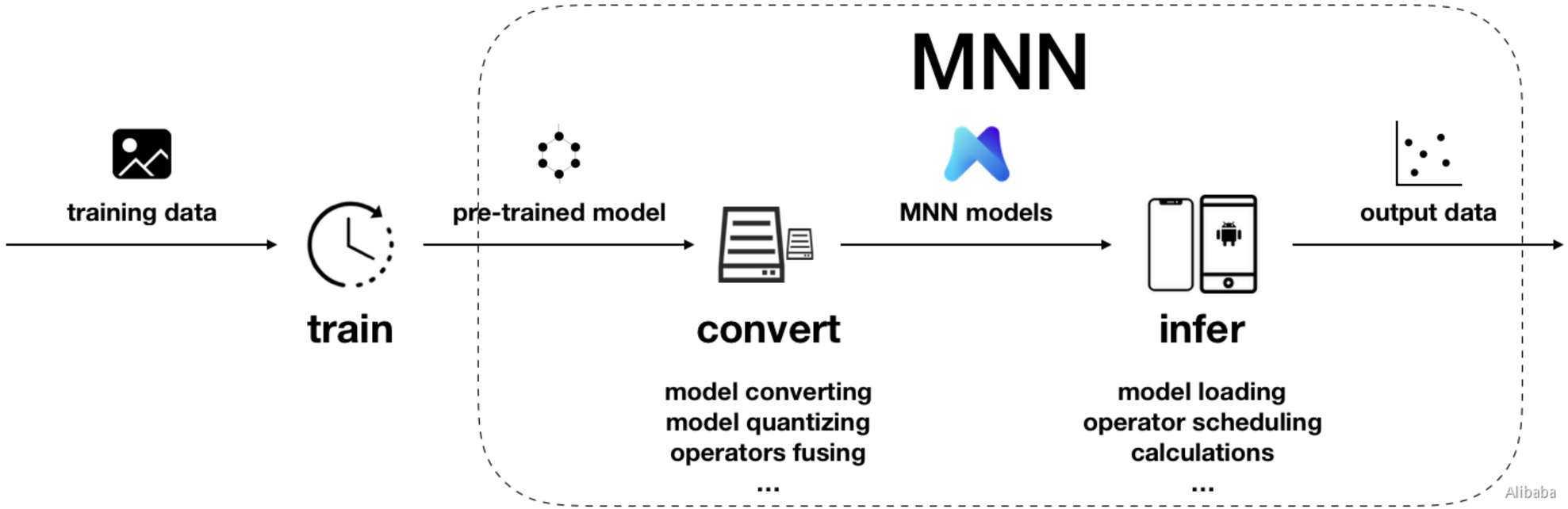
Alibaba Group

MNN's Solution of Heterogeneous Computing - Geometry Compute

Jiang Xiaotang

MNN - A Universal and Fast Inference Engine



MNN Support:

Model: CV / NLP / ASR /...

Inception	Mobilenet	Shufflenet
Yolo	NanoDet	GAN
RNN	Bert	Transformer
	.....	

