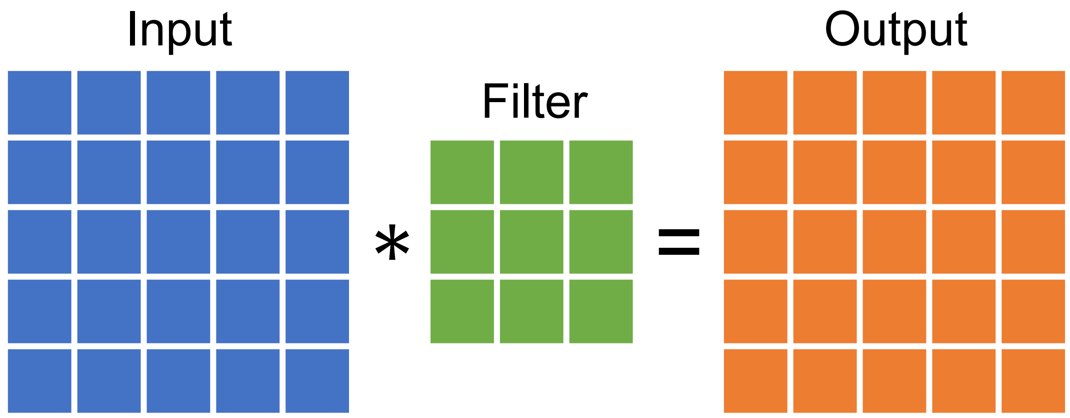
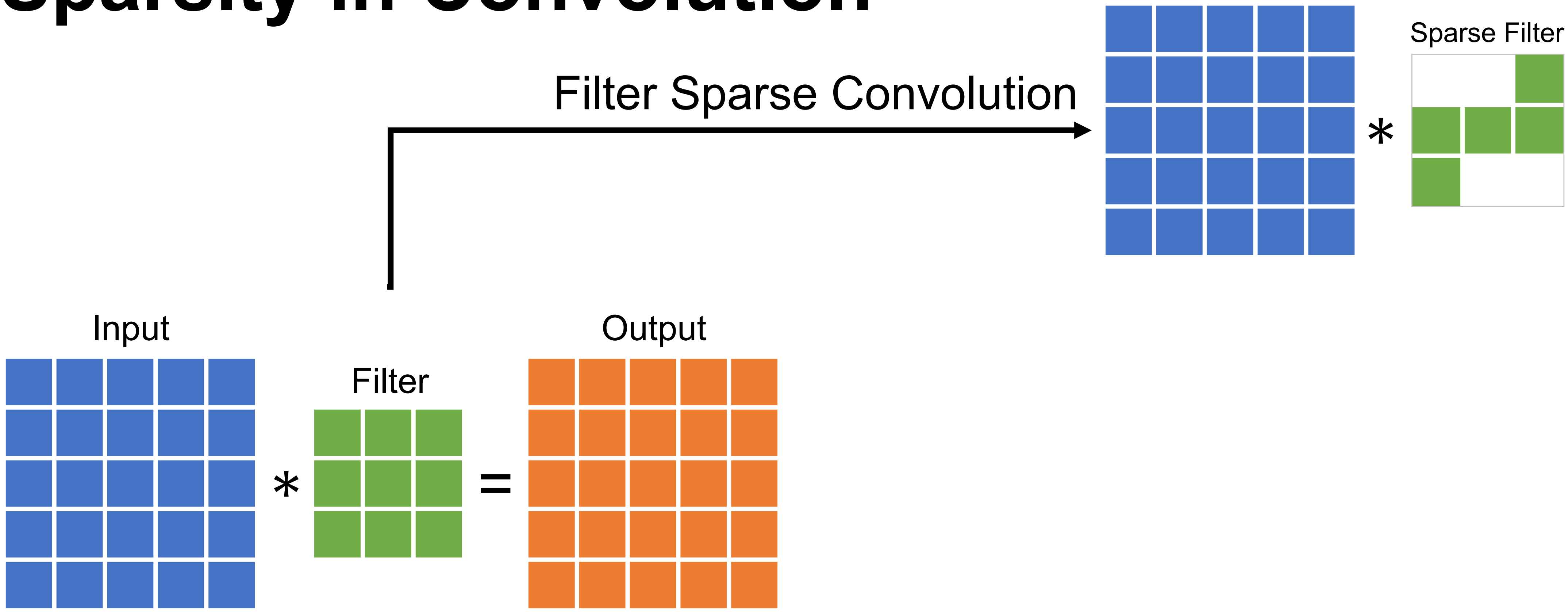
Unified Convolution Framework

