Building a Scalable AI/ML Software Stack for RISC-V

From PyTorch to Deployment on SiFive Intelligence XM Platforms

2025 RISC-V China Summit

Phoebe Chen

AIML Senior Software Engineer

AGENDA

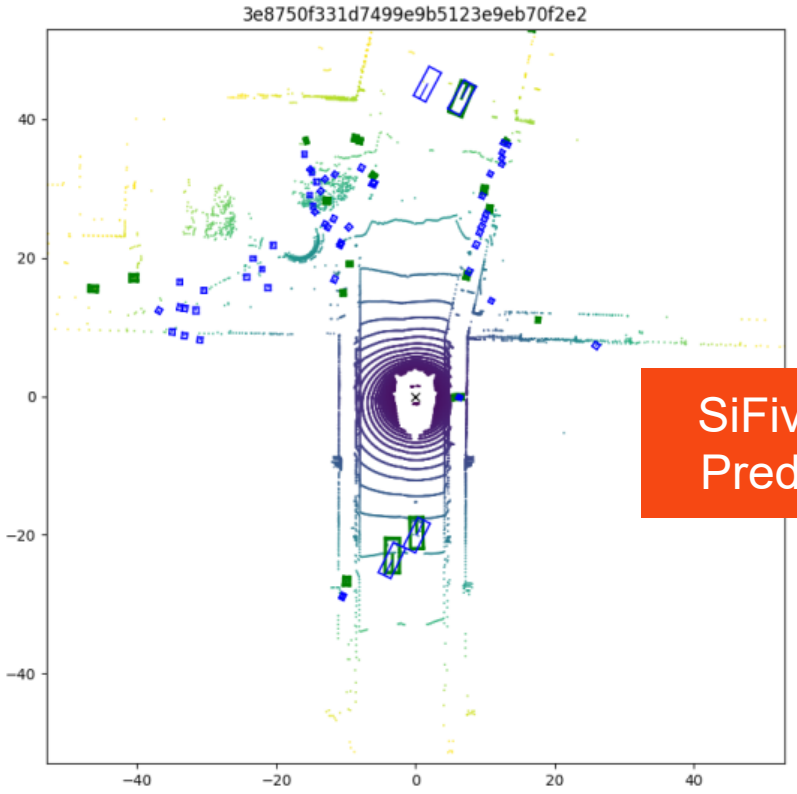
1. Showcase of Achievements
2. Introduce the SiFive AI/ML Software Stack
3. SiFive Intelligence XM Platforms
   - Integrate matmul accelerator into IREE
4. End-to-end deployment
   - Pytorch BEV perception models to RISC-V
5. Summary & Future works

SiFive

Showcase of Achievements

BEV Former Perception models on RISC-V

GPU Prediction



SiFive XM Prediction

HW Platform: SiFive XM Emulator

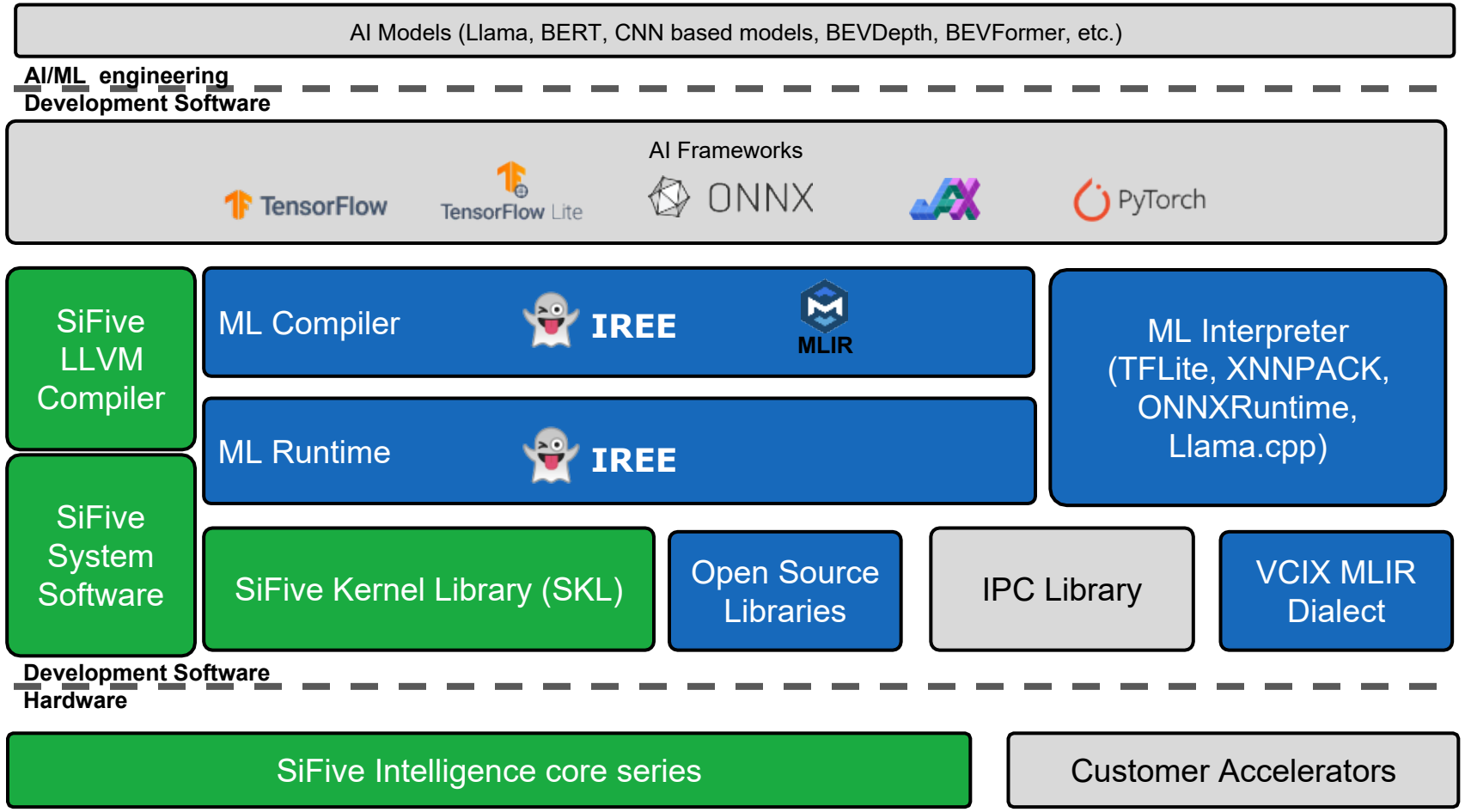
SiFive

Introduce the SiFive AI/ML Software Stack

SiFive

SiFive AI/ML Software Stack

- All critical components enabled and ready to run high profile e2e models on SiFive platforms
- **IREE** - Open-sourced MLIR compiler and runtime with SiFive uarch specific optimization
  - Utilizing highly tuned SiFive LLVM Compilers and SKL Libraries
  - Front end supports popular AI/ML models like Pytorch LLM
- Technology foundation to be extended easily by customers as needed with IREE source code provided