Framework

Hardware

X64 GPU ARM NPU

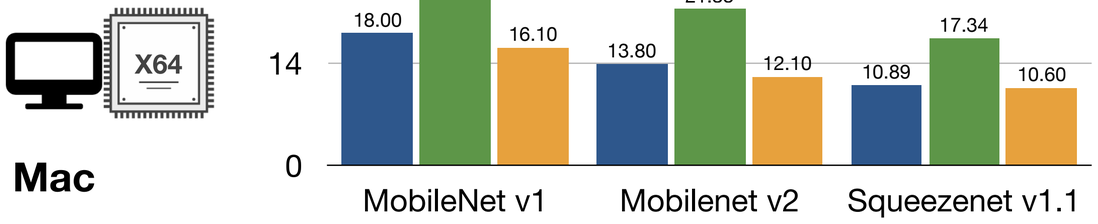
Device

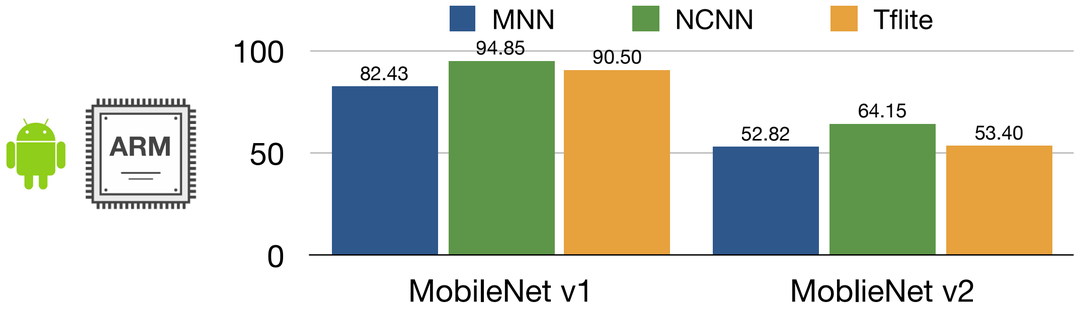
System

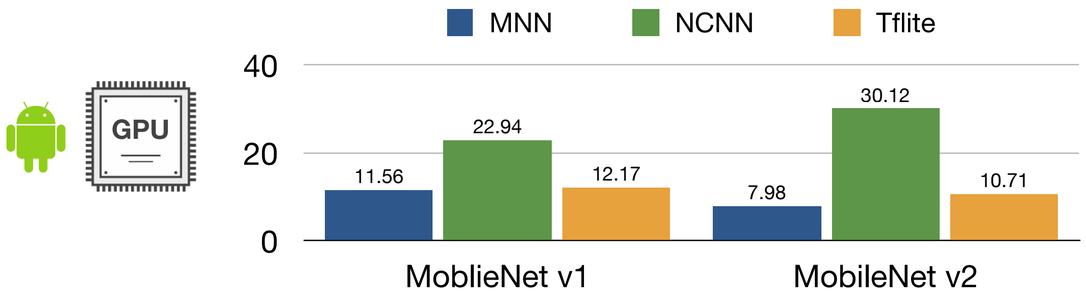
iOS

MNN - A Universal and Fast Inference Engine

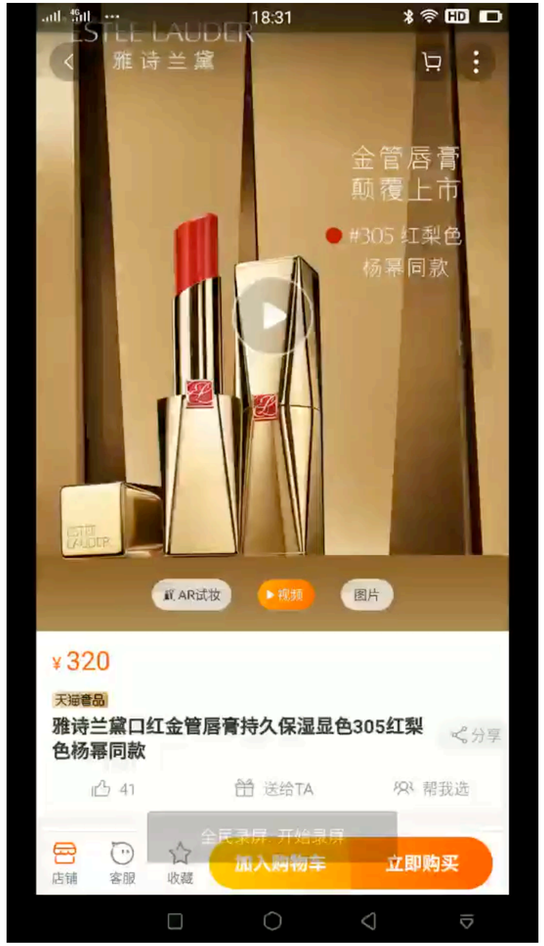
- Method for High Performance
  - Op Fusion
  - NC4HW4 Memory Layout
  - Winograd Convolution / Deconvolution
  - Strassen Matrix Multiply
  - Hand-write Assembly
  - SIMD: NEON / SSE / AVX2 / AVX512
  - CL / GLSL / CUDA
  - Low-precision: FP16 / BF16 / Int8
  - Auto-Tuning



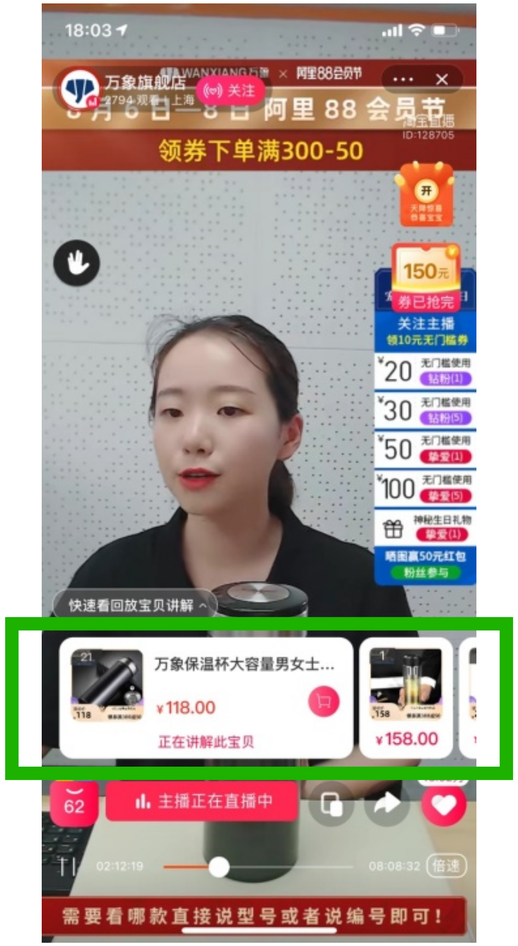




User Case



AR



Live

Image Recognition

Speech Recognition

Challenge of Universal Inference Engine

Alibaba Group

Models

Resnet Inception VGG Mobilenet Shufflenet

Fastrcnn Yolo SSD NanoDet GAN

RNN LSTM Bi-LSTM Bert Transformer

Frameworks

Caffe Onnx Tensorflow

Challenge

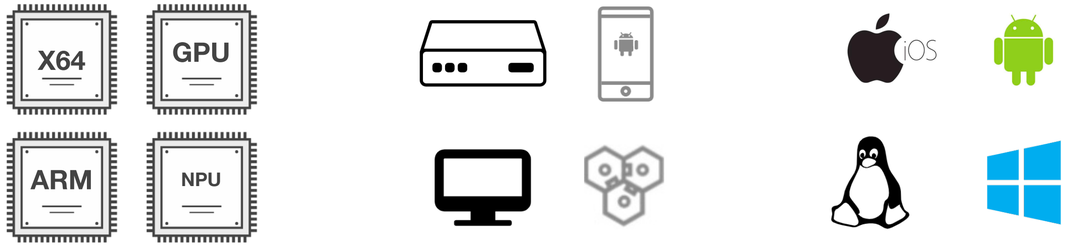
Various operator from different framework