Jaeyeon Won, Changwan Hong, Charith Mendis, Joel Emer, Saman Amarasinghe

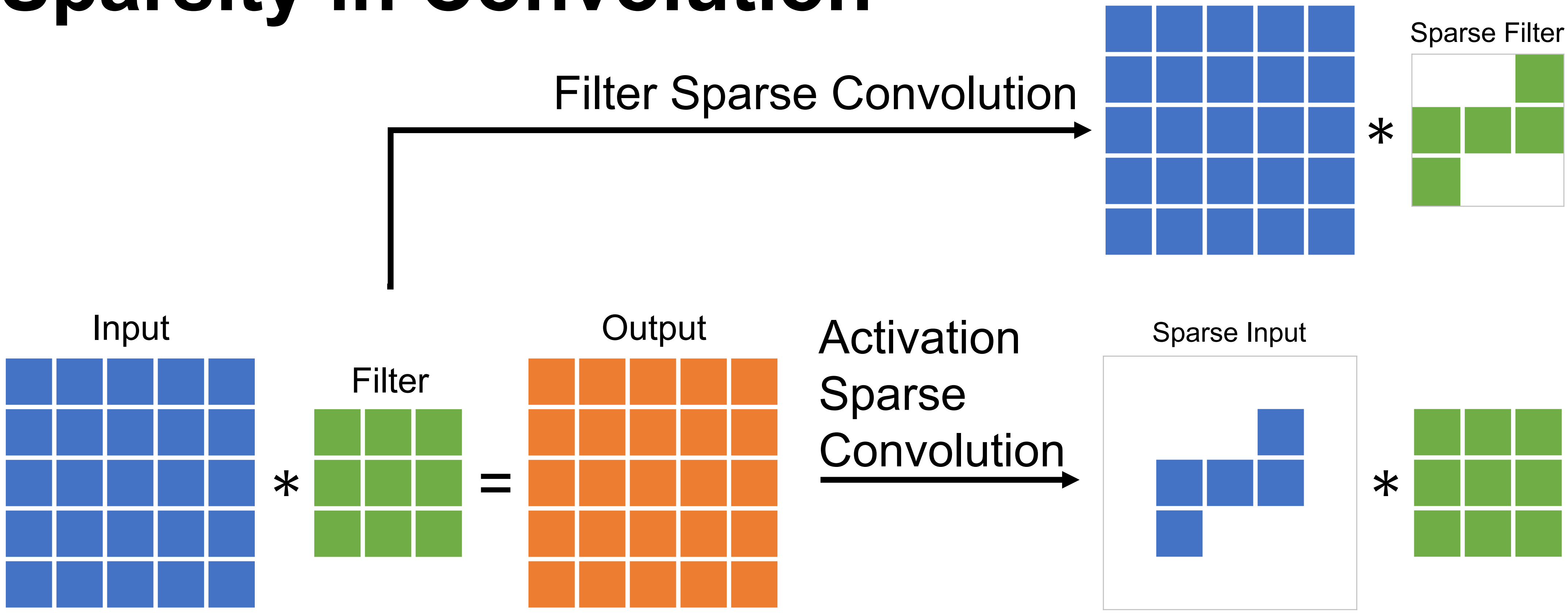
Convolution



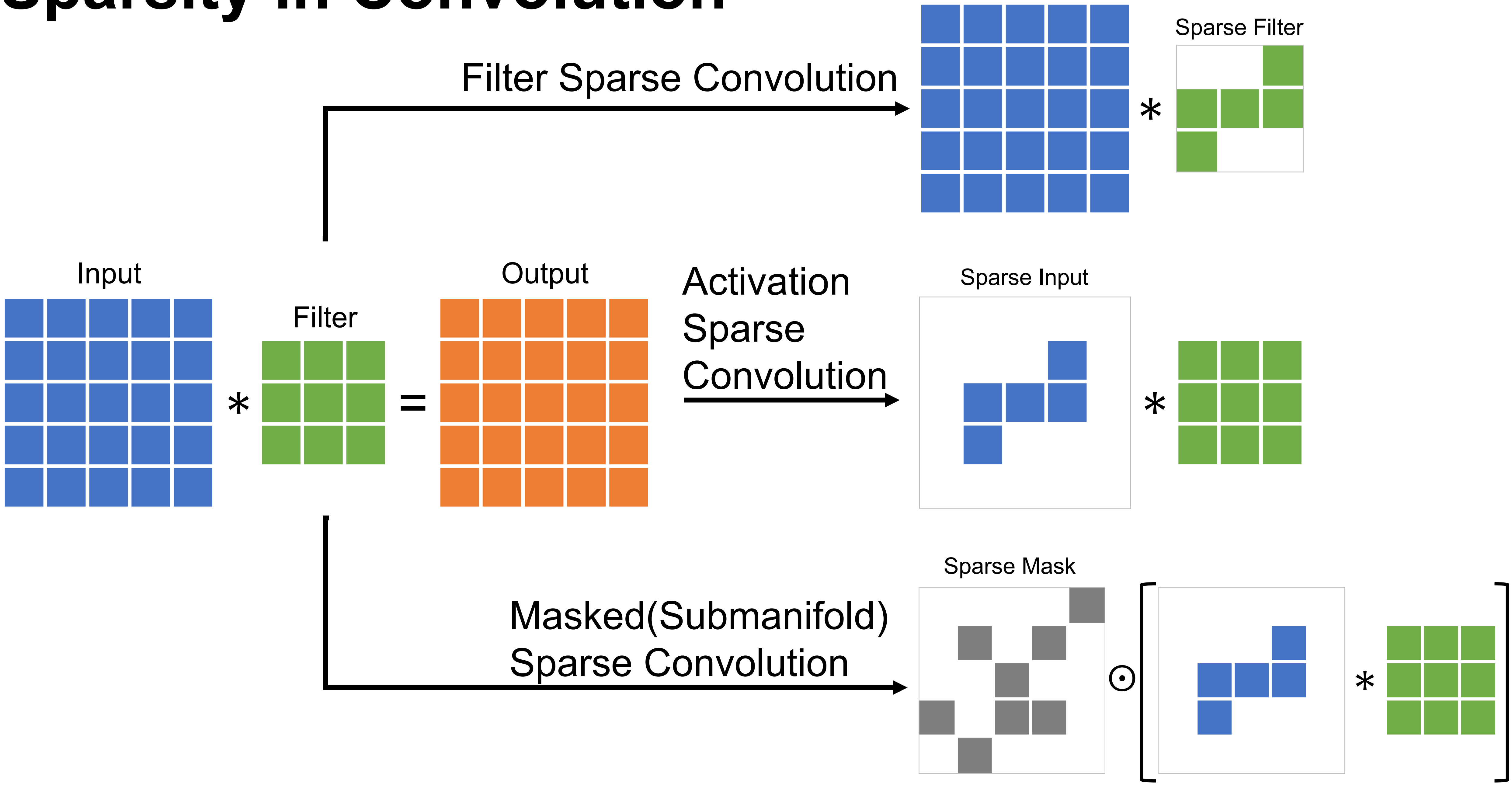
Sparsity in Convolution



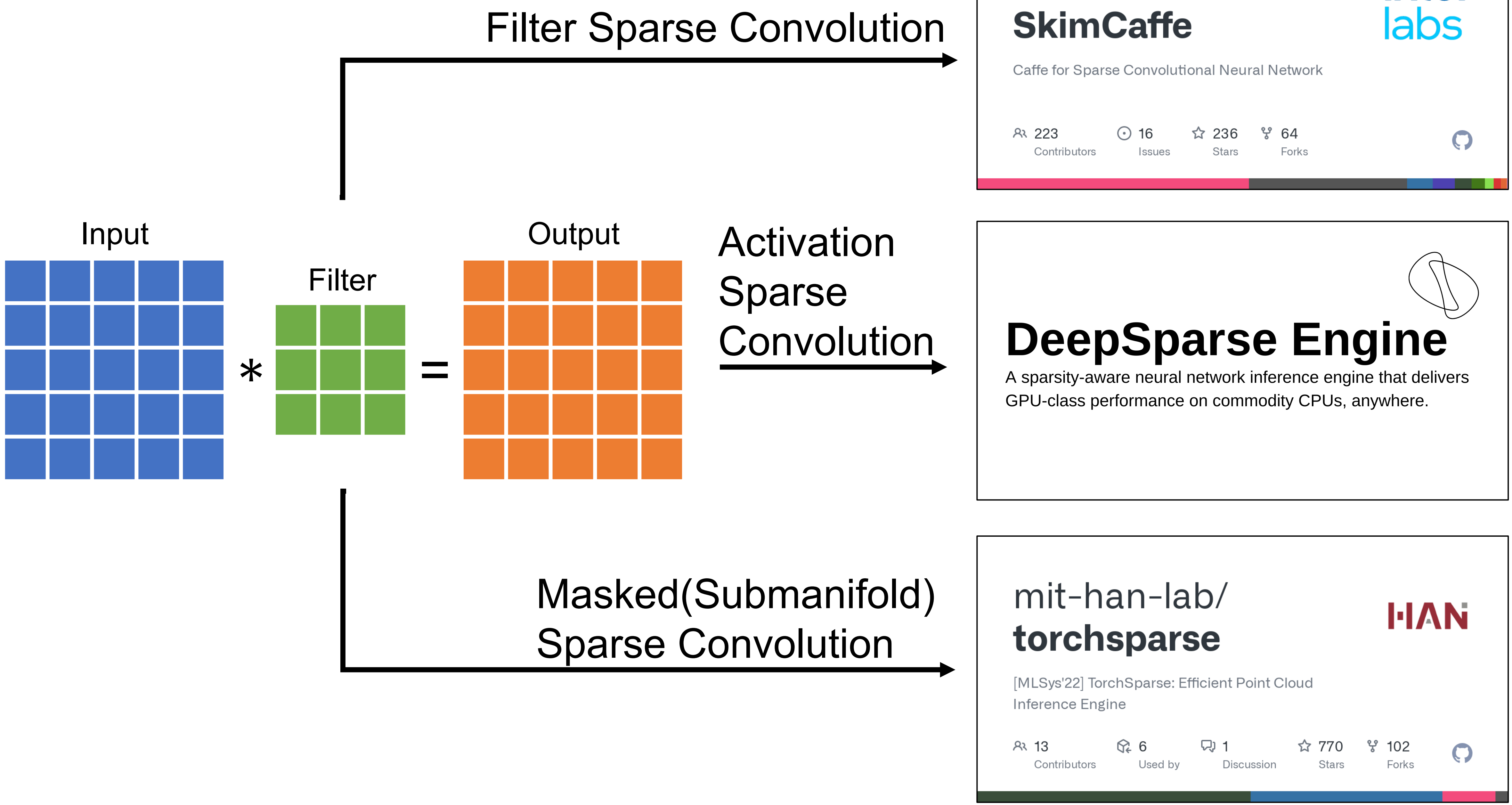
Sparsity in Convolution



Sparsity in Convolution



Sparsity in Convolution



Limitation of Library-based Approach

Name	Sparse Convolutions				Formats	Backends	
	Filter SpConv	Activation SpConv	Submanifold SpConv	Dual SpConv		CPU	GPU
SkimCaffe	✓	✗	✗	✗	1	✓	✗
TorchSparse	✗	✗	✓	✗	1	✓	✓
DeepSparse	✓	✓	✗	✗	1	✓	✗

1. Unoptimized for new sparse convolutions
2. Unoptimized for various formats and backends

Limitation of Library-based Approach

Name	Sparse Convolutions				Formats	Backends	
	Filter SpConv	Activation SpConv	Submanifold SpConv	Dual SpConv		CPU	GPU
SkimCaffe	✓	✗	✗	✗	1	✓	✗
TorchSparse	✗	✗	✓	✗	1	✓	✓
DeepSparse	✓	✓	✗	✗	1	✓	✗

1. Unoptimized for new sparse convolutions
2. Unoptimized for various formats and backends

Limitation of Library-based Approach

Name	Sparse Convolutions				Formats	Backends	
	Filter SpConv	Activation SpConv	Submanifold SpConv	Dual SpConv		CPU	GPU
SkimCaffe	✓	✗	✗	✗	1	✓	✗
TorchSparse	✗	✗	✓	✗	1	✓	✓
DeepSparse	✓	✓	✗	✗	1	✓	✗

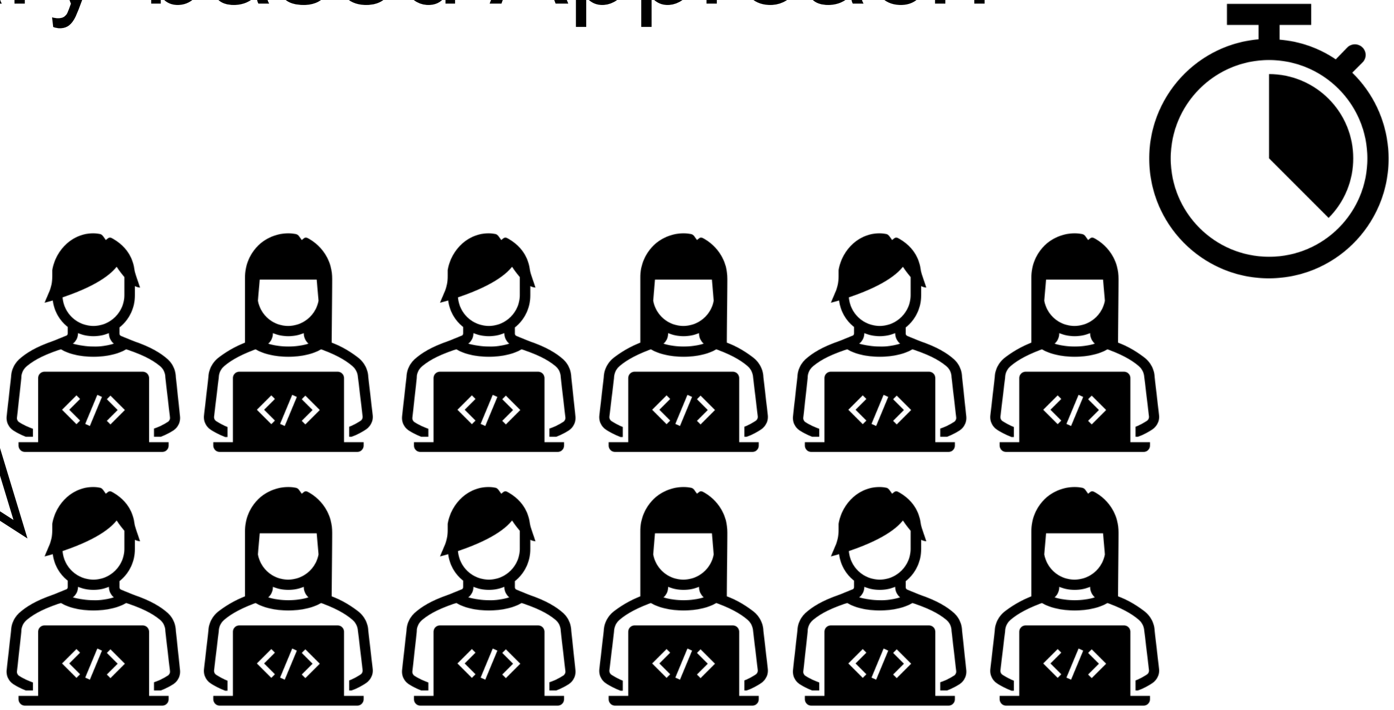
1. Unoptimized for new sparse convolutions
2. Unoptimized for various formats and backends

Limitation of Library-based Approach

<TODO list>

1. Optimize on Edge Device
2. Optimize on new GPU
3. Add New Features

…



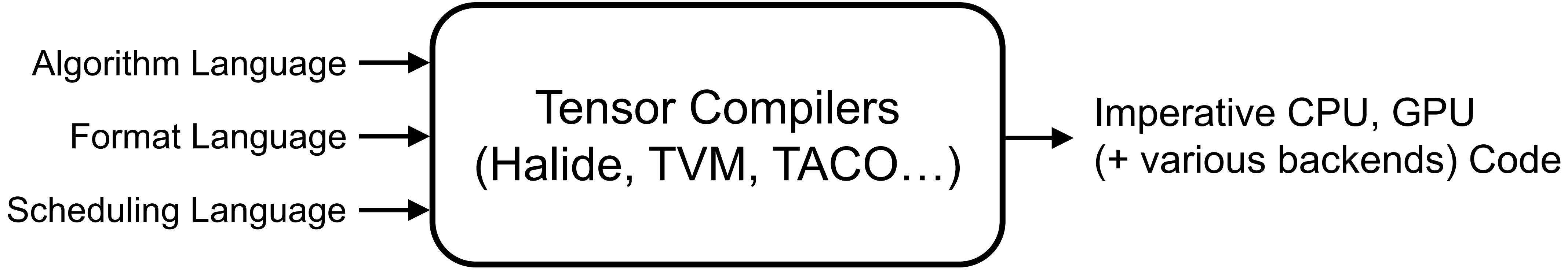
1. Unoptimized for new sparse convolutions
2. Unoptimized for various formats and backends

Unified Convolution Framework (UCF)

Name	Sparse Convolutions				Formats	Backends	
	Filter SpConv	Activation SpConv	Submanifold SpConv	Dual SpConv		CPU	GPU
SkimCaffe	✓	✗	✗	✗	1	✓	✗
TorchSparse	✗	✗	✓	✗	1	✓	✓
DeepSparse	✓	✓	✗	✗	1	✓	✗
Our Work (TACO-UCF)	✓	✓	✓	✓	> 100	✓	✓

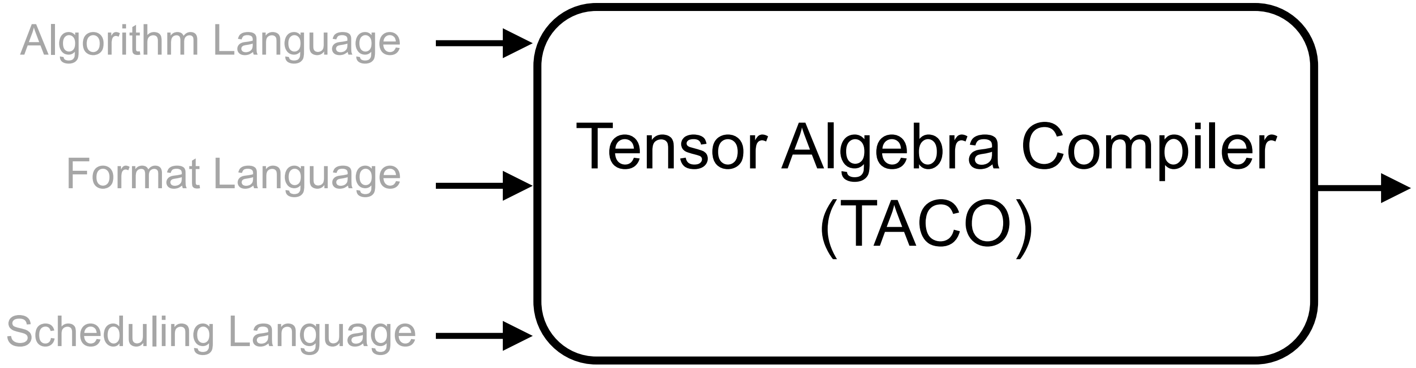
Unified Convolution Framework is a compiler that supports
(1) all sparse convolutions
(2) on various formats and backends