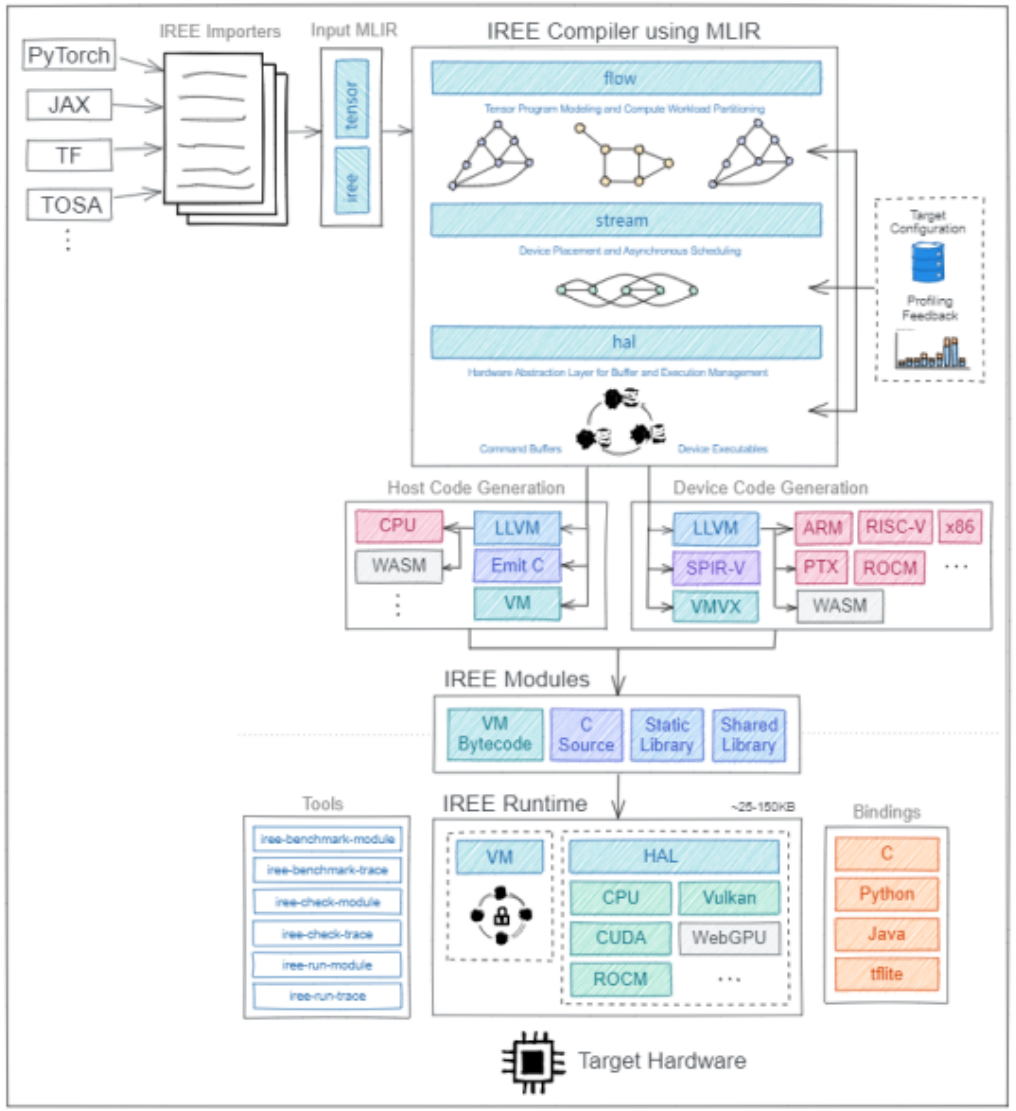
SiFive owned and maintained component

SiFive collaboration with others

Component provided by others

IREE (Intermediate Representation Execution Environment)

MLIR e2e based compiler & runtime



Compiler

- Support various **front-ends** and **models**
- Provide tiling policy for memory hierarchy
- Provide padding mechanism
- High level optimizations
- **Intra-op parallelization**
- **Code-gen through LLVM**
- Support MLIR code-gen + **ukernel execution**

Runtime

- **Inter-op parallelization** (async execution) and task scheduling
- Tracy Profiling
- **Linux & bare-metal** environments