MNN

Various Device for operator implement

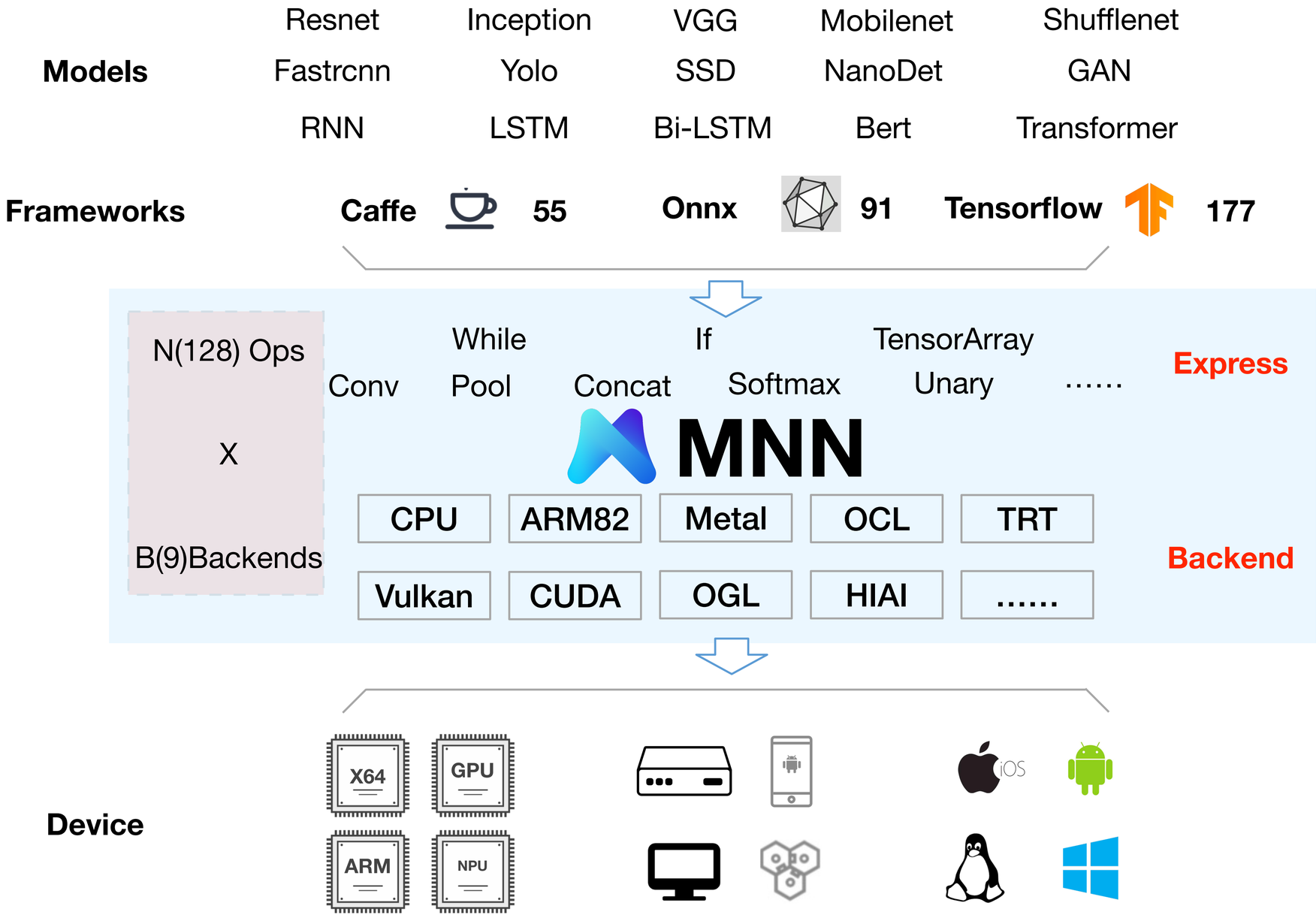
Different API for using same Hardware

Device



Challenge of Universal Inference Engine

Alibaba Group



Classification for Operators

Alibaba Group

MNN-Express Support

Control flow

While If

Heavy Work O(NB)

Data Reorder

TensorArrayGather TensorArraySplit

TensorArrayScatter TensorArrayConcat

Concat SpaceToDepth

Reshape BatchToSpace

Permute Crop

Pad Slice

.....

Fused

Resize Conv3D

RNN Dilate

Conv2D Pool

Softmax Deconv

NMS LRN

MatMul LSTM

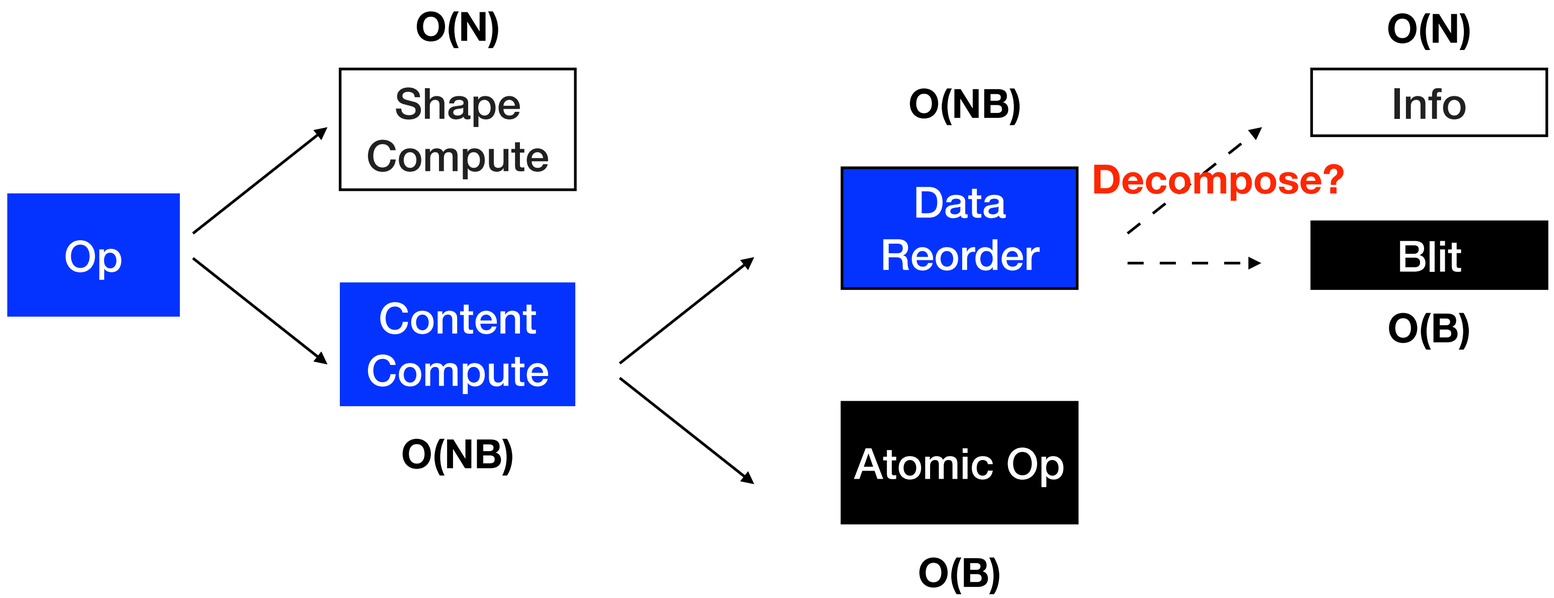
.....

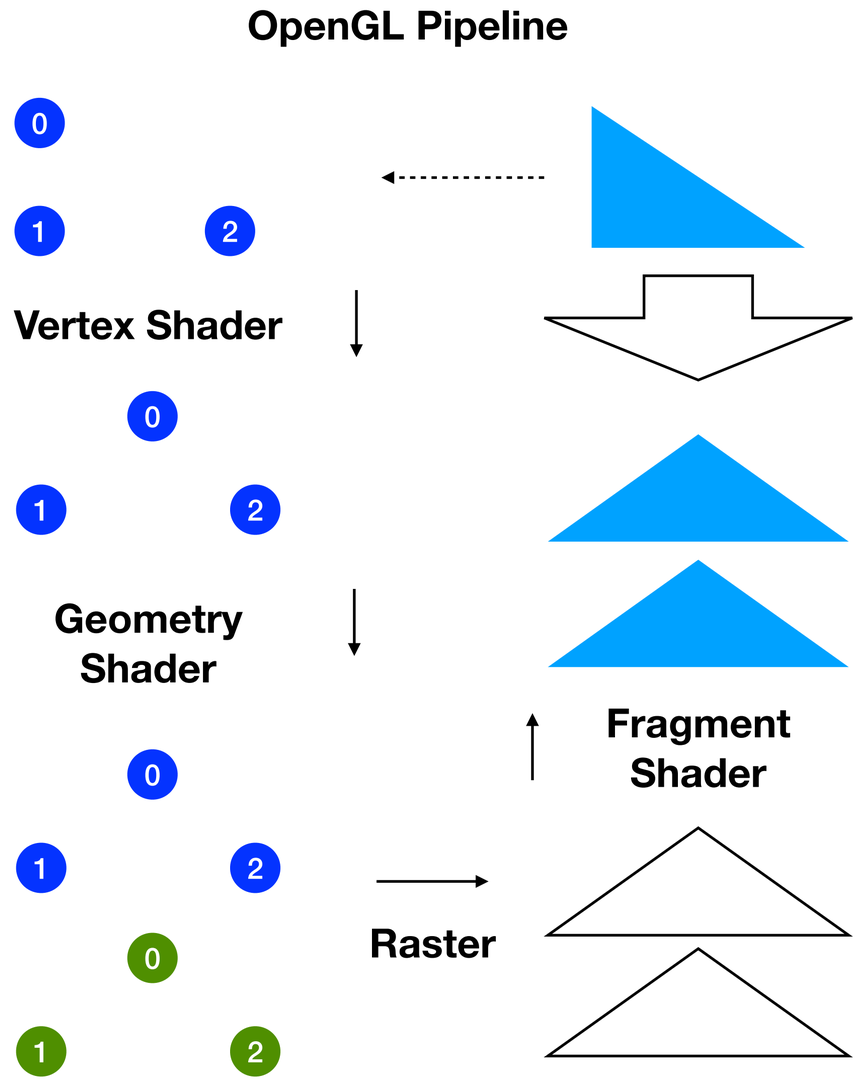
Easy to Support O(B)

