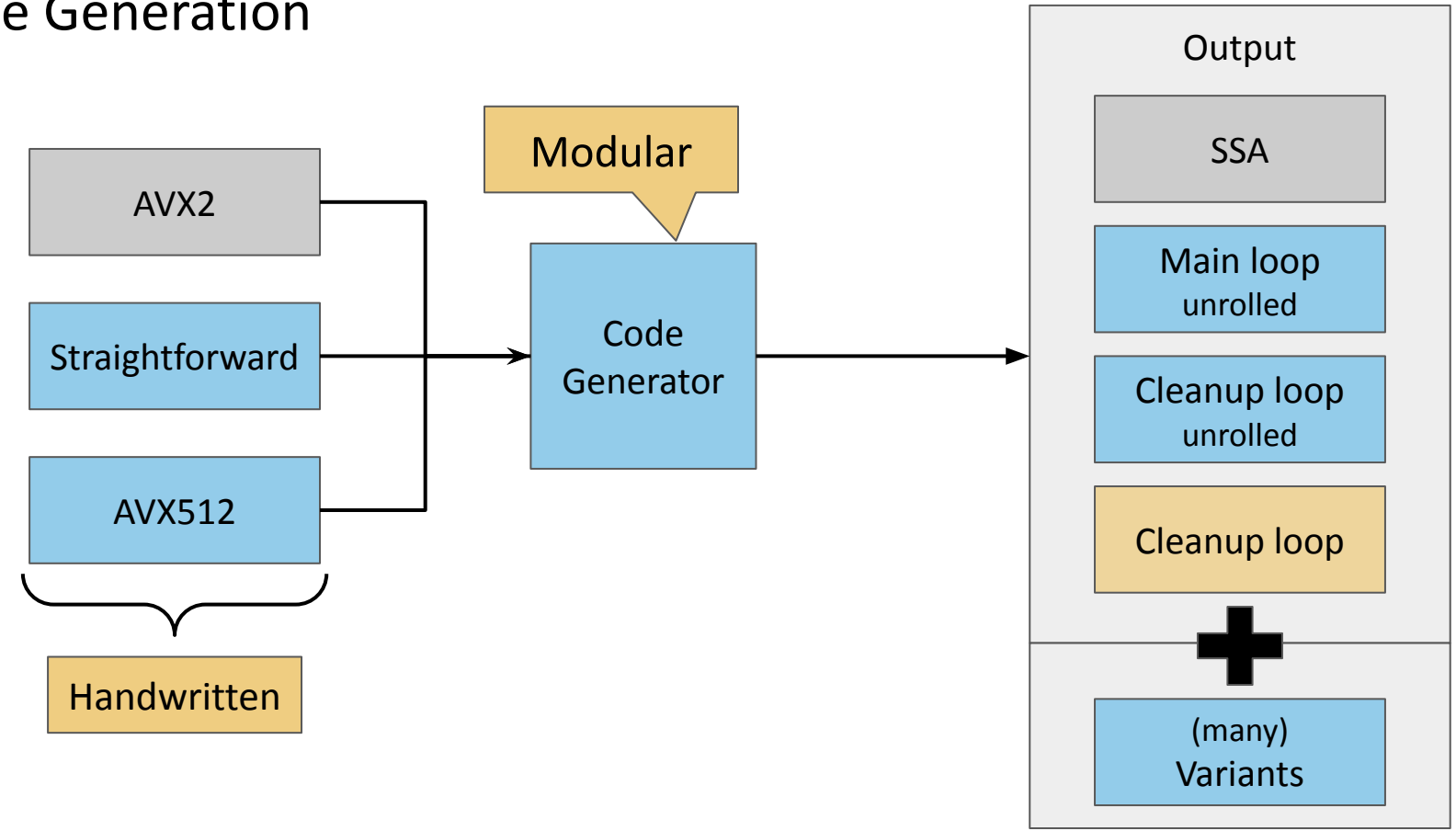


Source: M.Dukhan, The Indirect Convolution Algorithm, *arxiv 2019*