Background : Tensor Compiler



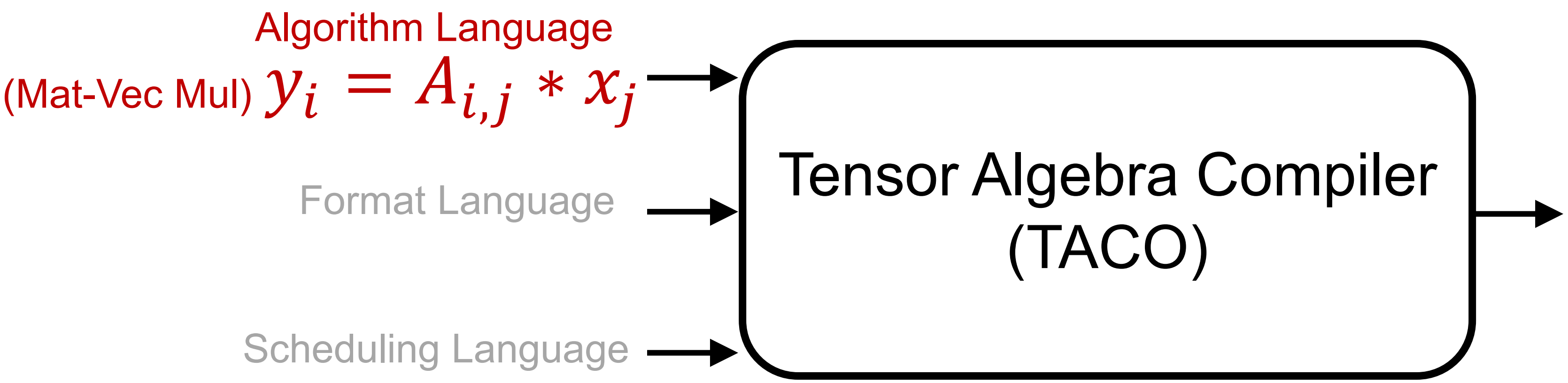
Background : Tensor Compiler (TACO)

What we want : Sparse Matrix – Dense Vector Multiplication Kernel

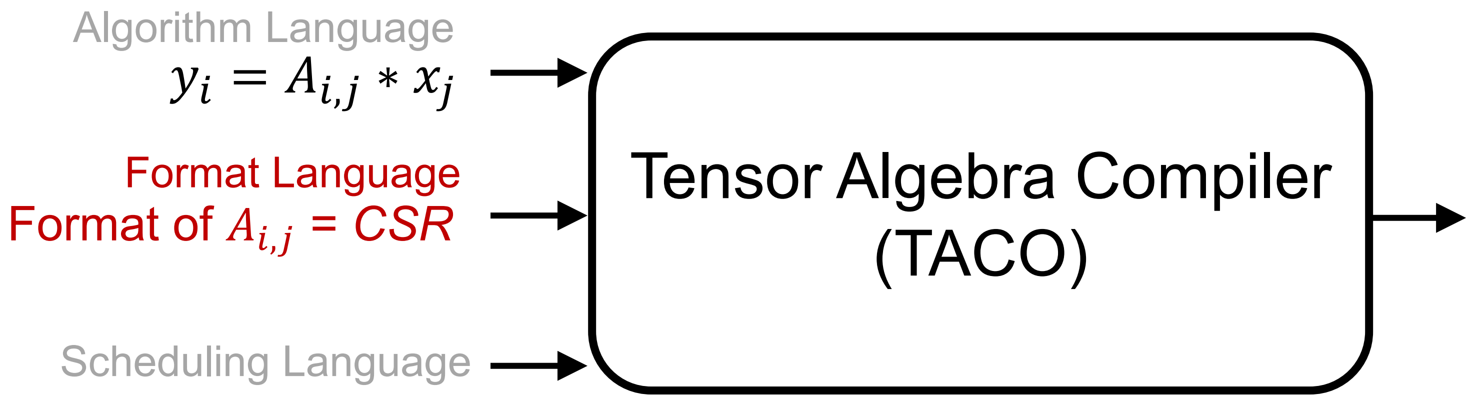


Background : Tensor Compiler (TACO)

What we want : Sparse Matrix – Dense Vector Multiplication Kernel



Background : Tensor Compiler (TACO)



Background : Tensor Compiler (TACO)

Algorithm Language

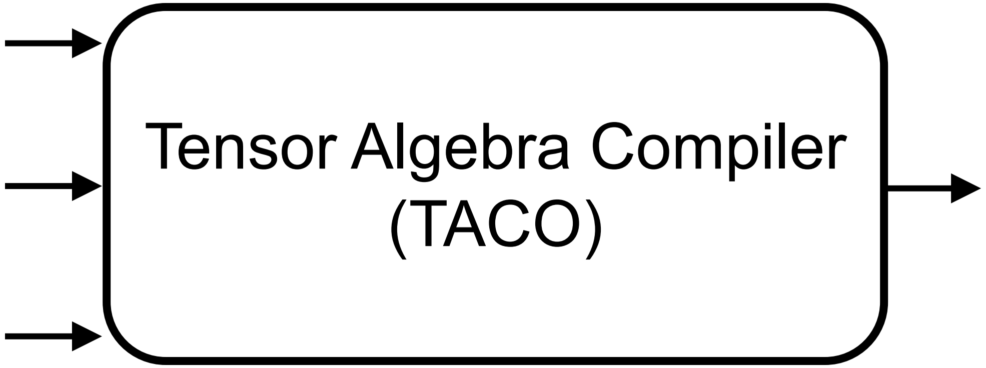
$y_i = A_{i,j} * x_j$

Format Language

Format of $A_{i,j}$ = *CSR*

Scheduling Language

```
.split(i,i0,i1,32)
.reorder(i0,i1,j)
.parallelize(i0)
```



Background : Tensor Compiler (TACO)

Algorithm Language

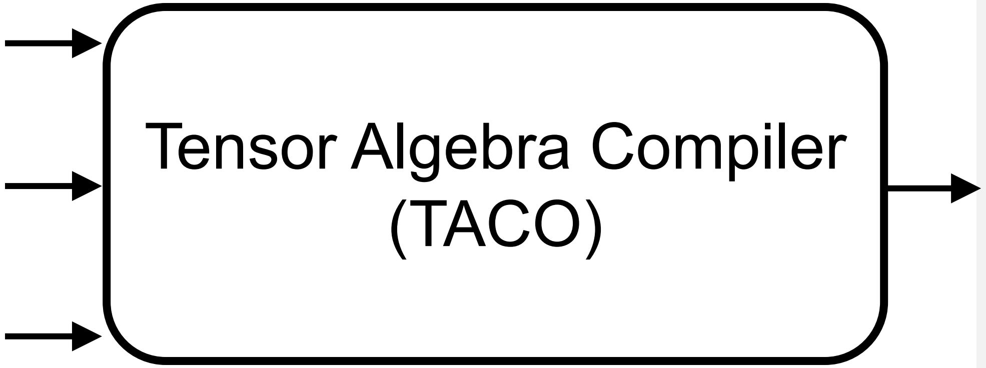
$y_i = A_{i,j} * x_j$

Format Language

Format of $A_{i,j}$ = *CSR*

Scheduling Language

```
.split(i,i0,i1,32)
.reorder(i0,i1,j)
.parallelize(i0)
```



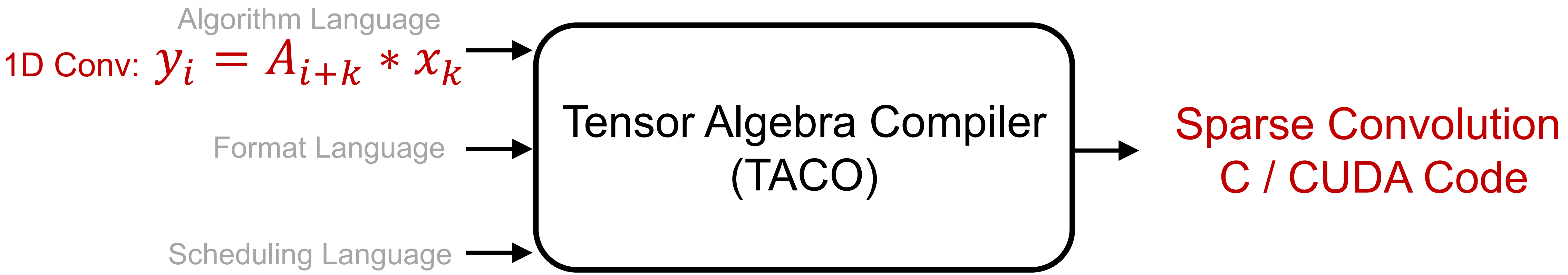
Sparse Matrix – Vector Multiply

```
#pragma omp parallel for
for (int32_t i0=0;
     i0<(A1_dimension/32); i0++)
{
  for (int32_t i1=0; i1<32; i1++)
  {
    int32_t i = i0*32 + i1;
          ⋮
```

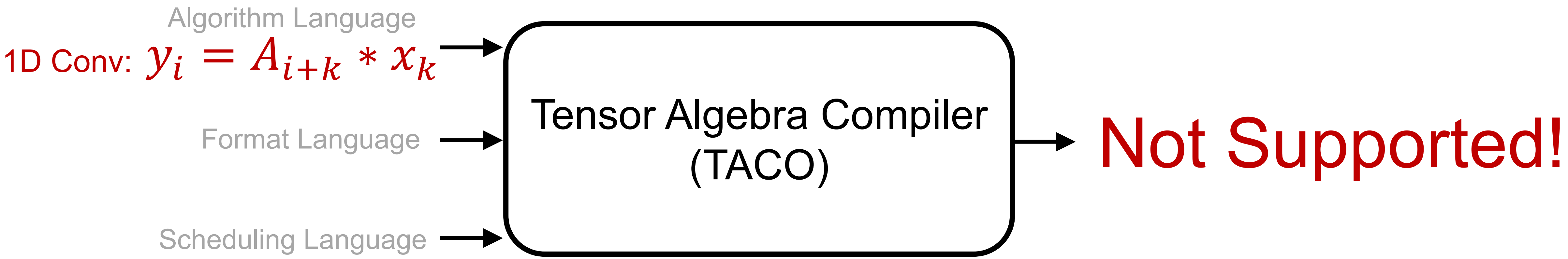
Generated C/CUDA Code

Background : Tensor Compiler (TACO)