Atomic

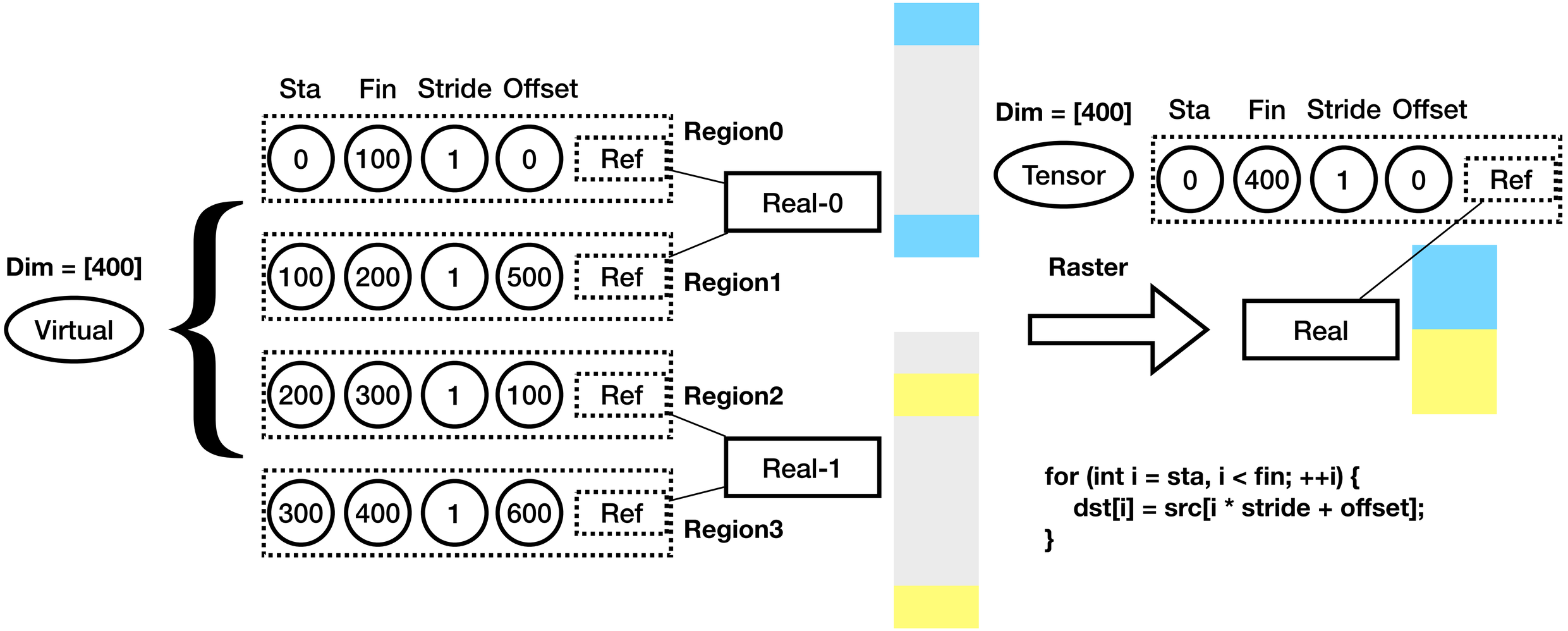
Unary Reduce Binary

Operator Decomposition



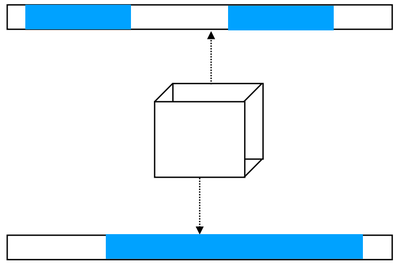


Virtual Tensor and Raster



Geometry Compute

```
struct View {
    int32_t offset = 0;
    int32_t stride[3] = {1, 1, 1};
};
struct Region {
    View src;
    View dst;
    int32_t size[3] = {1, 1, 1};
    Tensor* origin;
};
```



```
For x, y, z in range(size):
  Dst[x * dst.stride[0]
   + y * dst.stride[1]
   + z * dst.stride[2]
   + dst.offset
  ] = Src[x * src.stride[0]
   + y * src.stride[1]
   + z * src.stride[2]
   + src.offset]
```

Data Reorder Op:
Compute Region

A: [2, 4] → **Transpose [1, 0]**

Virtual B: [4, 2]

Region0:
srcOffset: 0
DstOffset:0
Size: 1, 4, 2
dstStride: 0, 2, 1
srcStride: 0, 1, 4

A: [2, 4] → **Slice [1, 1, 0, 4]**

Virtual B: [1, 4]

Region0:
srcOffset: 4
DstOffset:0
Size: 1, 1, 4
dstStride: 0, 4, 1
srcStride: 0, 4, 1

A: [2, 4]
B: [2, 4] → **Concat Axis = 1**

Virtual C: [2, 8]

Region0:	**Region1:**
srcOffset: 0	srcOffset: 0
DstOffset:0	DstOffset:4
Size: 1, 2, 4	Size: 1, 2, 4
dstStride: 0, 8, 1	dstStride: 0, 8, 1
srcStride: 0, 4, 1	srcStride: 0, 4, 1