Application

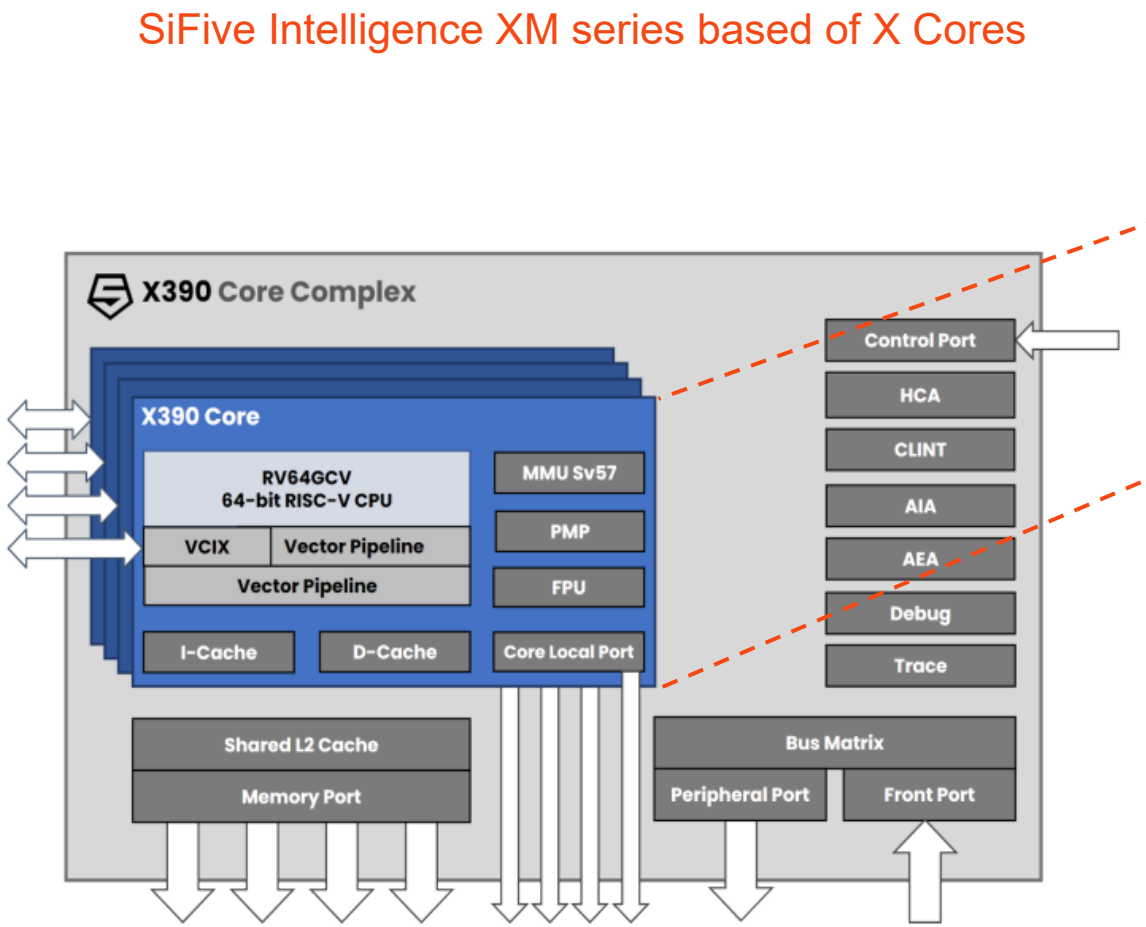
- Python binding

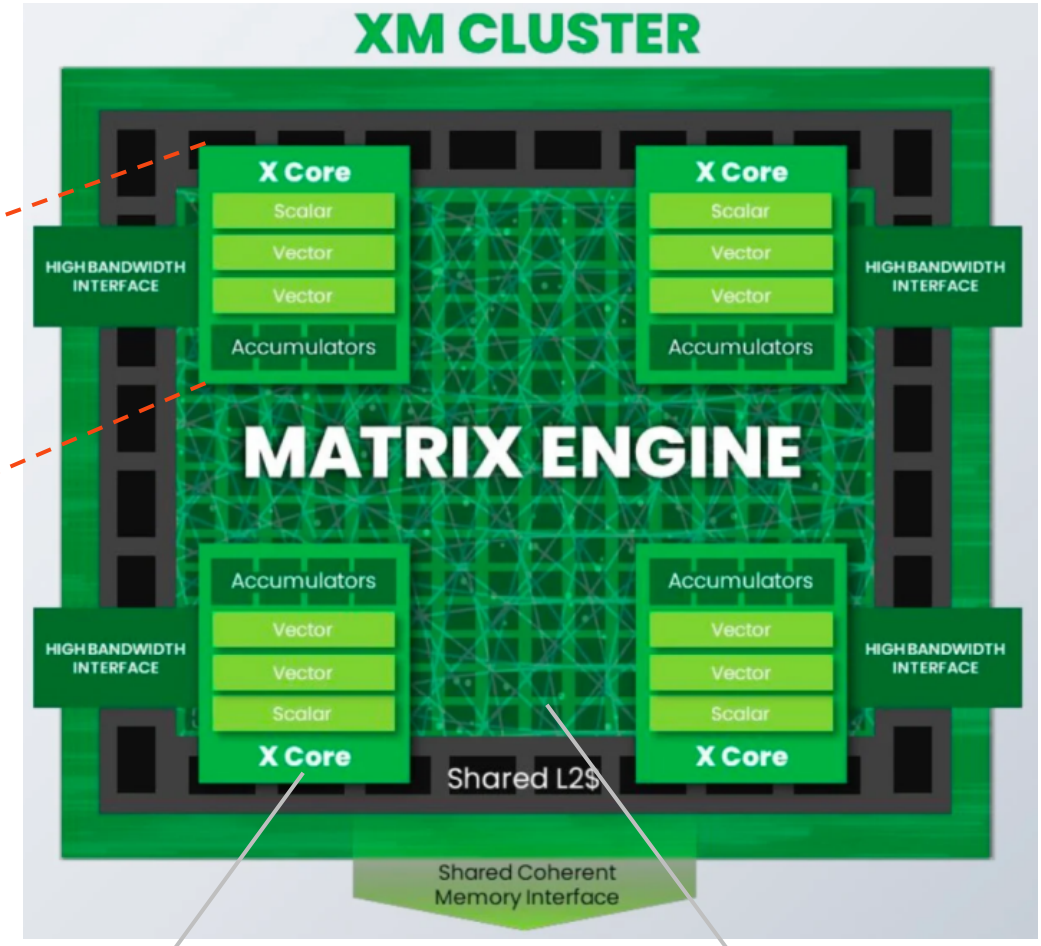
Open-source in **IREE** repo

SiFive Intelligence XM Platforms:
Integrate Matmul accelerator into IREE

SiFive Intelligence XM Platforms

SiFive Intelligence XM series based of X Cores



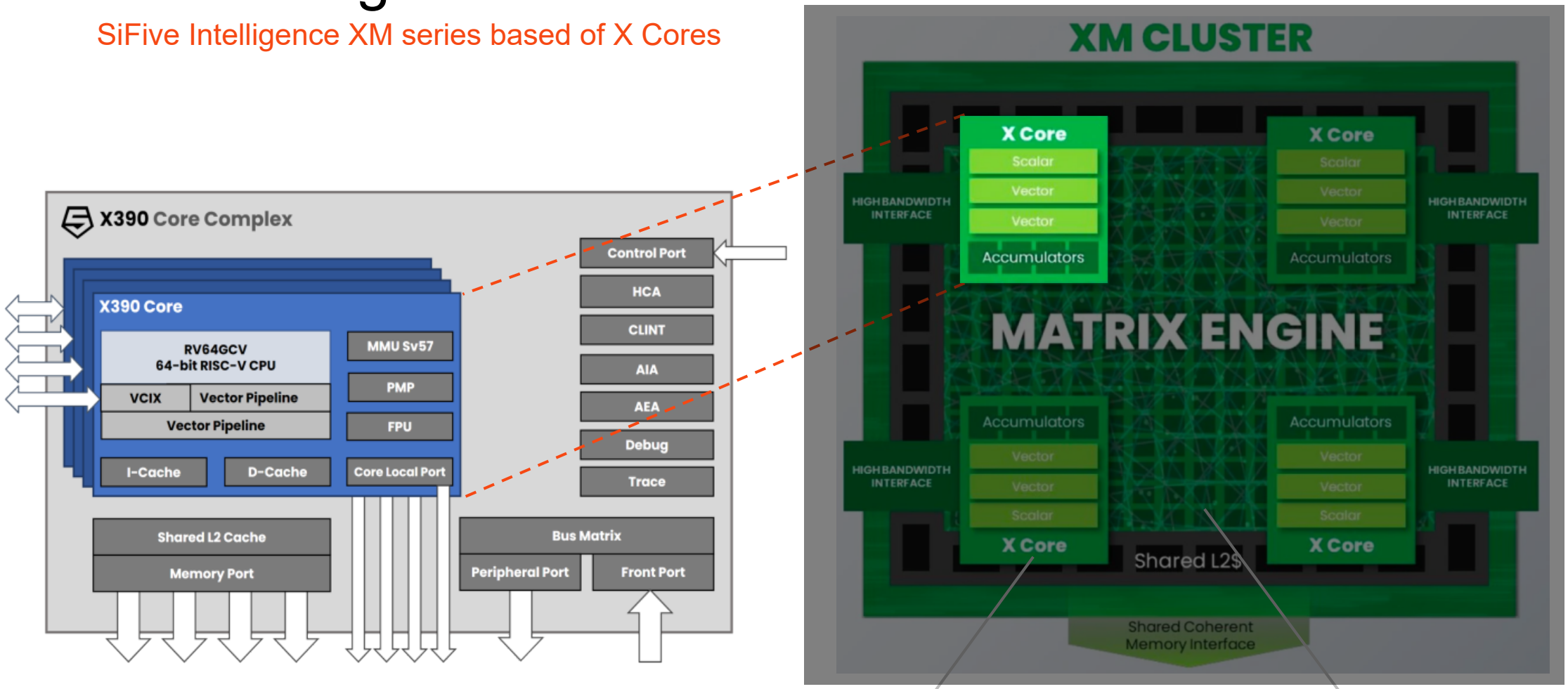


- Scalar + Vector (RVV)
- Do anything else

Do matrix multiplication + transpose

SiFive Intelligence XM Platforms