What we want : Sparse Convolution Kernel

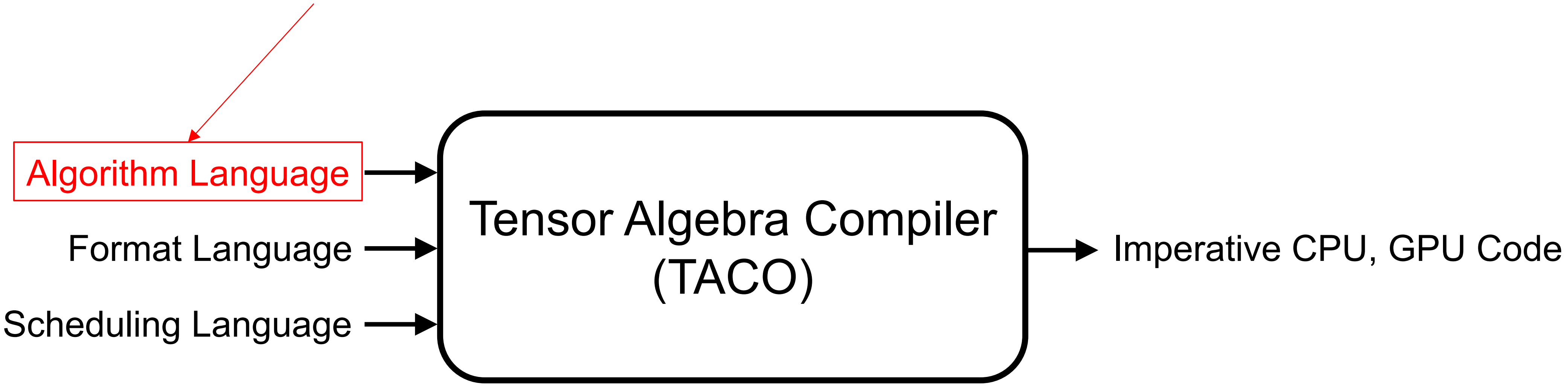


Background : Tensor Compiler (TACO)

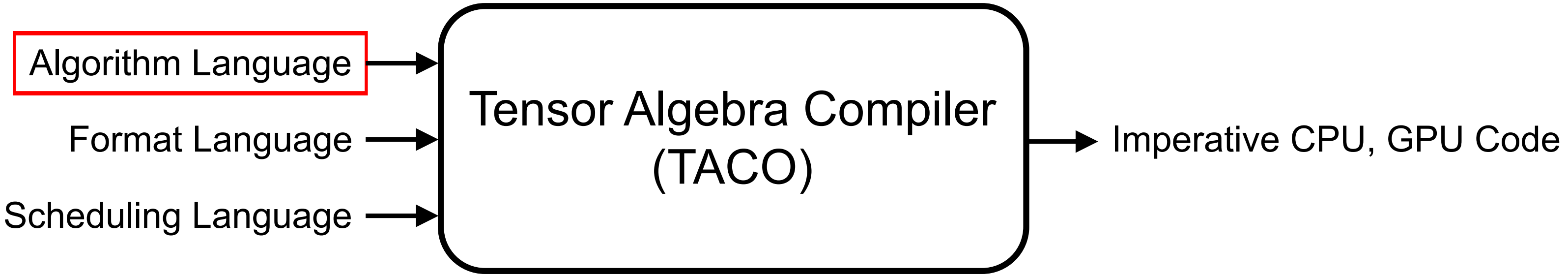


Why can’t TACO support convolutions?

TACO’s Algorithm Language does not accept ”Affine Index”



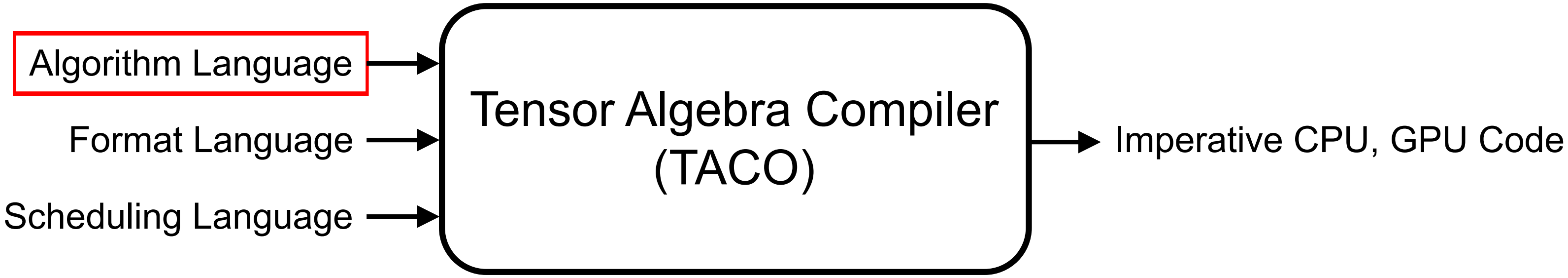
Why can’t TACO support convolutions?



✅ What TACO can support

❌ What TACO cannot support

Why can't TACO support convolutions?



What TACO can support

$$C_i = A_i * B_i$$ (Element-wise Mul)

$$C_i = A_{2i+1} * B_{3i-1}$$

$$C_{i,j} = A_{i,k} * B_{j,k}$$ (MatMul)

Single Variable Affine Expression (SVAE)

What TACO cannot support

Why can't TACO support convolutions?

Algorithm Language →
Format Language →
Scheduling Language →

Tensor Algebra Compiler (TACO) → Imperative CPU, GPU Code

✅ What TACO can support

$C_i = A_i * B_i$ (Element-wise Mul)

$C_i = A_{2i+1} * B_{3i-1}$

$C_{i,j} = A_{i,k} * B_{j,k}$ (MatMul)

Single Variable Affine Expression (SVAE)

❌ What TACO cannot support

$C_i = A_{i+k} * B_k$ (1DConv)

multiple variables (i and k)

Multiple Variable Affine Expression (MVAE)

Lowering Technique

Single Variable Affine Expression (SVAE)