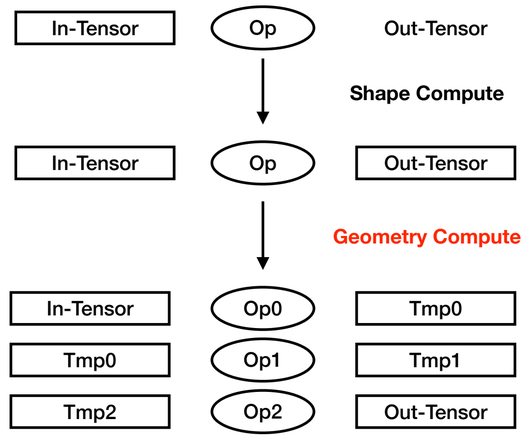
Fused Op: Decompose

Convolution

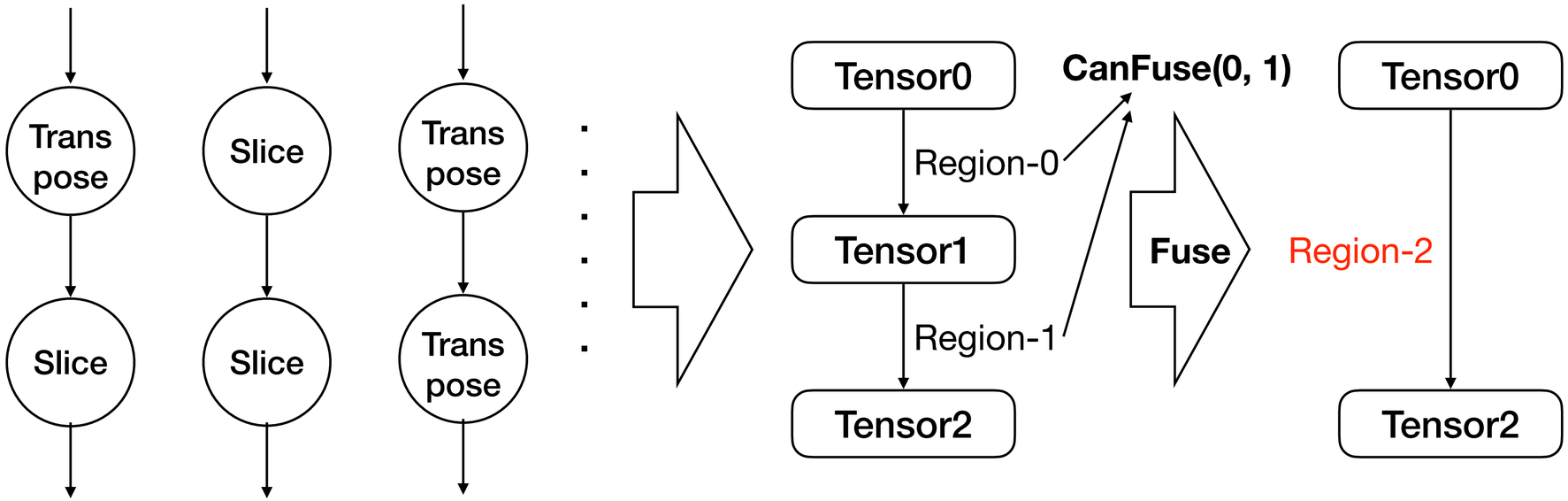
$$X(n, c_i, h, w), W(c_o, c_i, k_h, k_h), Y(n, c_o, h, w)$$
$$Y = Conv(X, W)$$

Im2Col + GEMM

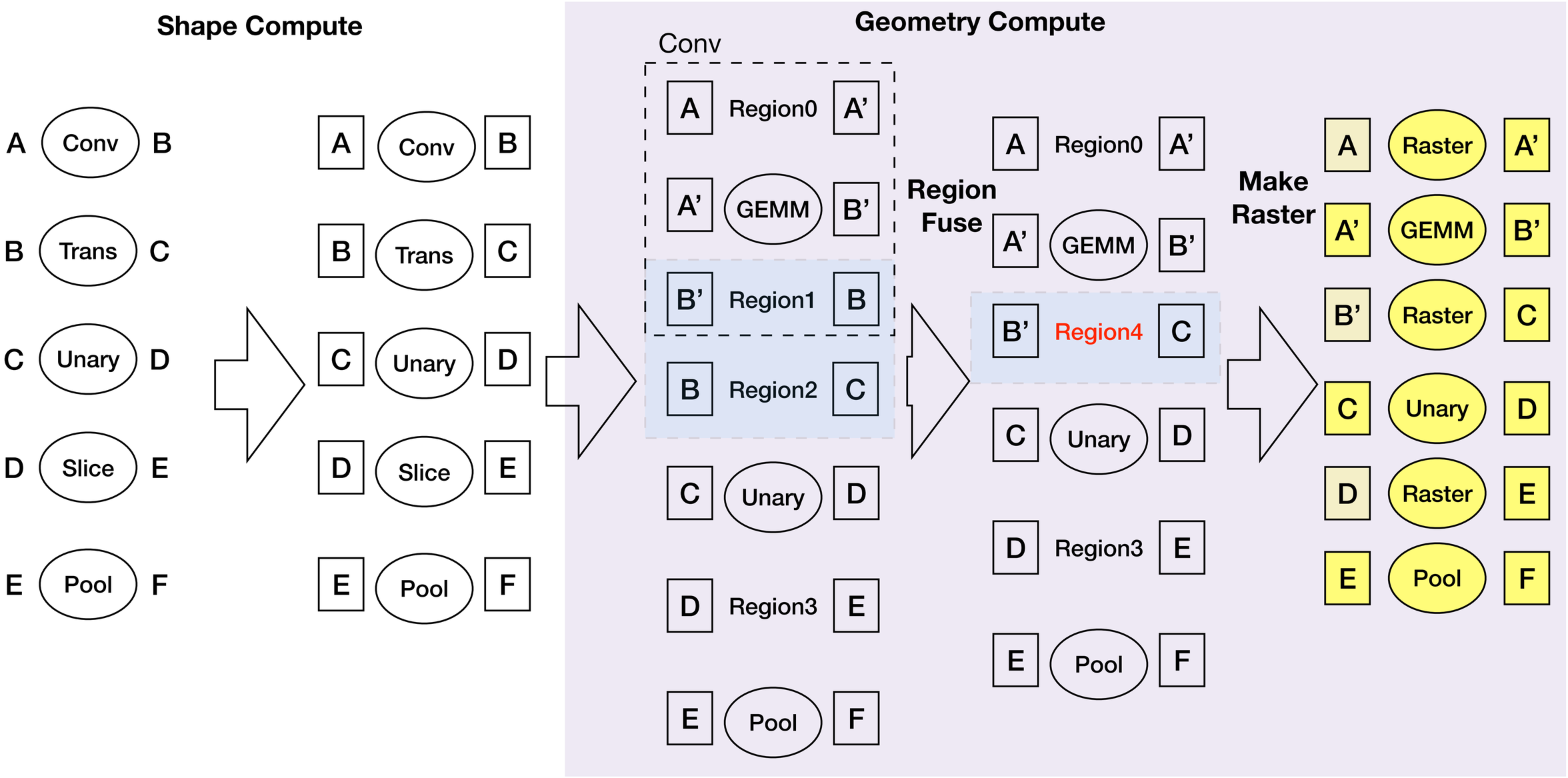
$$OX = Reorder(X)([n, c_i, h, w] \rightarrow [nhwc_o, c_i k_h k_w])$$
$$OW = Reorder(W)([c_o, c_i, k_h, k_w] \rightarrow [c_o, c_i k_h k_w])$$
$$OY = MatMul(OX, OW)$$
$$Y = IndexMapping(OY)$$