SiFive Intelligence XM series based of X Cores



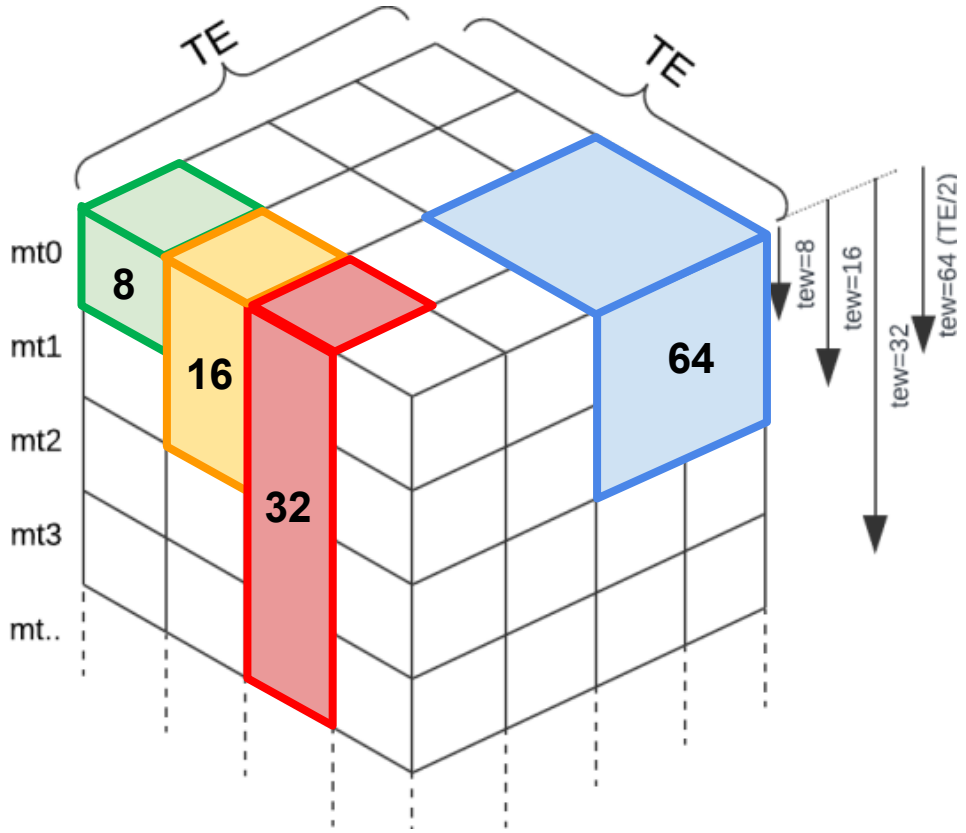
- Scalar + Vector (RVV)
- Do anything else

Do matrix multiplication + transpose

SiFive RISC-V Matrix Extension

Term Definition

- Tile State: Add per-hart architectural state in the form of square matrix tiles
- TE: Tile element = dimension per tile
- TEW: Tile element width (sew * Twiden)
- Twiden: Xsfmm enable & widening multiplier