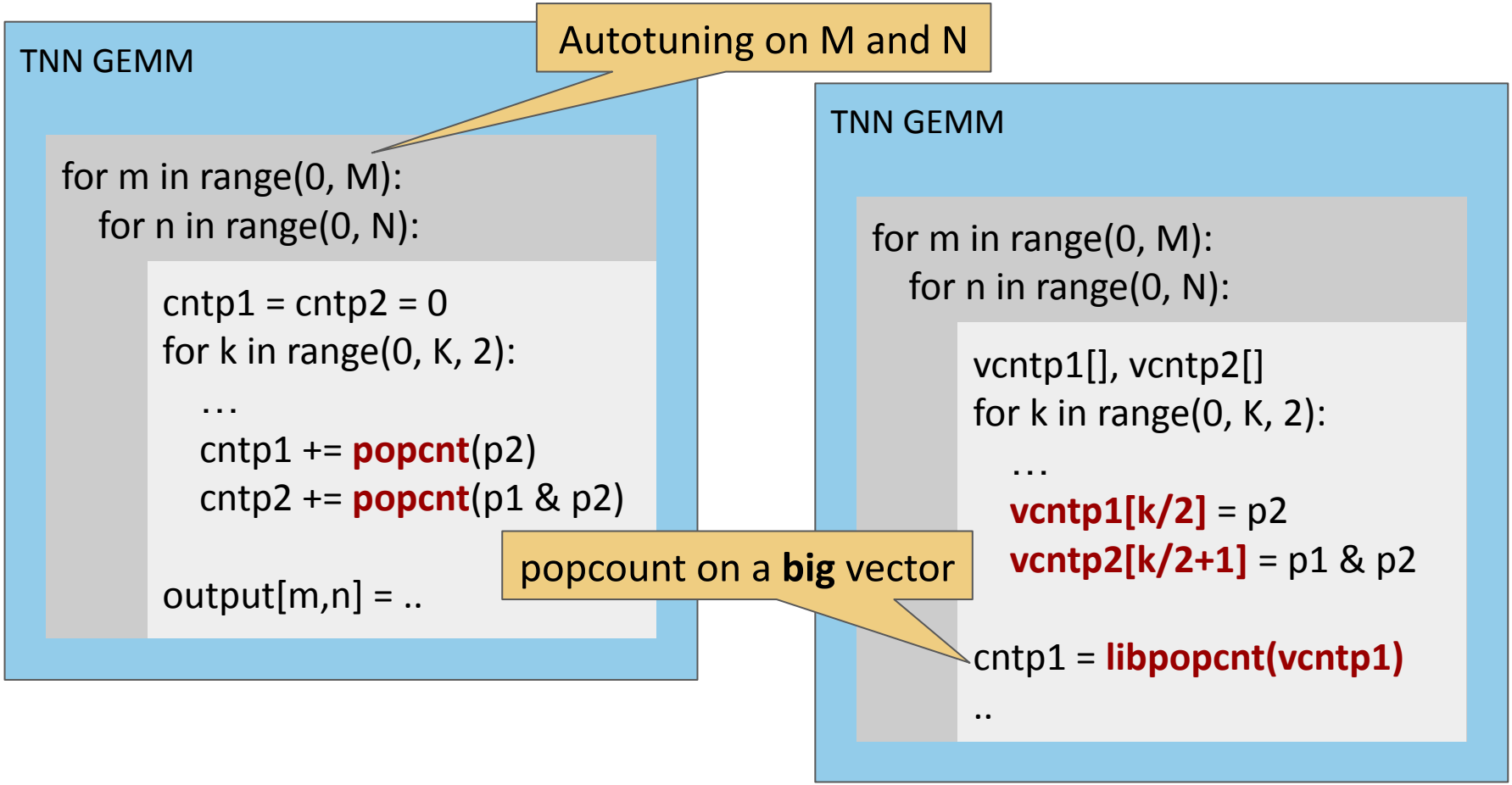




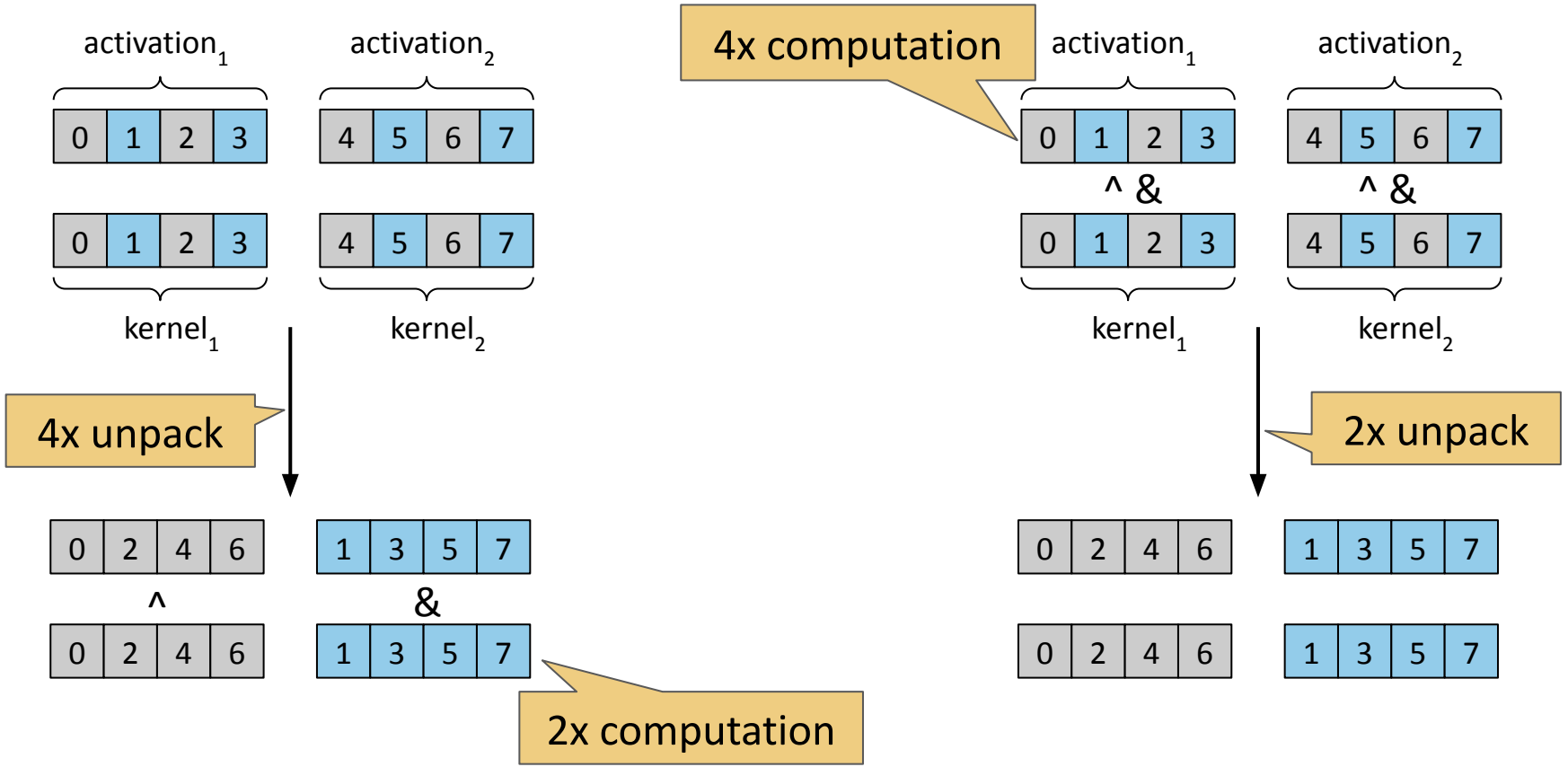