Region Fuse


Trans pose
Slice
Slice
Slice
Trans pose
Trans pose
Tensor0
Region-0
Tensor1
Region-1
Tensor2
CanFuse(0, 1)
Fuse
Tensor0
Region-2
Tensor2

Complement in Inference Engine



New Architecture Based on Geometry Compute

Alibaba Group

Traditional Architecture

$$T = C_{bn}BN$$

CPU

Convolution	Transpose
Deconvolution	Slice
LSTM	Unary
GRU	Reduce
MatMul	Concat
Binary	BatchToSpace

OCL

Convolution	Transpose
Deconvolution	Slice
LSTM	Unary
GRU	Reduce
MatMul	Concat
Binary	BatchToSpace

DSP

Convolution	Transpose
Deconvolution	Slice
LSTM	Unary
GRU	Reduce
MatMul	Concat
Binary	BatchToSpace

CUDA

Convolution	Transpose
Deconvolution	Slice
LSTM	Unary
GRU	Reduce
MatMul	Concat
Binary	BatchToSpace

Geometry Compute Architecture

$$T = C_g(N - N_o) + C_{bn}BN_o$$

Geometry

Convolution
Deconvolution
LSTM
GRU
Transpose
Concat
Slice
BatchToSpace

CPU

Unary
MatMul
Binary
Reduce
Raster

OCL

Unary
MatMul
Binary
Reduce
Raster

DSP

Unary
MatMul
Binary
Reduce
Raster

CUDA

Unary
MatMul
Binary
Reduce
Raster

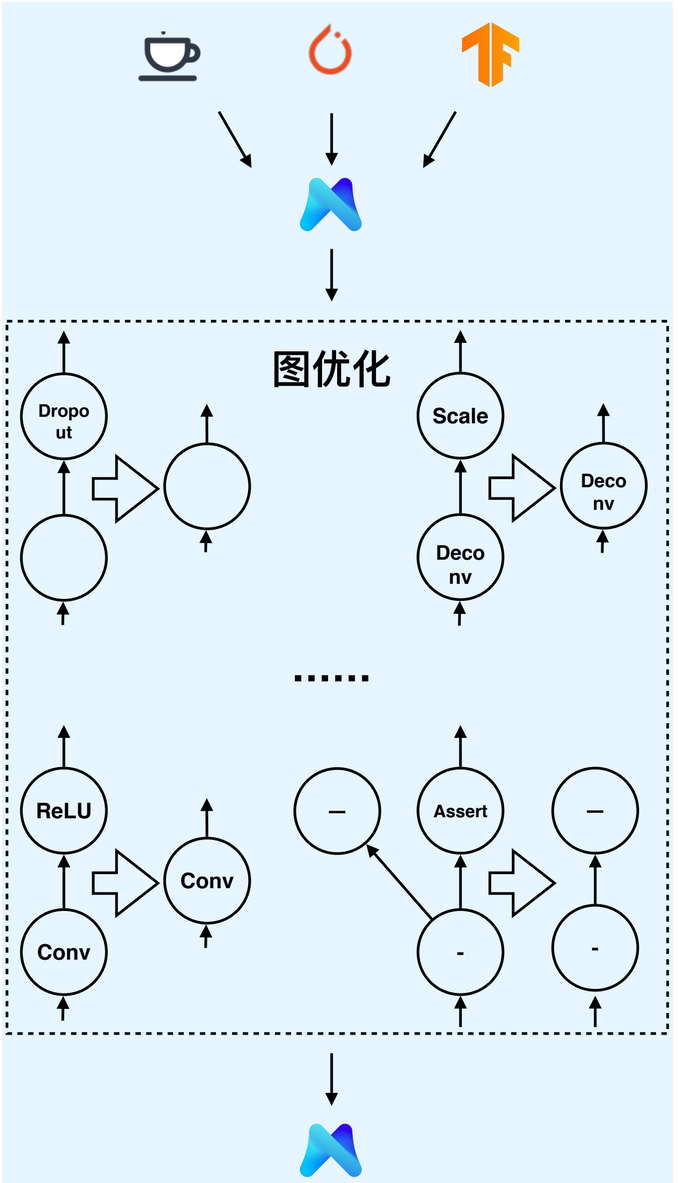
Data Reorder Op use Geometry (No Cost)