Tile State

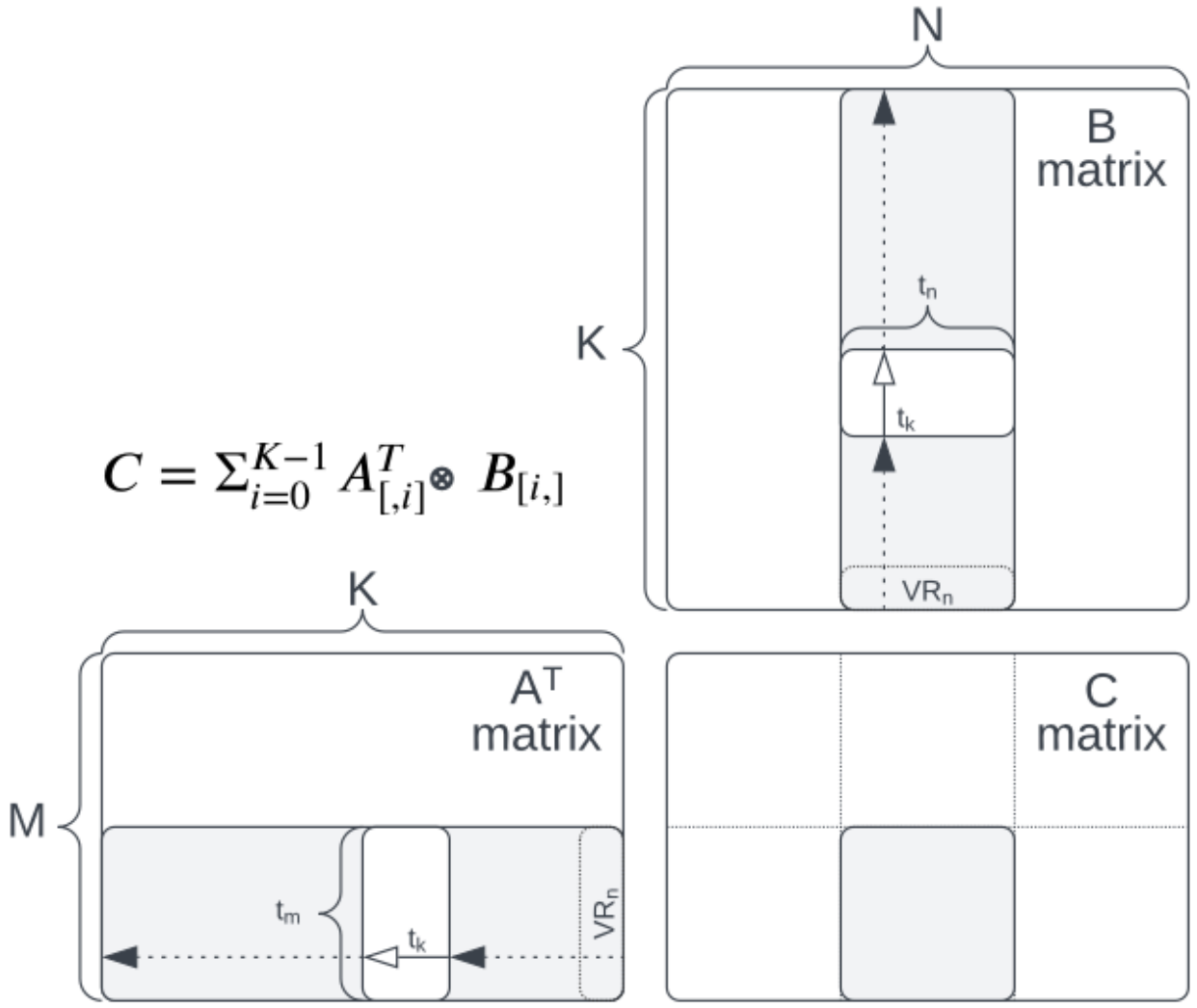
TEW=8 (TE x TE)	TEW=16 (TE x TE)	TEW=32 (TE x TE)	TEW=64 (TE x TE)
16 x Tile states mt0 ~ mt15	8 x Tile states mt0, mt2, mt4….mt14	4 x Tile state mt0, mt4, mt8, mt12	8x Tile state mt0, mt2, mt4….mt14

SiFive RISC-V Matrix Extension

SiFive Matrix Multiply Instruction

- **sf.mm** mt# vs2, vs1
  - Outer Product Operation
  - C[M,N] += A[K,M]^T*B[K,N]

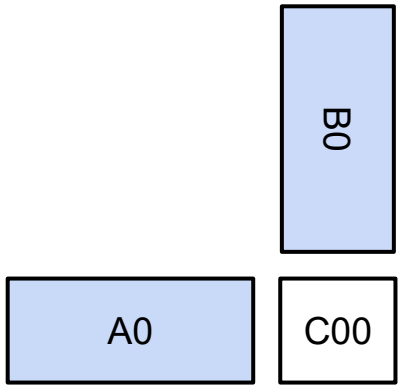
$$C = \Sigma_{i=0}^{K-1} A_{[,i]}^{T} \otimes B_{[i,]}$$



Tile K Loop Optimization for Matmul

Single Tile K Loop

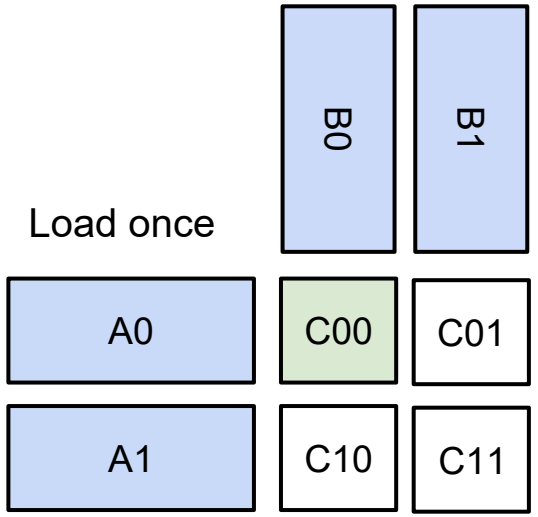
Before: IREE only support on …




```
...
For k: 1 to k

Load A0
Load B0
Matmul C00 += A0 * B0
```

Multi-tile Tile K Loop

Now: We make IREE also support on …



Instruction scheduling opt

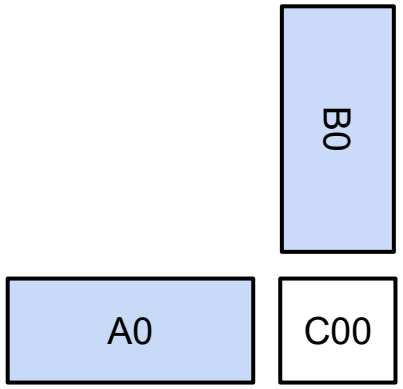
1

```
...
For k: 1 to k

Load A0
Load B0
Matmul C00 += A0 * B0
Load B1
Matmul C01 += A0 * B1
Load A1
Matmul C10 += A1 * B0
Matmul C11 += A1 * B1

Matmul C00 += A0 * B0
Matmul C01 += A0 * B1
Matmul C10 += A1 * B0
Matmul C11 += A1 * B1
```