Code Generation



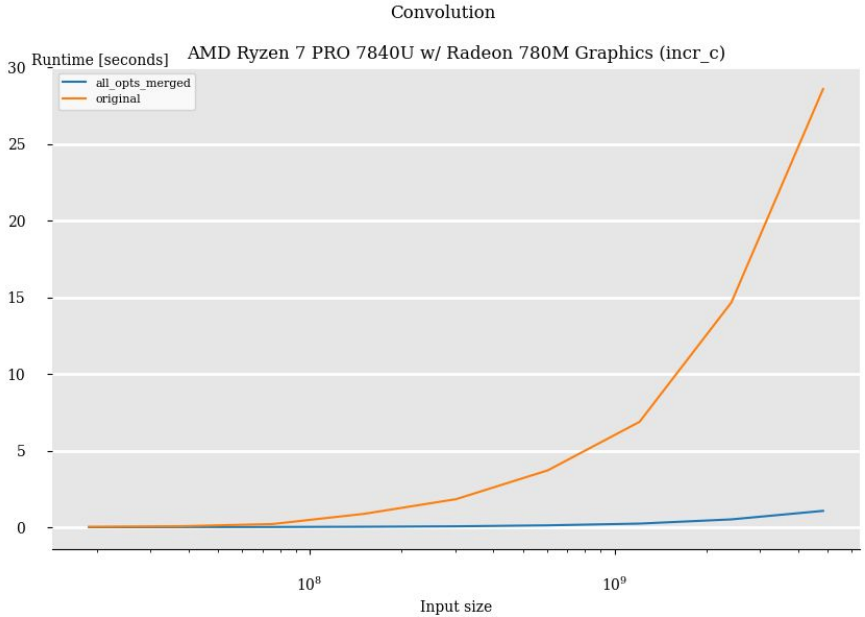
Code Generation - Variant: Autotuning + libpopcnt