↓

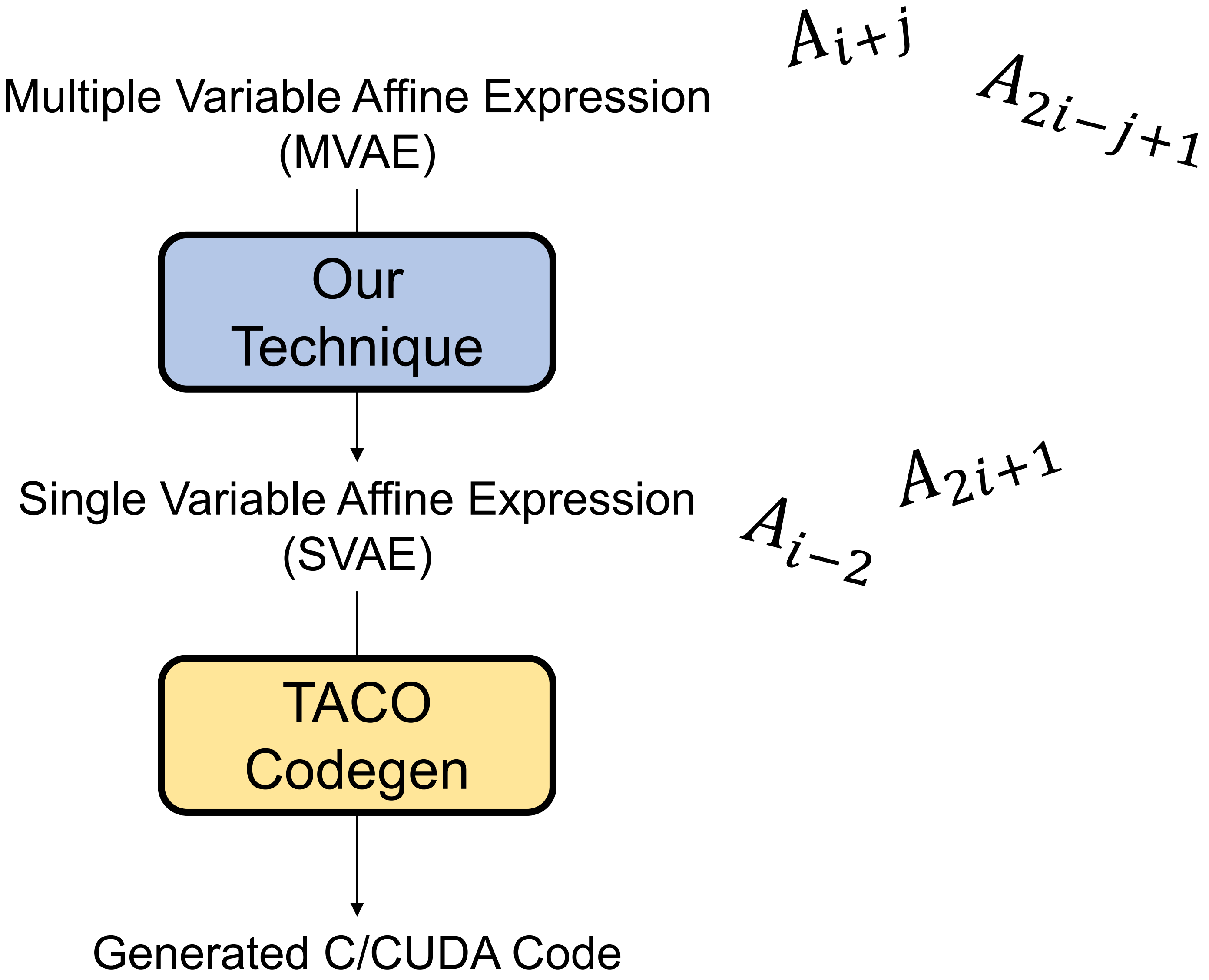
TACO Codegen

↓

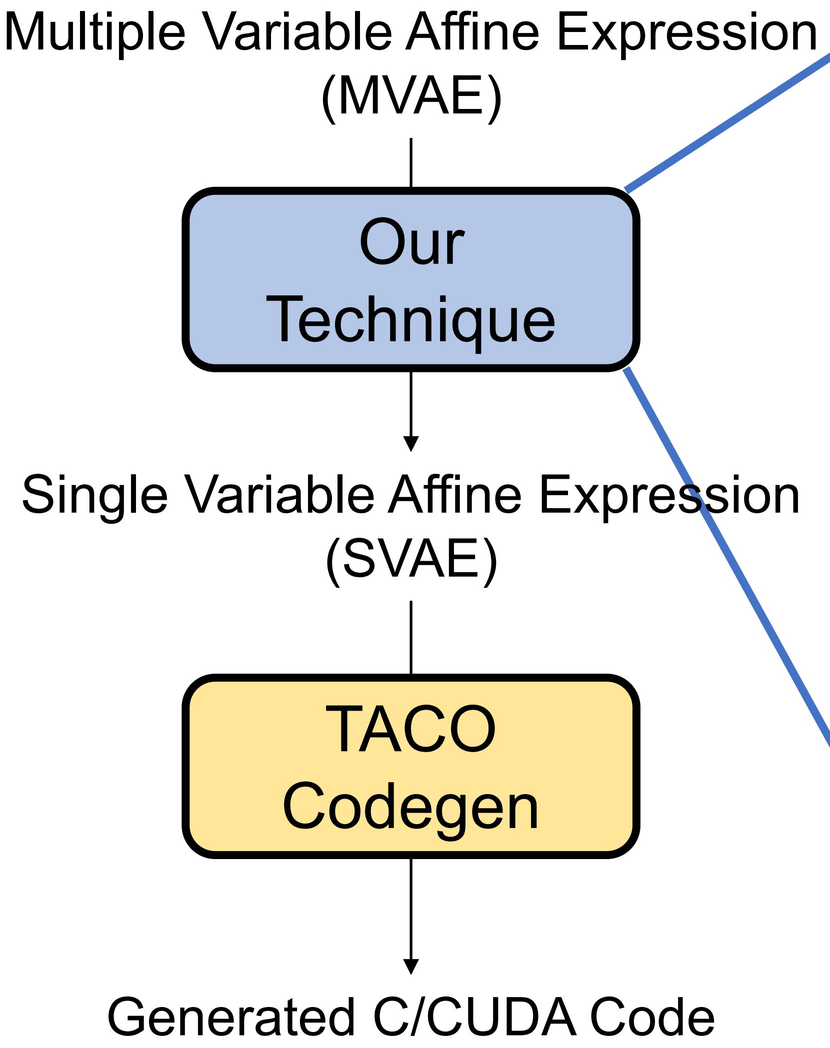
Generated C/CUDA Code

A_{i-2} A_{2i+1}

Lowering Technique



Lowering Technique

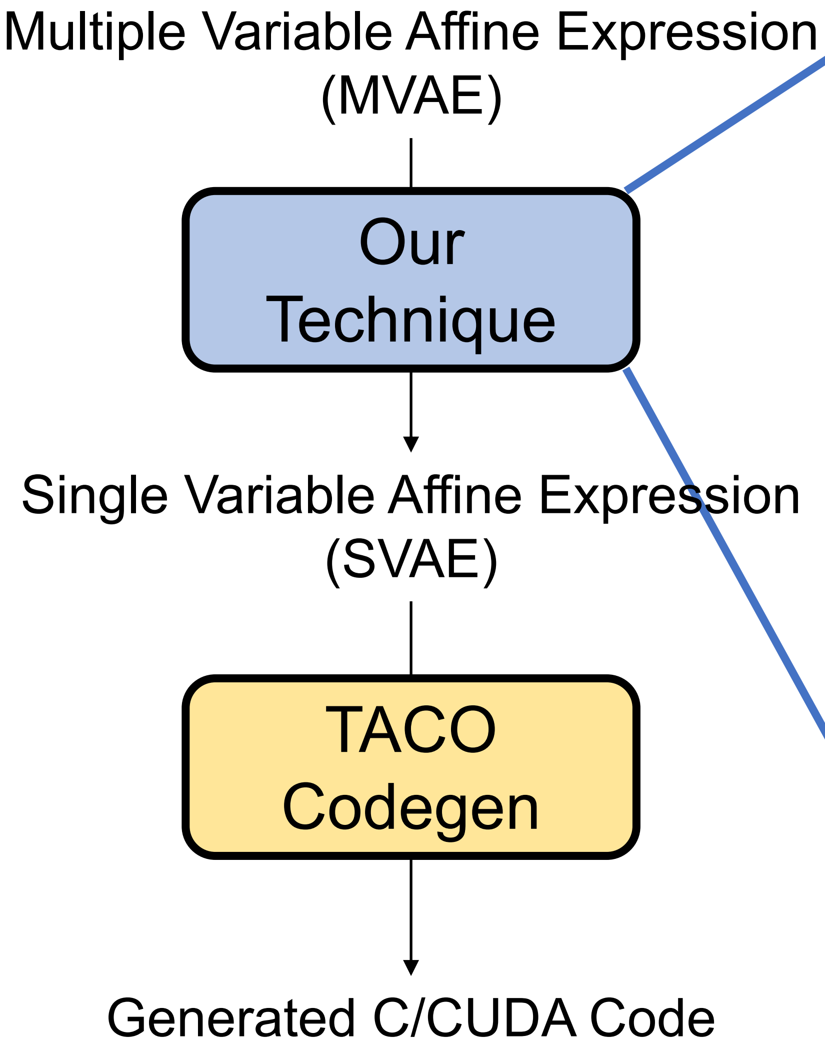


MVAE = Stride * Base-variable + Offset

$$A_{3i+2j+5} = A_{2*j+(3i+5)}$$

Stride   Base-variable   Offset

Lowering Technique



MVAE = Stride * Base-variable + Offset

$$A_{3i+2j+5} = A_{2*j+(3i+5)}$$

Stride — Base-variable — Offset

```
For i:
  offset = 3*i+5
  For j: //Base-variable j
    access A[2*j + offset]
```

Lowering Technique

Multiple Variable Affine Expression (MVAE)

Our Technique

Single Variable Affine Expression (SVAE)

TACO Codegen

Generated C/CUDA Code

MVAE = Stride * Base-variable + Offset

$$A_{3i+2j+5} = A_{2*j+(3i+5)}$$

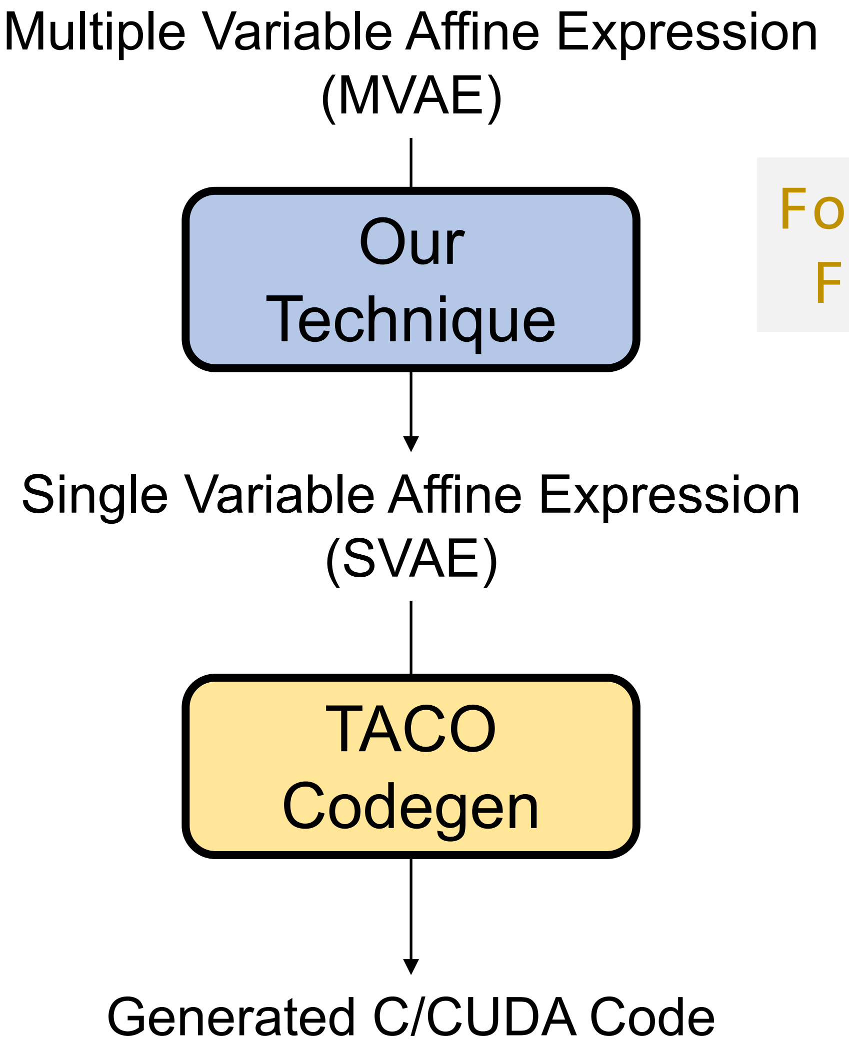
Stride: 2, Base-variable: j, Offset: $(3i+5)$

||

$$A_{3*i+(2j+5)}$$

Stride: 3, Base-variable: i, Offset: $(2j+5)$

Lowering Technique



MVAE

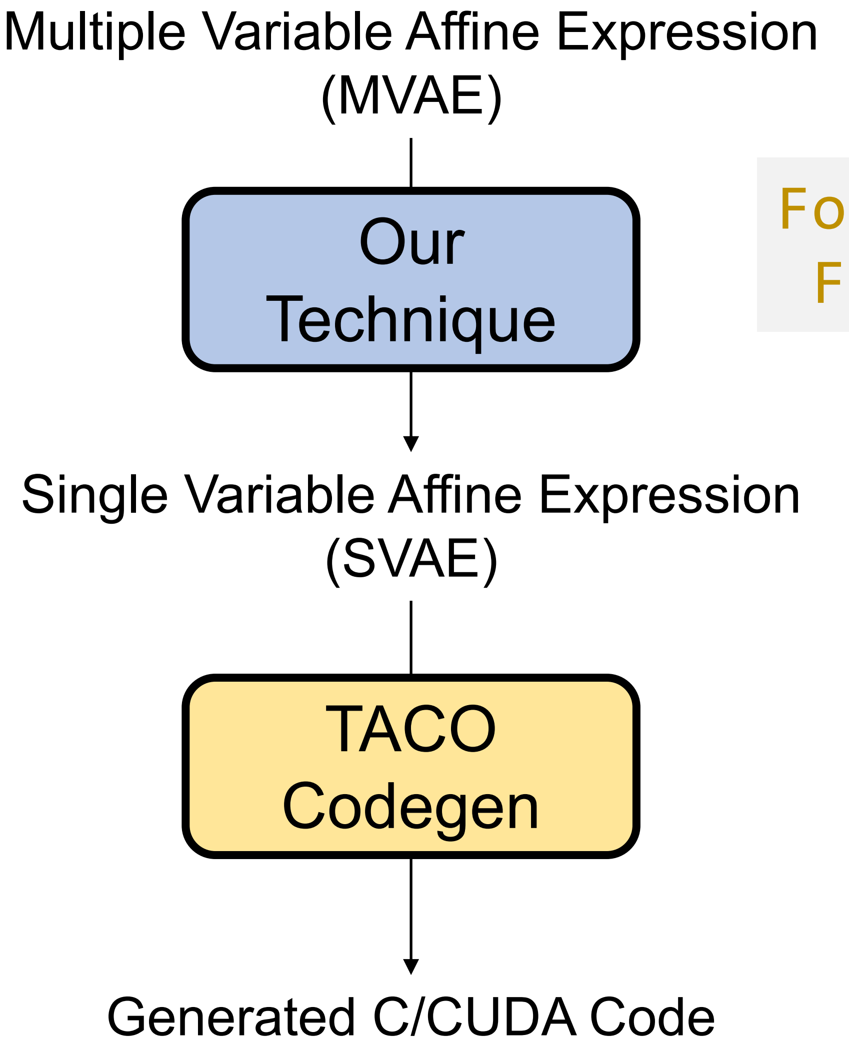
$$3i + 2j + 5$$

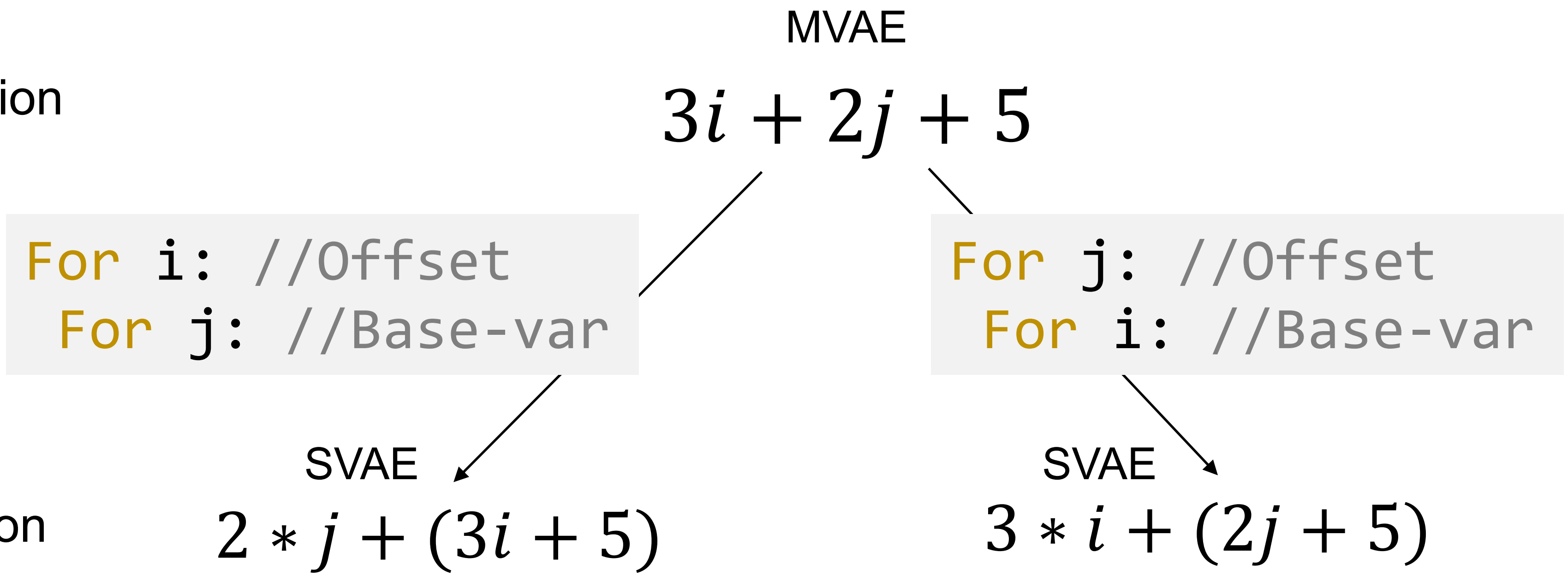
```
For i: //Offset
 For j: //Base-var
```