Tile K Loop Optimization for Matmul

Single Tile K Loop

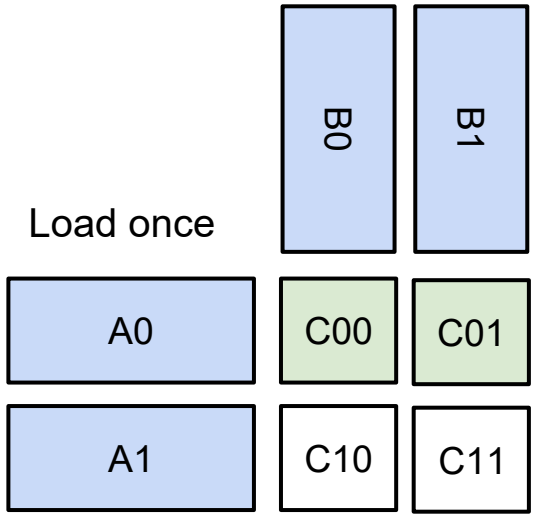
Before: IREE only support on …




```
...
For k: 1 to k

Load A0
Load B0
Matmul C00 += A0 * B0
```

Multi-tile Tile K Loop

Now: We make IREE also support on …



Instruction scheduling opt

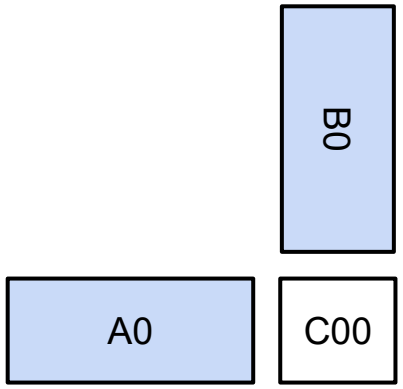
2

```
...
For k: 1 to k

Load A0
Load B0
Matmul C00 += A0 * B0
Load B1
Matmul C01 += A0 * B1
Load A1
Matmul C10 += A1 * B0
Matmul C11 += A1 * B1

Matmul C00 += A0 * B0
Matmul C01 += A0 * B1
Matmul C10 += A1 * B0
Matmul C11 += A1 * B1
```

Tile K Loop Optimization for Matmul

Single Tile K Loop

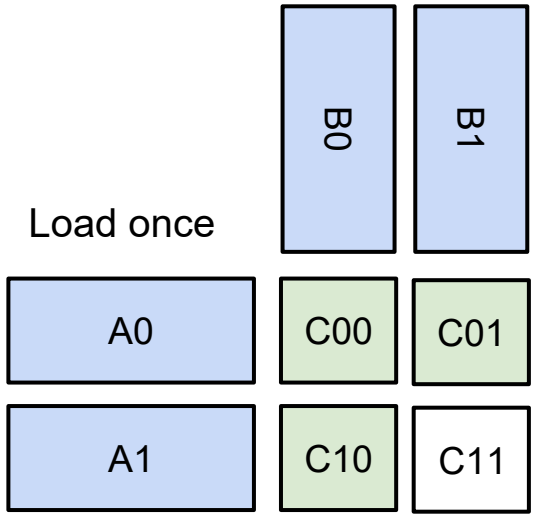
Before: IREE only support on …




```
...
For k: 1 to k

Load A0
Load B0
Matmul C00 += A0 * B0
```

Multi-tile Tile K Loop

Now: We make IREE also support on …



Instruction scheduling opt

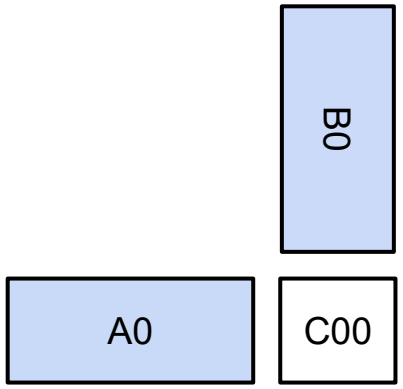
3

```
...
For k: 1 to k

Load A0
Load B0
Matmul C00 += A0 * B0
Load B1
Matmul C01 += A0 * B1
Load A1
Matmul C10 += A1 * B0
Matmul C11 += A1 * B1

Matmul C00 += A0 * B0
Matmul C01 += A0 * B1
Matmul C10 += A1 * B0
Matmul C11 += A1 * B1
```

Tile K Loop Optimization for Matmul

Single Tile K Loop

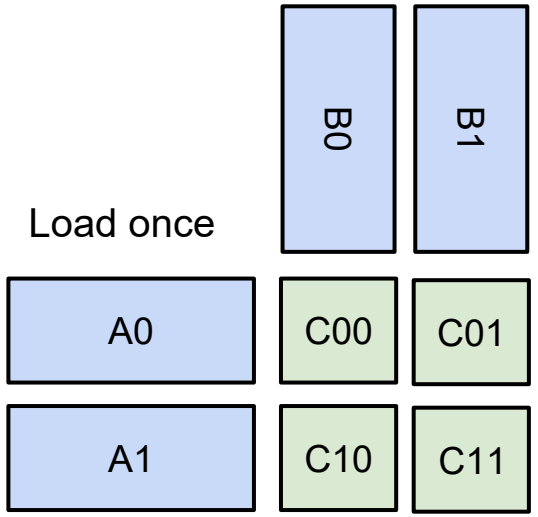
Before: IREE only support on …




```
...
For k: 1 to k

Load A0
Load B0
Matmul C00 += A0 * B0
```