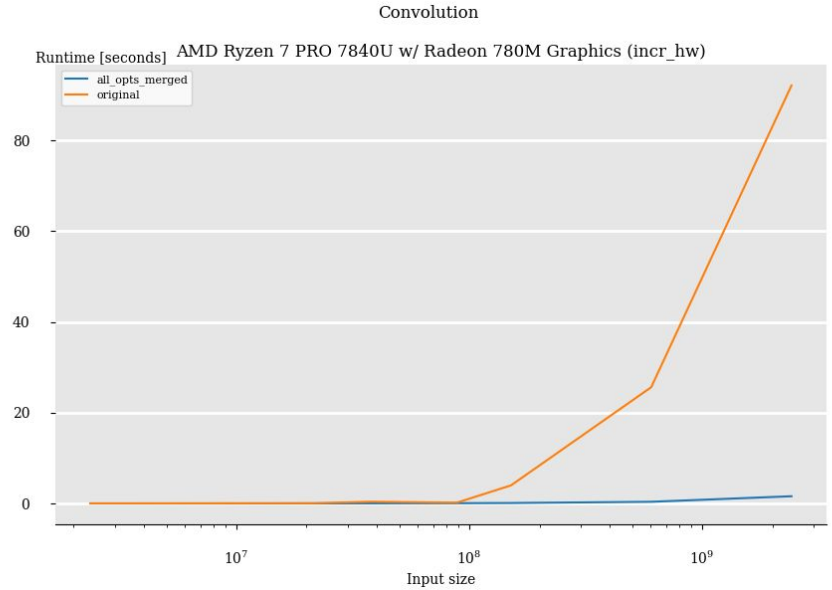


Code Generation - Variant: Less Unpacking

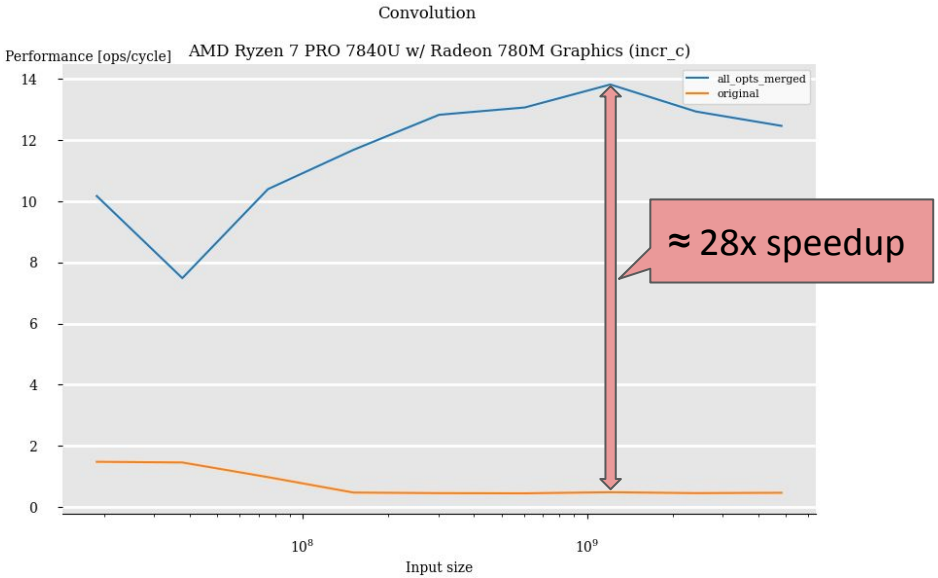


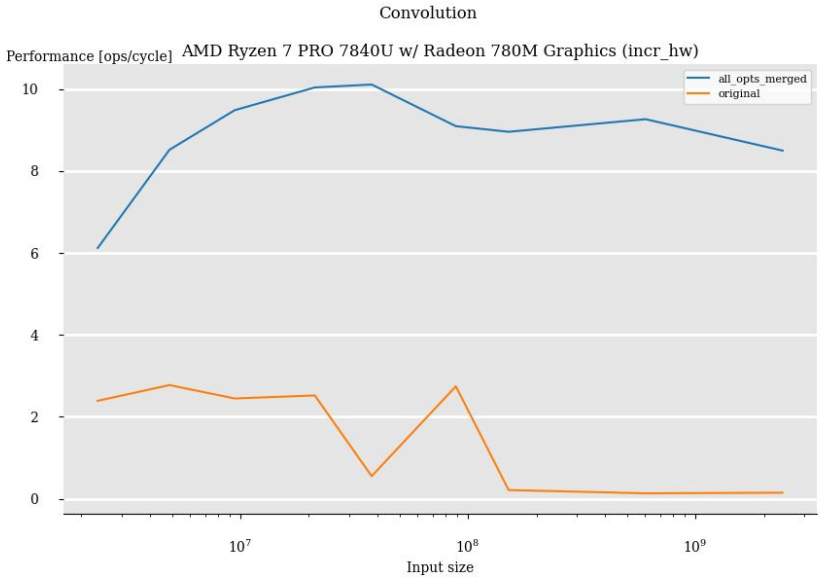
Comparing to the Original