SVAE

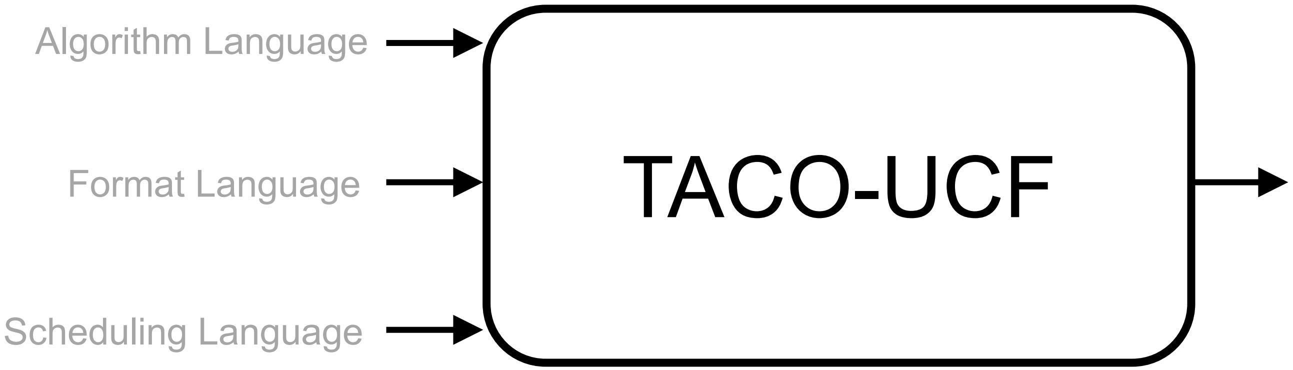
$$2 * j + (3i + 5)$$

Lowering Technique





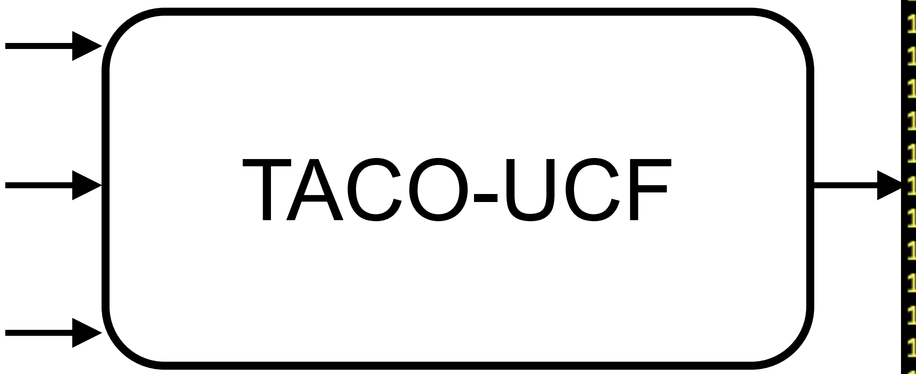
Sparse Convolution w/ UCF

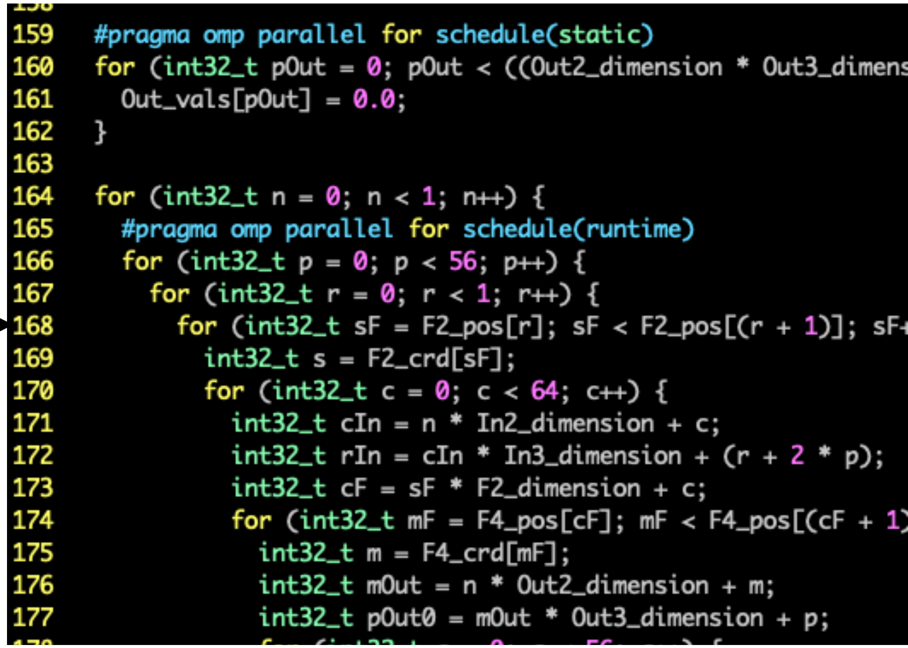


Sparse Convolution w/ UCF

2D Conv

$$O_{p,q,m} = I_{p+r,q+s,c} * F_{m,r,s,c}$$





Sparse CPU/GPU Convolution Kernel

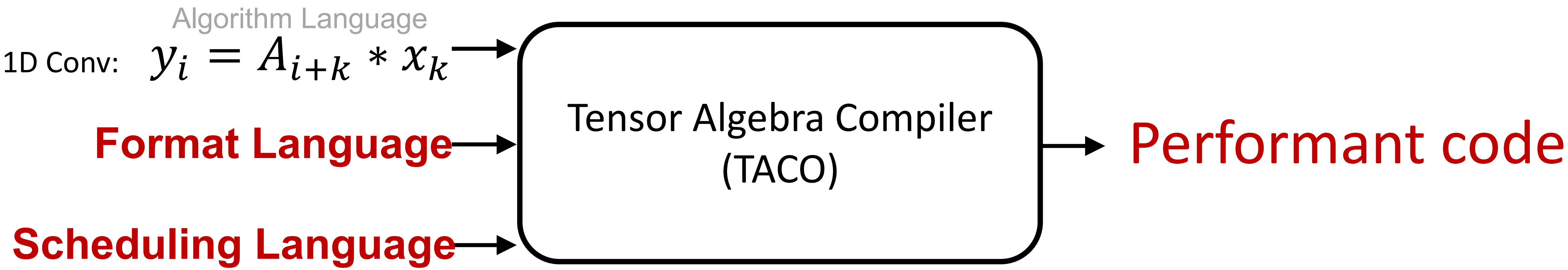
Evaluation

CPU : Intel Xeon E5-2680 v3 (24 threads)
GPU : Nvidia V100

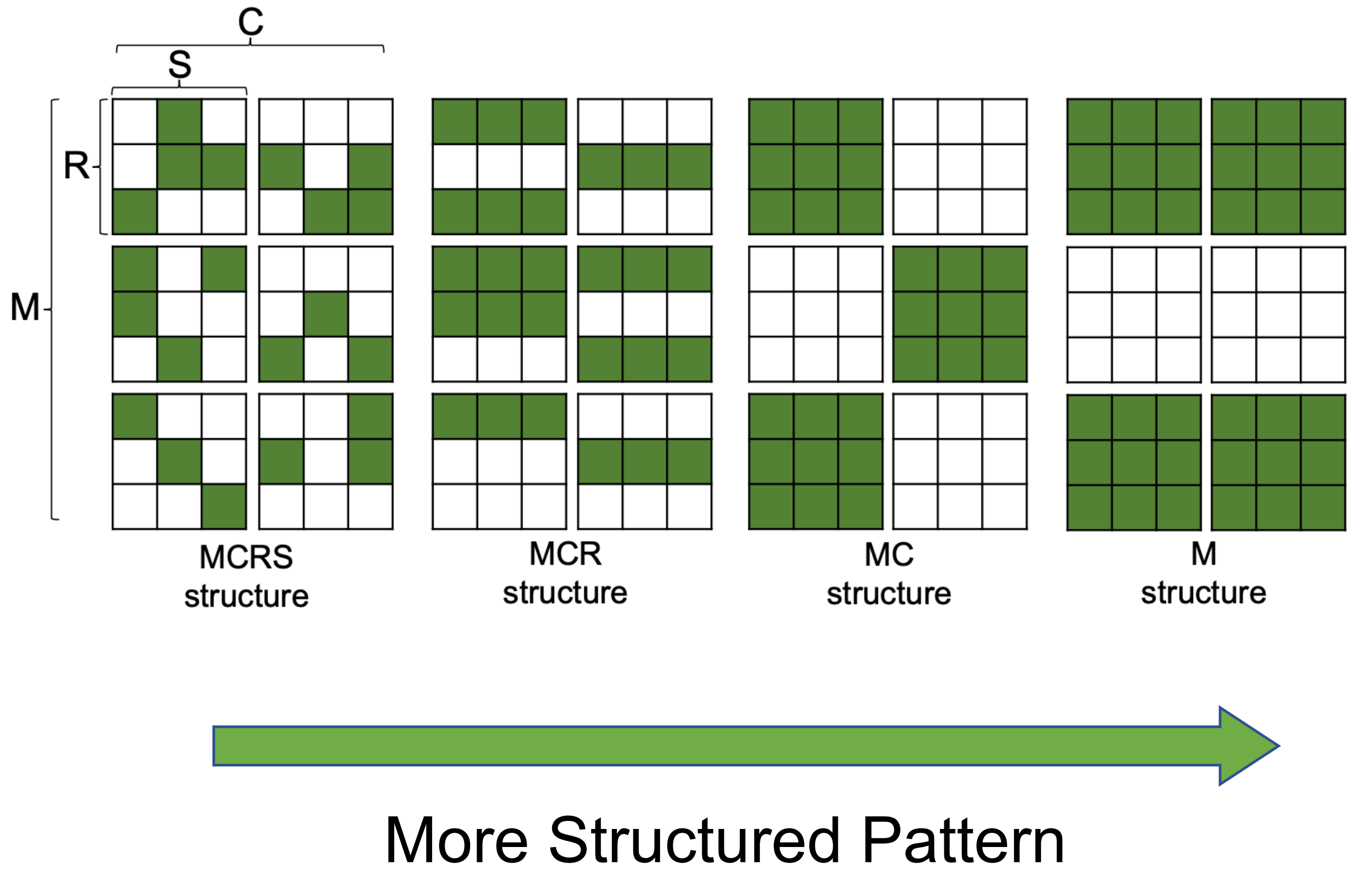
1. Importance of Format and Schedule.