Multi-tile Tile K Loop

Now: We make IREE also support on …



Instruction scheduling opt

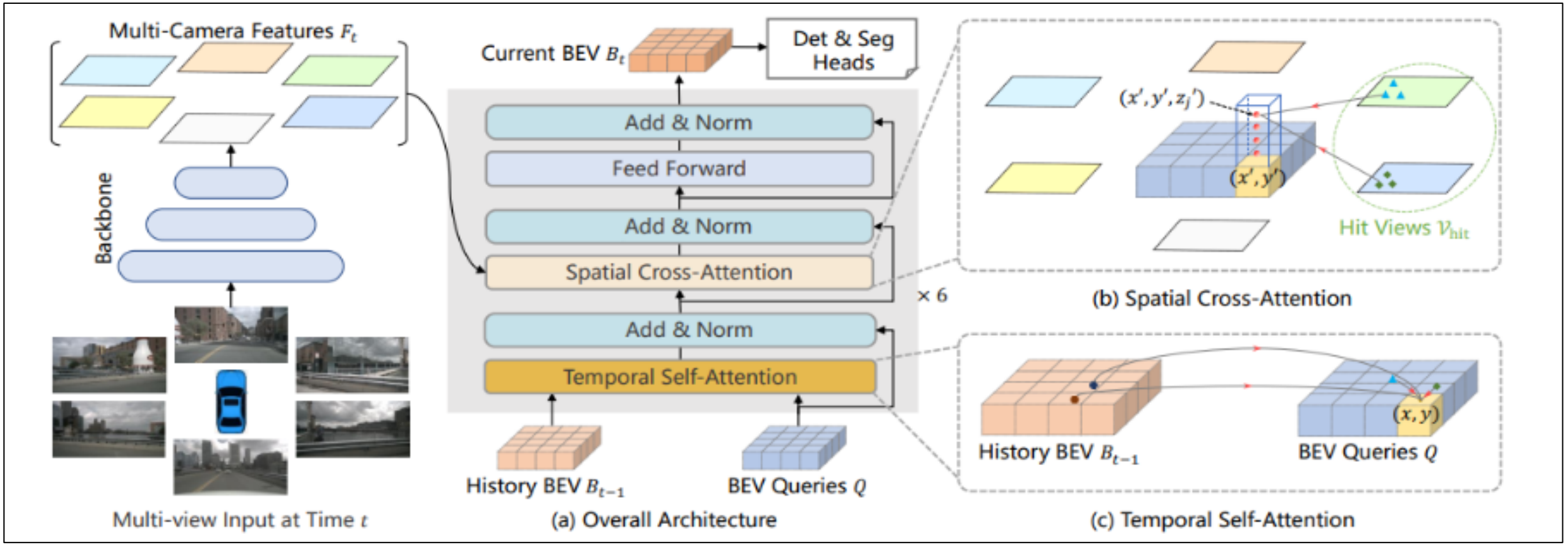
```
...
For k: 1 to k

Load A0
Load B0
Matmul C00 += A0 * B0
Load B1
Matmul C01 += A0 * B1
Load A1
Matmul C10 += A1 * B0
Matmul C11 += A1 * B1

Matmul C00 += A0 * B0
Matmul C01 += A0 * B1
Matmul C10 += A1 * B0
Matmul C11 += A1 * B1
```

4

End-to-end deployment:
Pytorch BEV Perception models to RISC-V

SiFive

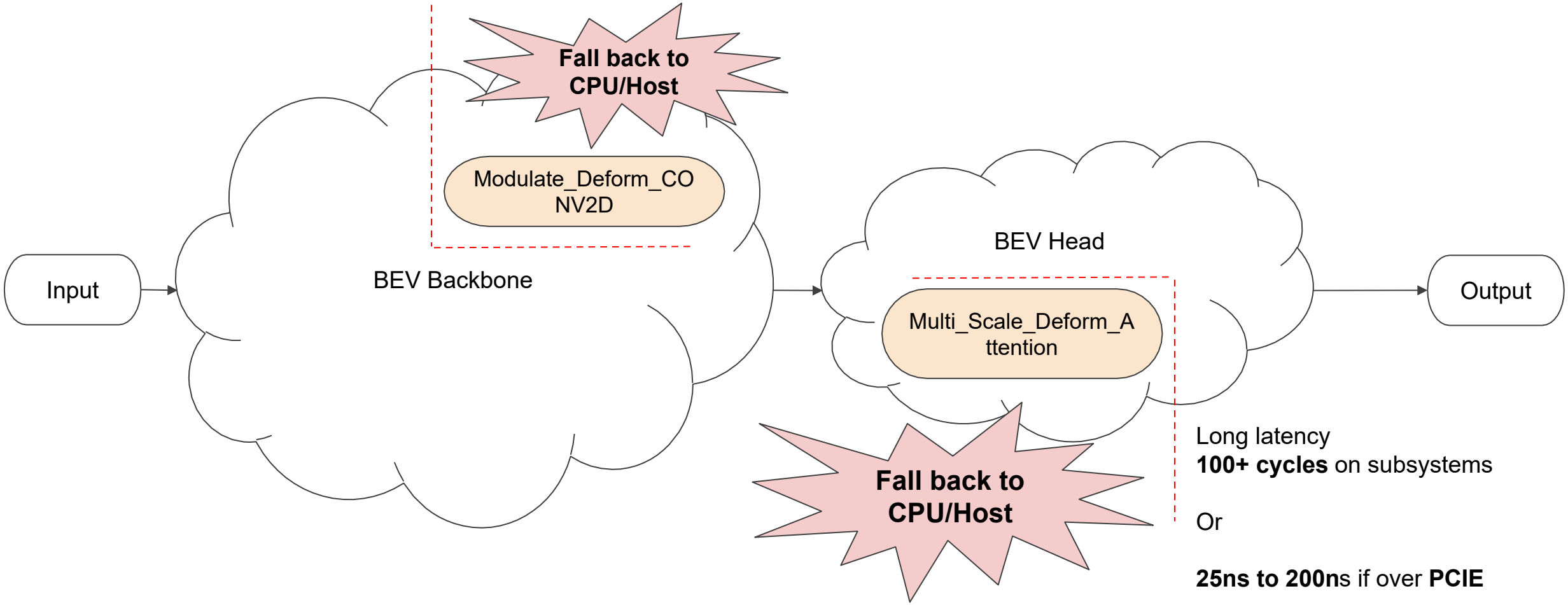
BEV(Bird's Eye View) Former Perception Models



BEVFormer Github

What is the problem?