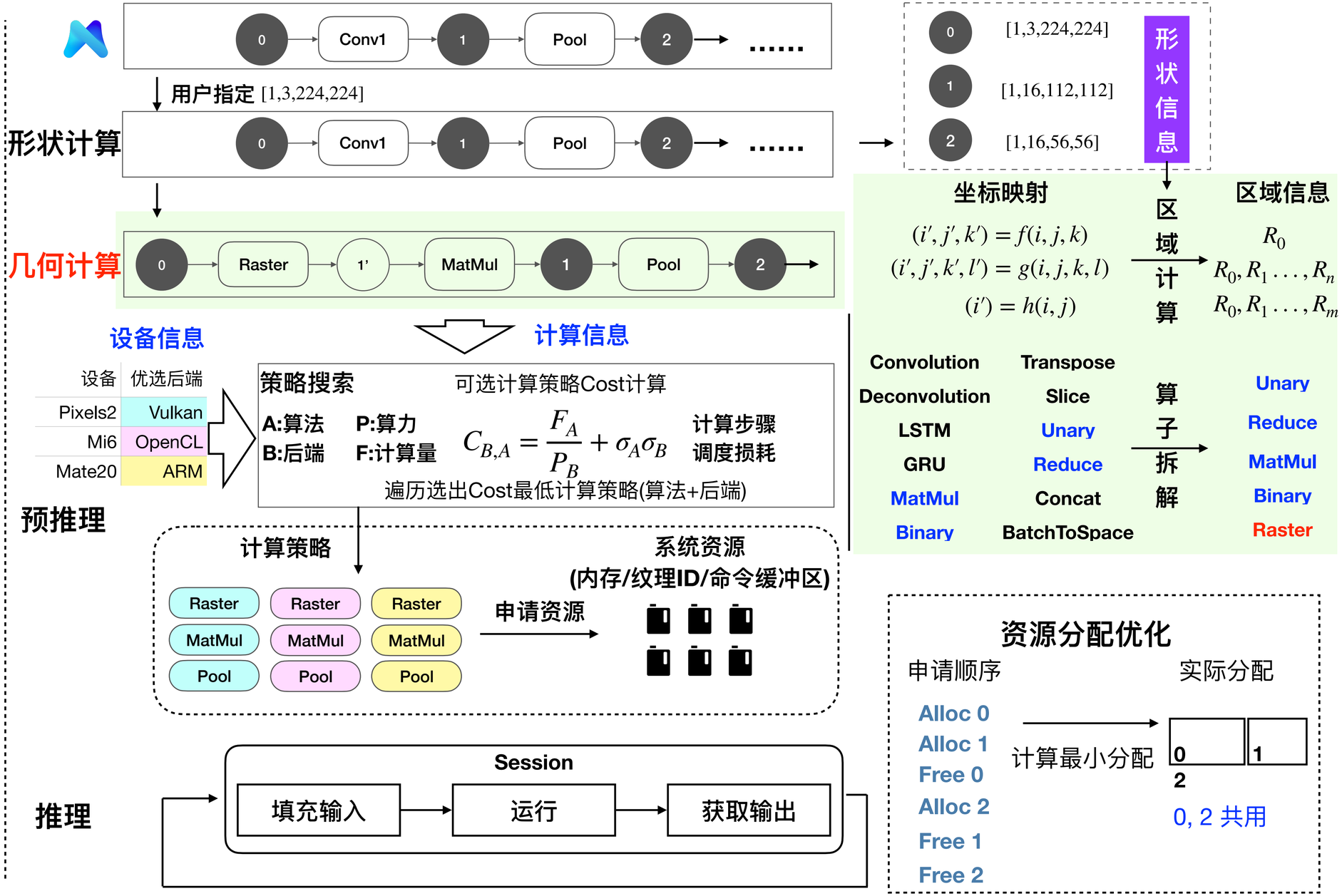
Fused Op decompose (Has Extra Cost)