2. Performance Comparison
   - Filter Sparse Convolution
   - Submanifold Sparse Convolution

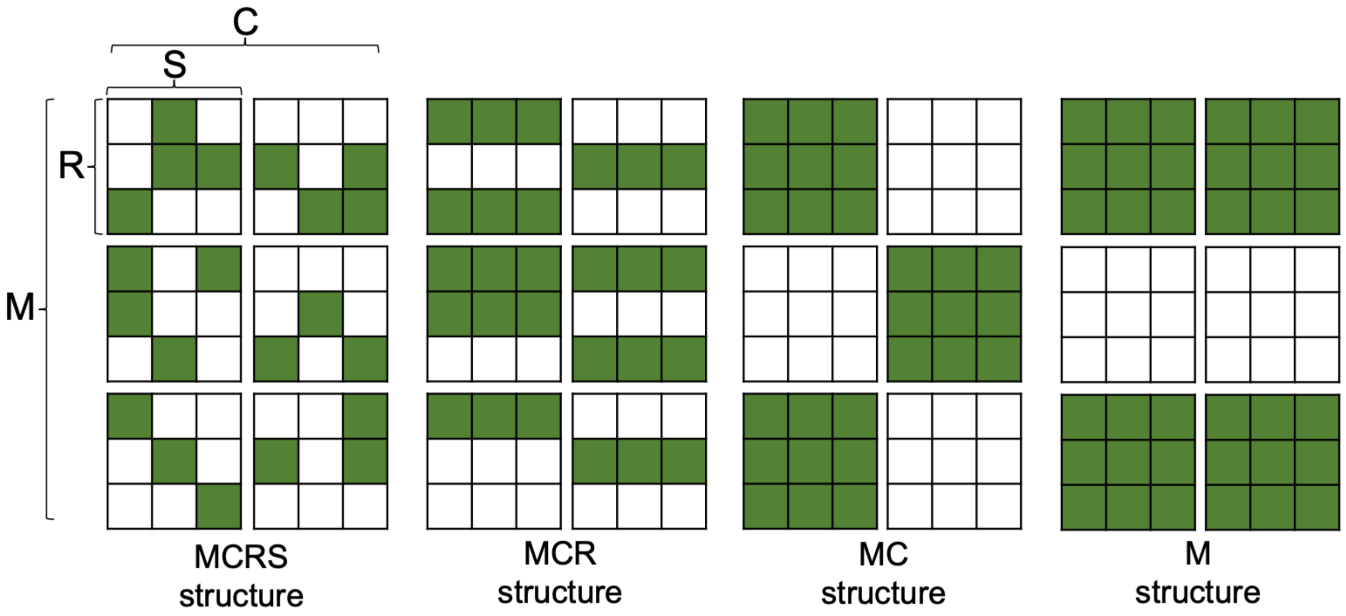
Format and Schedule Matters.



Format and Schedule Matters.



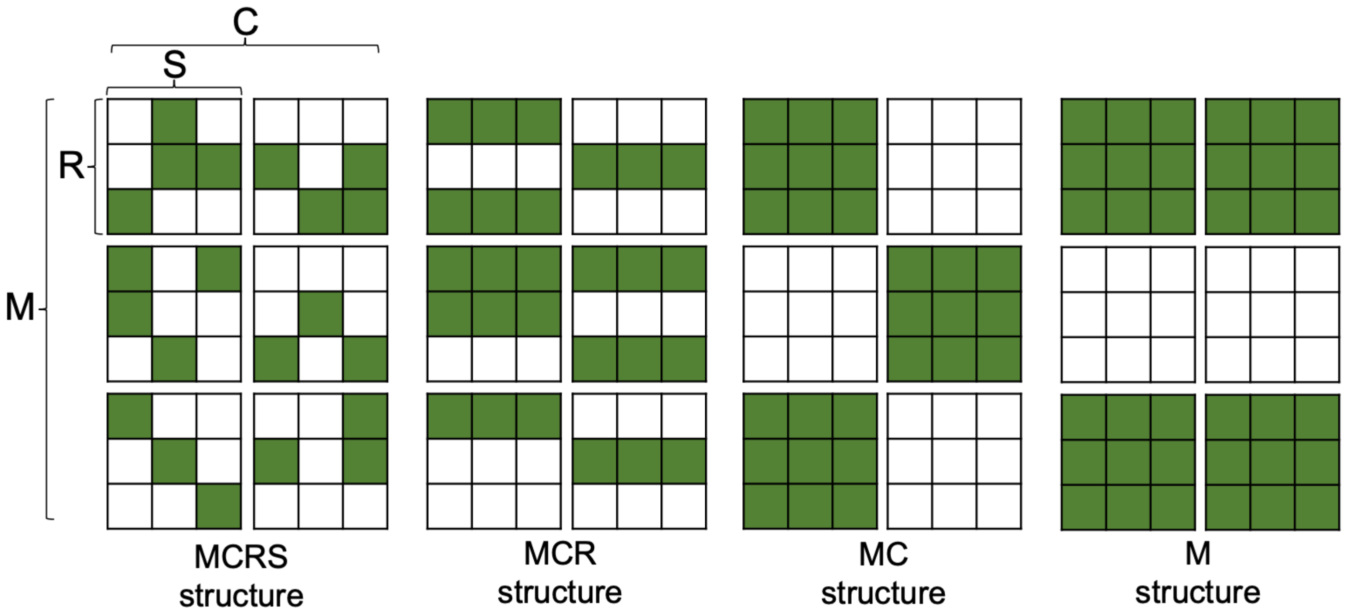
Format and Schedule Matters.



Memory Saving over Uncompressed(Dense) Representation

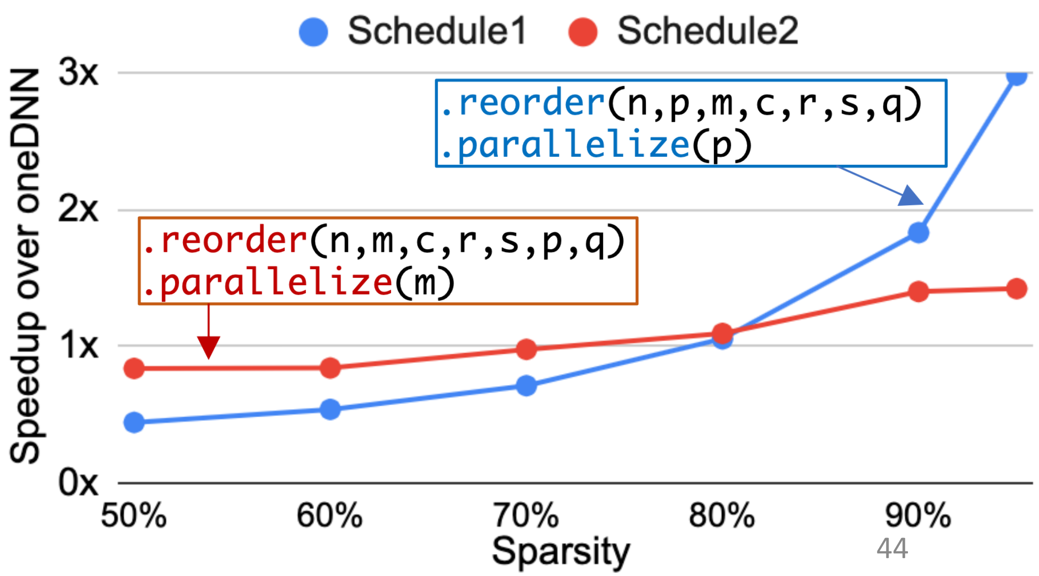
80% Sparsity	MCRS Structure (Unstructured)	MCR Structure	MC Structure	M Structure
Format1	1.08x	1.55x	1.72x	1.84x
Format2	**1.17x**	**2.64x**	3.17x	3.41x
Format3	1.03x	1.83x	**4.42x**	4.78x
Format4	0.99x	0.99x	0.99x	**5x**

Format and Schedule Matters.



Memory Saving over Uncompressed(Dense) Representation

80% Sparsity	**MCRS Structure (Unstructured)**	**MCR Structure**	**MC Structure**	**M Structure**
Format1	1.08x	1.55x	1.72x	1.84x
Format2	**1.17x**	**2.64x**	3.17x	3.41x
Format3	1.03x	1.83x	**4.42x**	4.78x
Format4	0.99x	0.99x	0.99x	**5x**



Evaluation

CPU : Intel Xeon E5-2680 v3 (24 threads)
GPU : Nvidia V100

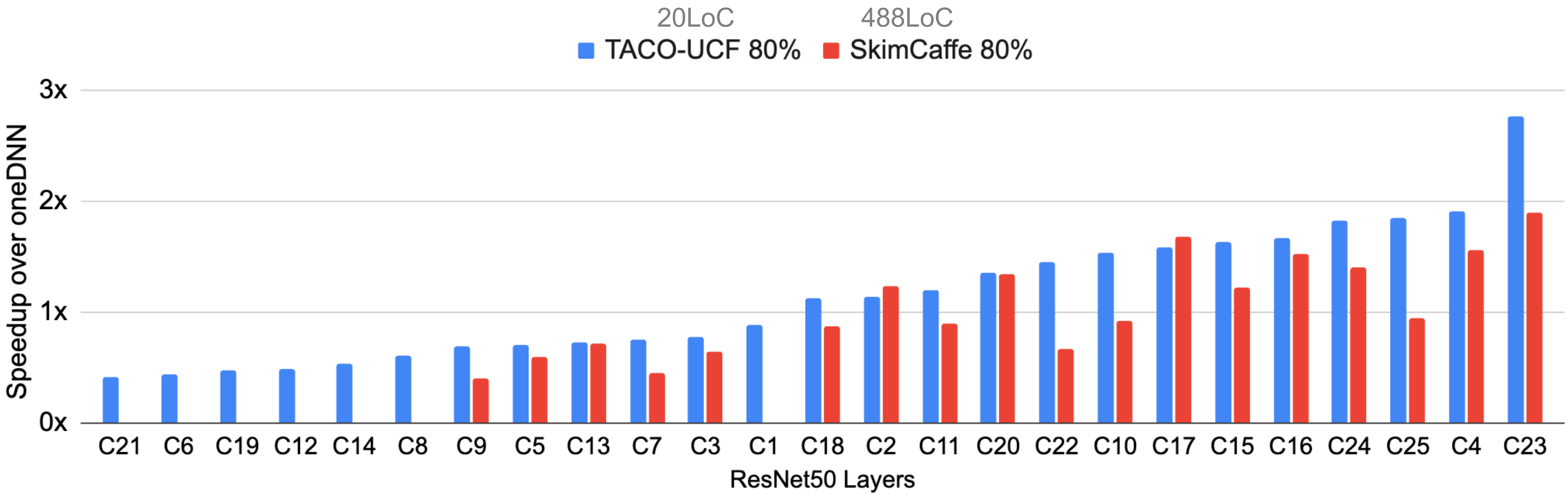
1. Importance of Format and Schedule.