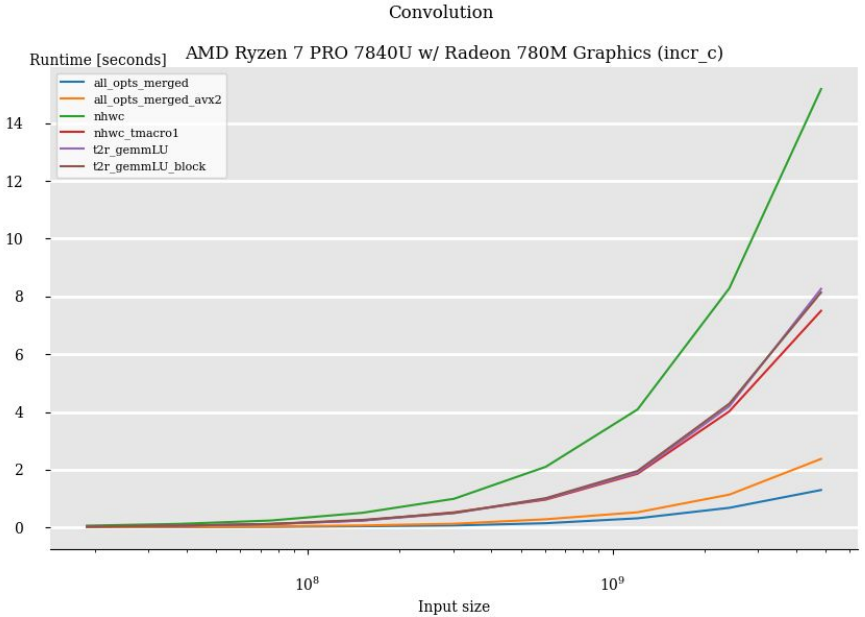


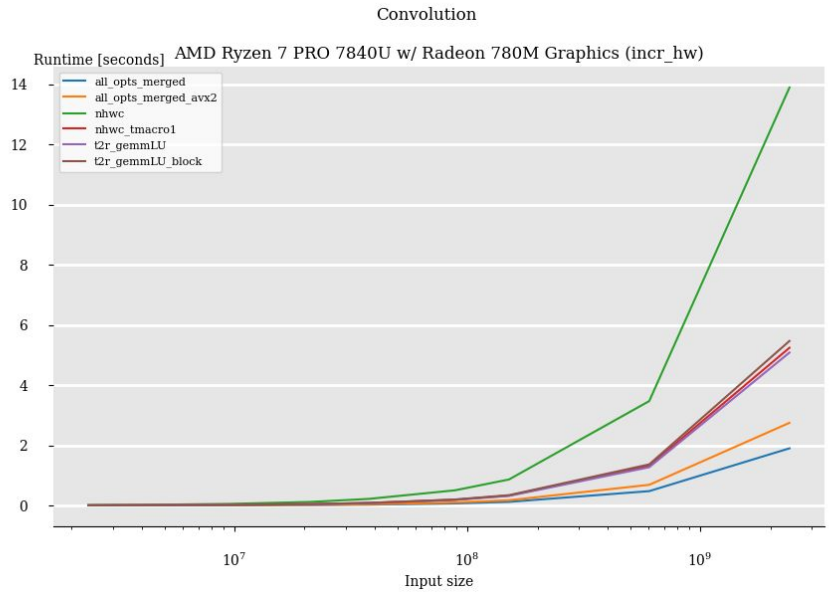
Comparing to the Original





Result: Optimization Steps





Result: Optimization Steps