Core Fused Op hold

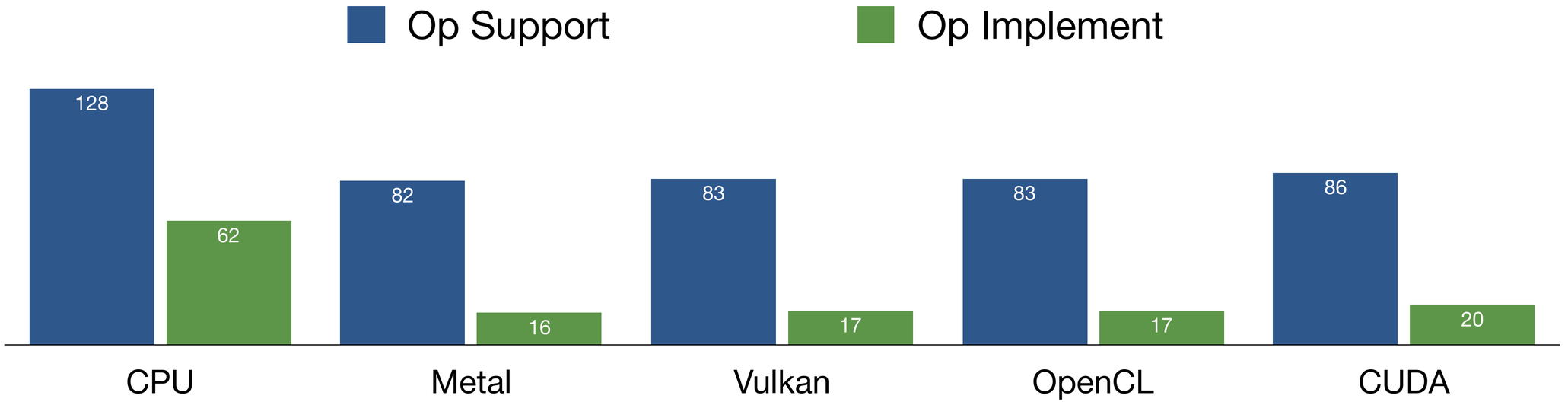
MNN Full Work flow

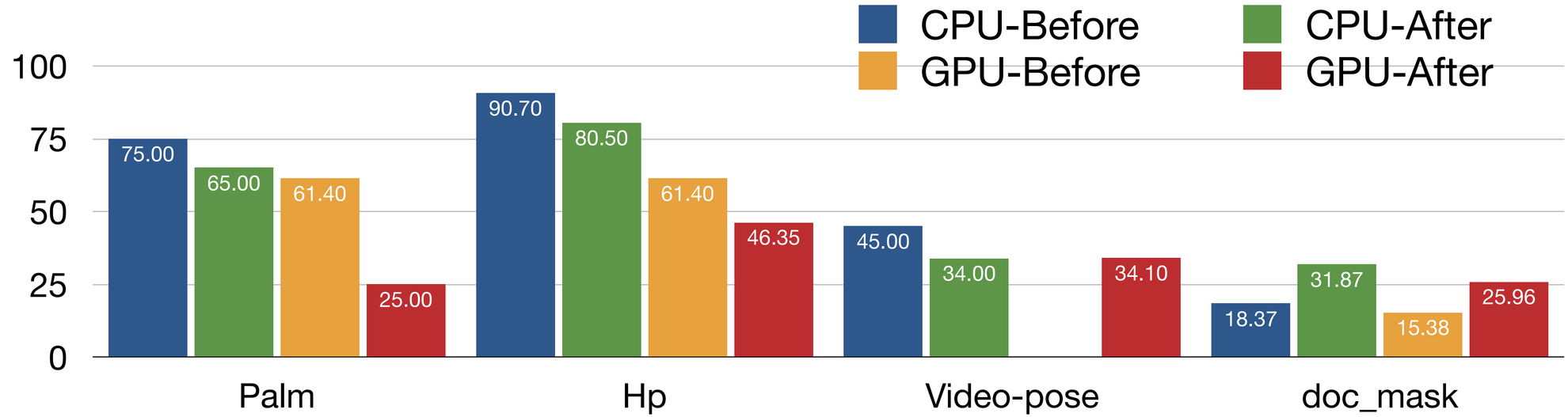
Alibaba Group



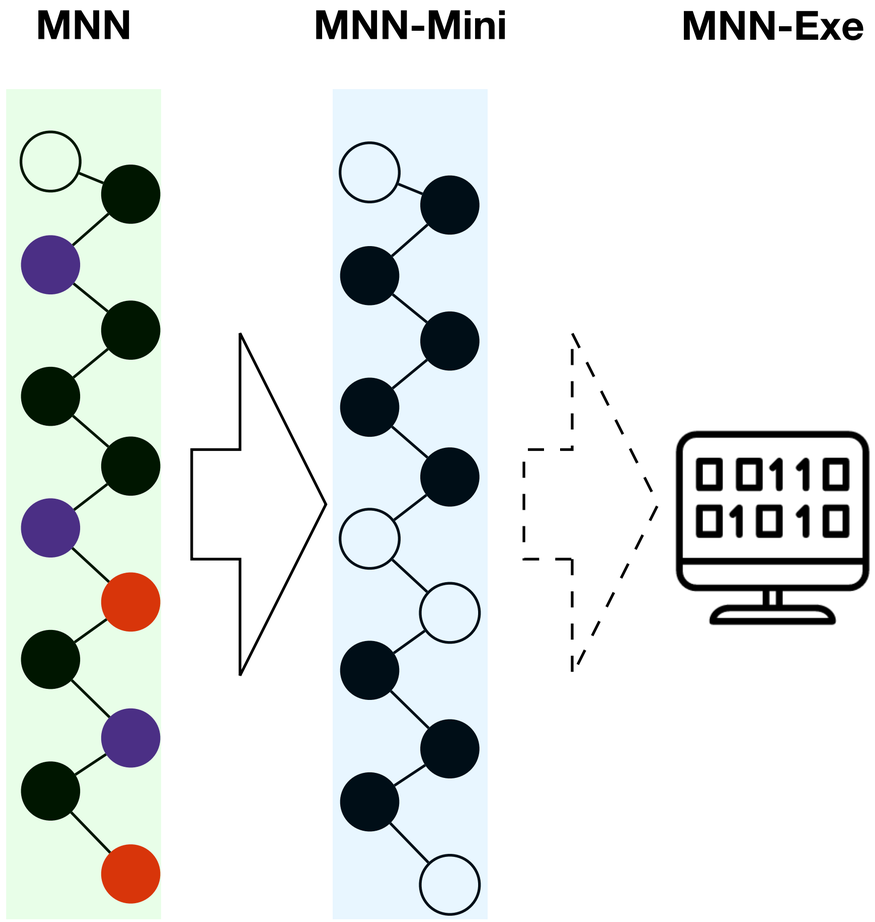


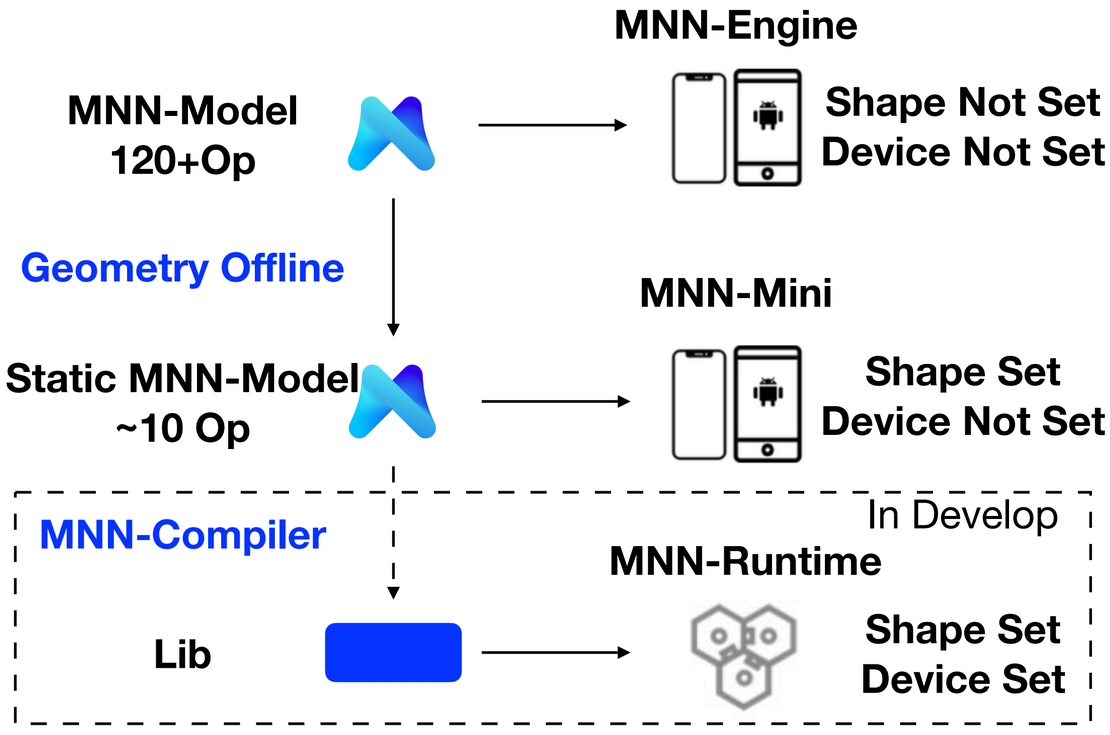
Result of Geometry Compute Refract



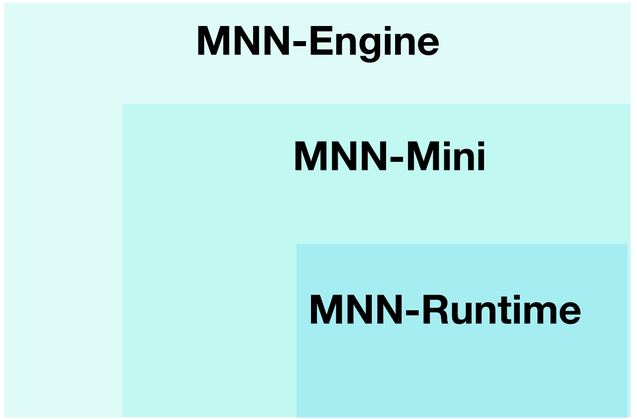


Geometry for Multi-Level Deploy





MNN-Compiler:
Fuse, Codegen
and AutoTurning



Future

Alibaba Group

- https://github.com/alibaba/MNN

- Geometry Compute
  - Speed: Cost Less Decomposition for Fused Op
  - Post-treat/Pre-treat: GIR for more CV / Data Pretreat / Post-treat
  - Train: Grad Based on Geometry Compute
- MNN
  - More Frontend Support, such as Torchscript
  - More Backends for Hardware Support: NPU / DSP
  - Model Compress Algorithm Combined with Compiler Technology

Alibaba Group

THANKS