2. Performance Comparison

    - **Filter Sparse Convolution**
    - Submanifold Sparse Convolution

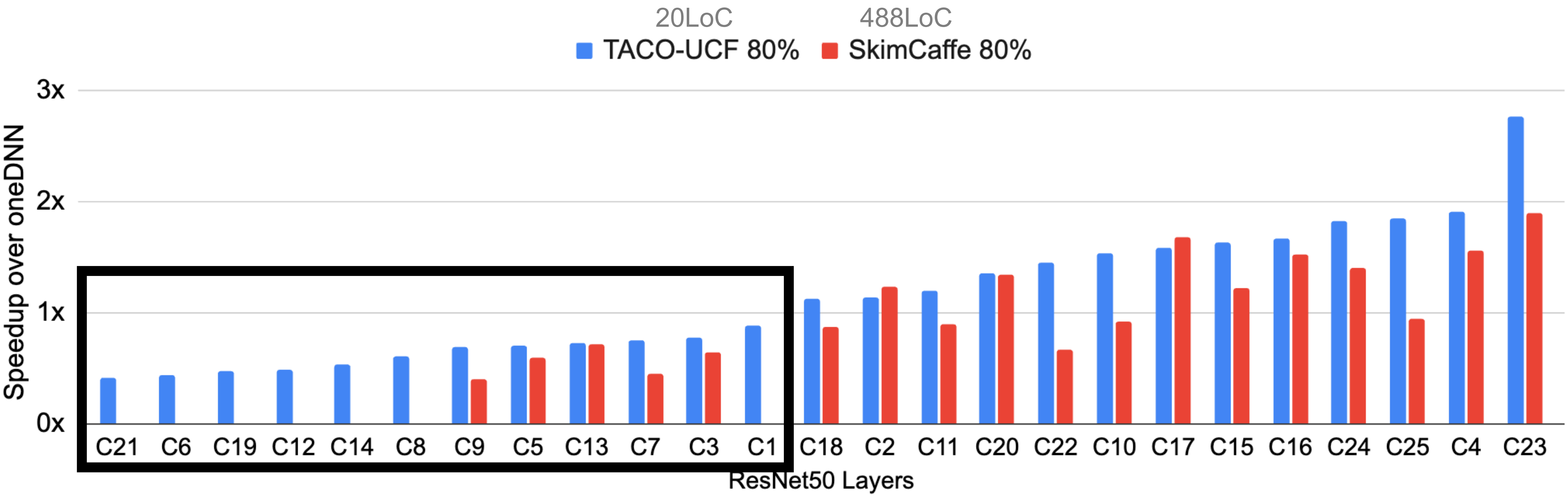
Evaluation – Filter Sparse Convolution

80% sparsity - pruned ResNet50
CPU : 24-core Intel Xeon



Evaluation – Filter Sparse Convolution

80% sparsity - pruned ResNet50
CPU : 24-core Intel Xeon



Not all layers can benefit from pruning!

Evaluation – Filter Sparse Convolution

ResNet50 on Nvidia V100 GPU

Pruning Sparsity	80%	91%	96%	98%
cuDNN	1.0×	1.0×	1.0×	1.0×
Escort	0.78×	1.09×	1.35×	1.49×
TACO-UCF	**1.08**×	**1.61**×	**2.15**×	**2.57**×

TACO-UCF > cuDNN at 80% Sparsity

Escort > cuDNN at 91% Sparsity

Evaluation

CPU : Intel Xeon E5-2680 v3 (24 threads)
GPU : Nvidia V100

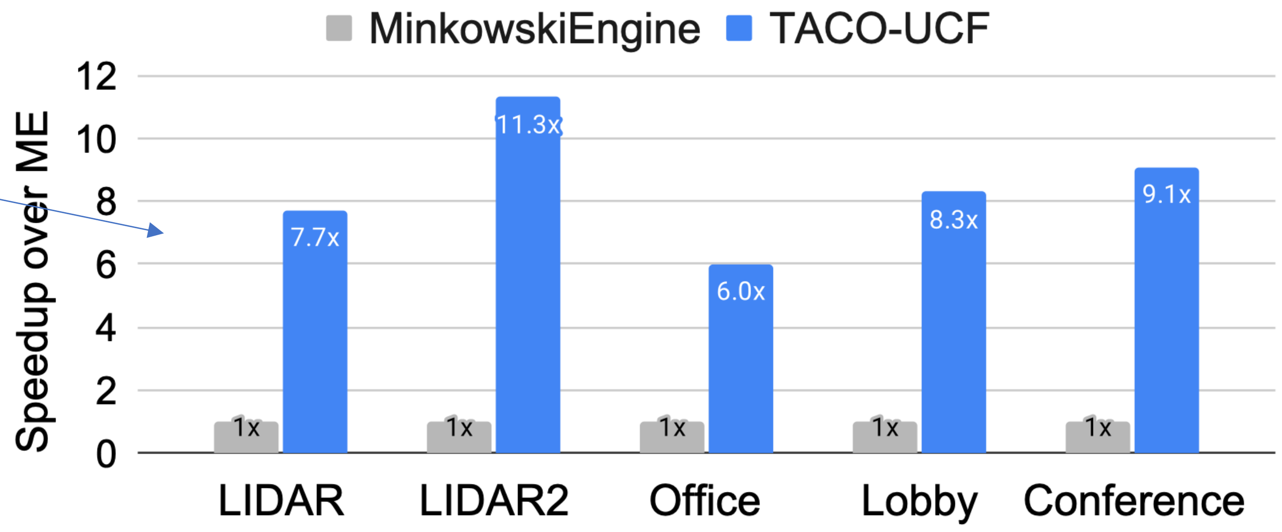
1. Importance of Format and Schedule.

2. Performance Comparison

- Filter Sparse Convolution
- **Submanifold Sparse Convolution**

Evaluation – Submanifold Sparse Convolution

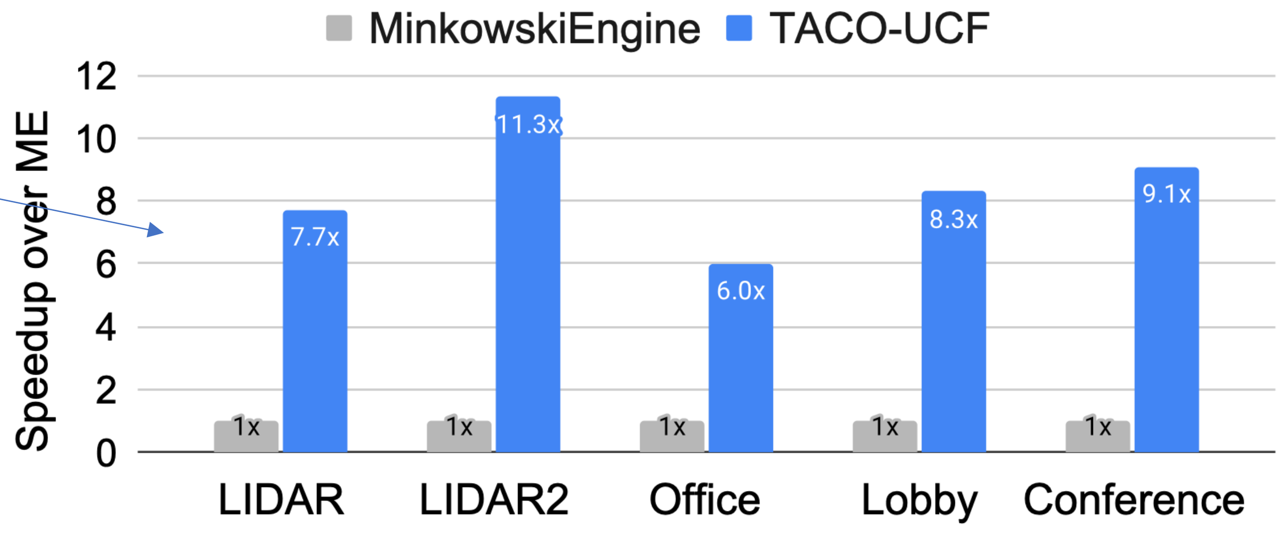
6x – 11x Speedup on CPU



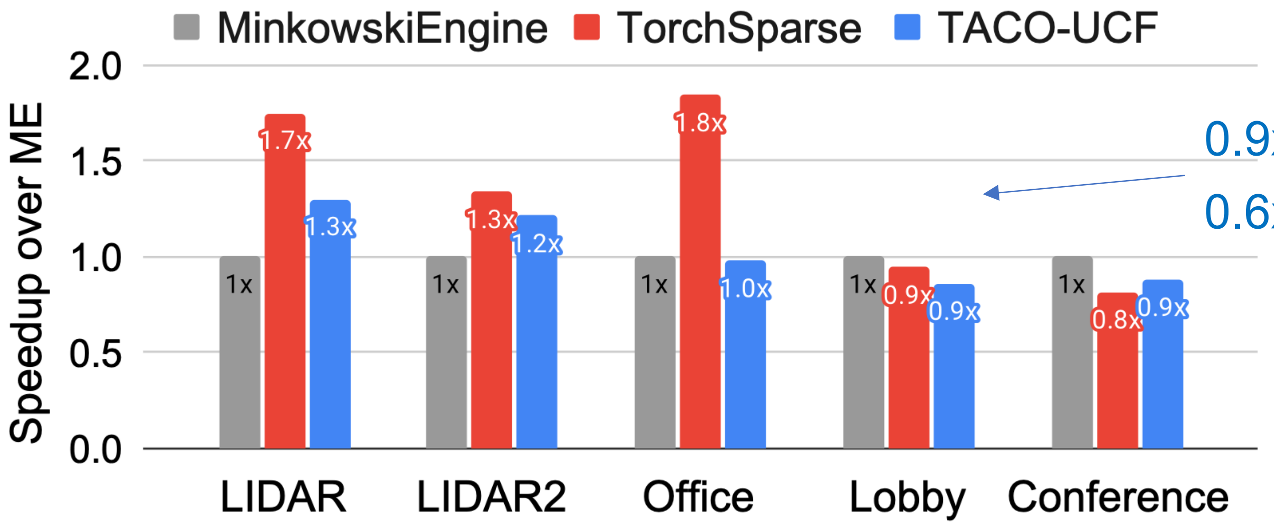
(a) CPU Result

Evaluation – Submanifold Sparse Convolution

6x – 11x Speedup on CPU



(a) CPU Result



(b) GPU Result

0.9x – 1.3x over MinkowskiEngine

0.6x – 1.1x over TorchSparse

Evaluation

CPU : Intel Xeon E5-2680 v3 (24 threads)
GPU : Nvidia V100

1. Importance of Format and Schedule.

2. Performance Comparison

Library Name	Filter Sparse Conv		Submanifold Sparse Conv	
	CPU	GPU	CPU	GPU
SkimCaffe				
Escort				
MinkowskiEngine				
TorchSparse				
Ours (Normalized)				

Evaluation

CPU : Intel Xeon E5-2680 v3 (24 threads)
GPU : Nvidia V100

1. Importance of Format and Schedule.

2. Performance Comparison

Library Name	Filter Sparse Conv		Submanifold Sparse Conv	
	CPU	GPU	CPU	GPU
SkimCaffe	76%	-	-	-
Escort	-	67%	-	-
MinkowskiEngine	-	-	12%	97%
TorchSparse	-	-	< 5%	123%
Ours (Normalized)	100%	100%	100%	100%

1. Better Performance
2. Versatile convolution support
3. Flexible Hardware
4. Less lines of code!

Evaluation

CPU : Intel Xeon E5-2680 v3 (24 threads)
GPU : Nvidia V100

1. Importance of Format and Schedule.

2. Performance Comparison

Library Name	**Filter Sparse Conv**		**Submanifold Sparse Conv**		**Dual Submanifold Sparse Conv**	
	CPU	**GPU**	**CPU**	**GPU**	CPU	GPU
SkimCaffe	**76%**	**-**	**-**	**-**		
Escort	**-**	**67%**	**-**	**-**		
MinkowskiEngine	**-**	**-**	**12%**	**97%**		
TorchSparse	**-**	**-**	**< 5%**	**123%**		
Ours (Normalized)	**100%**	**100%**	**100%**	**100%**		

Details
In Paper!

Thanks!