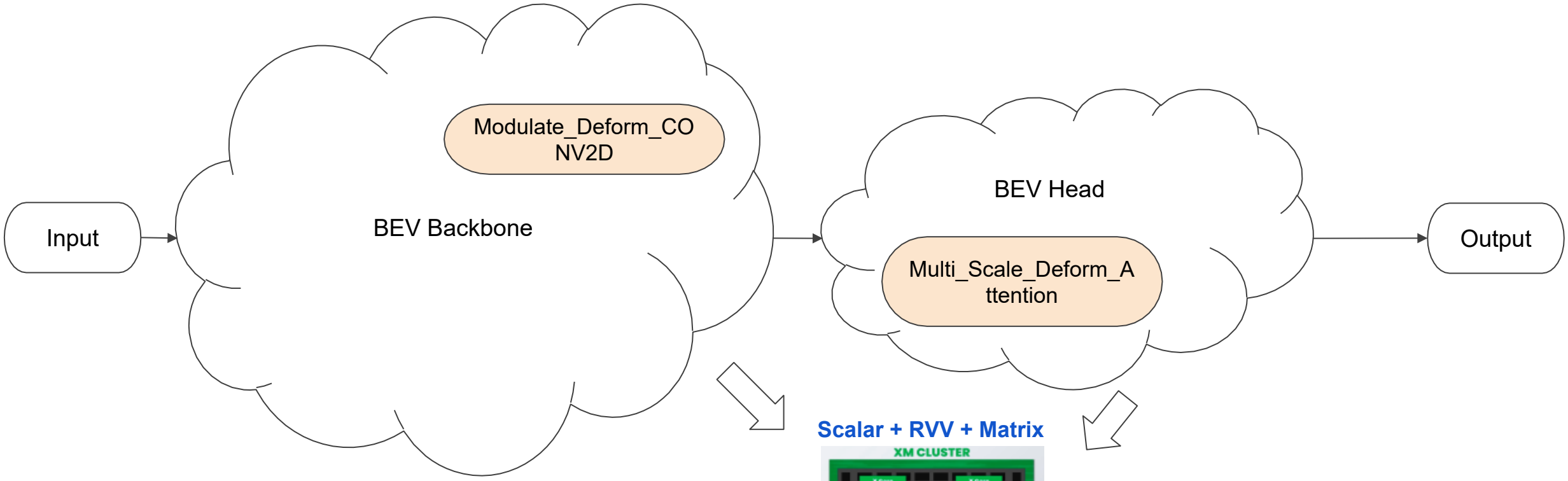
In general Heterogeneous platforms



SiFive

SiFive Intelligence XM Solve the pain point

An Unified Graph be executed in an unified platform getting lowest latency



An unified binary

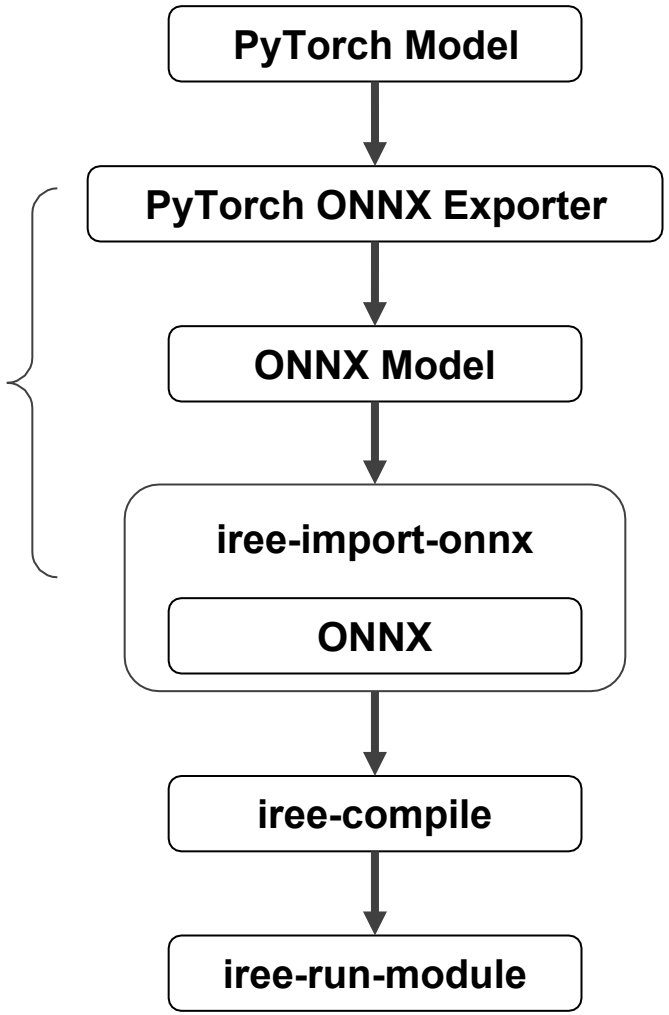
Objdump the compiled IREE executable(vmfb)

Scalar + RVV + Matrix

```
   3bce0: b3 83 76 00    add     t2, a3, t2
; bevdepth_17.mlir:1044
   3bce4: 33 8e 67 40    sub     t3, a5, t1
   3bce8: 57 7e 0e 84    sf.vsettn       t3, t3
   3bcec: 57 60 e0 43    sf.vtzero.t     mt0
   3bcf0: b3 4e b3 20    sh2add  t4, t1, a1
   3bcf4: 13 0f b0 01    li      t5, 0x1b
   3bcf8: 93 8f 02 00    mv      t6, t0
   3bcfc: 57 f0 08 84    sf.vsettn       zero, a7
   3bd00: 07 e4 0f 02    vle32.v v8, (t6)
   3bd04: 57 70 0e 84    sf.vsettn       zero, t3
   3bd08: 07 e8 0e 02    vle32.v v16, (t4)
   3bd0c: 77 10 88 f2    sf.mm.f.f       mt0, v8, v16
   3bd10: 93 8f 0f 02    addi    t6, t6, 0x20
   3bd14: 13 0f ff ff    addi    t5, t5, -0x1
   3bd18: 93 8e 0e 10    addi    t4, t4, 0x100
   3bd1c: e3 10 0f fe    bnez    t5, 0x3bcfc <.text+0x223b4>
   3bd20: 93 0e 00 00    li      t4, 0x0
```

The end-to-end flow

- Required to convert to ONNX because BEV series models most stick on pytorch-1.x and highly depends on OpenMMLab libraries



Creating a PyTorch model with custom OPs.

Adding registered OPs to the TorchScript.

The ONNX model w/ custom OPs

Adding registered OPs to in ONNX.
Translate ONNX to torch-mlir

Adding lowering path for the custom OPs
Adding custom OPs implementation

Convert PyTorch model to an unified BEV Perception model

Summary & Future works

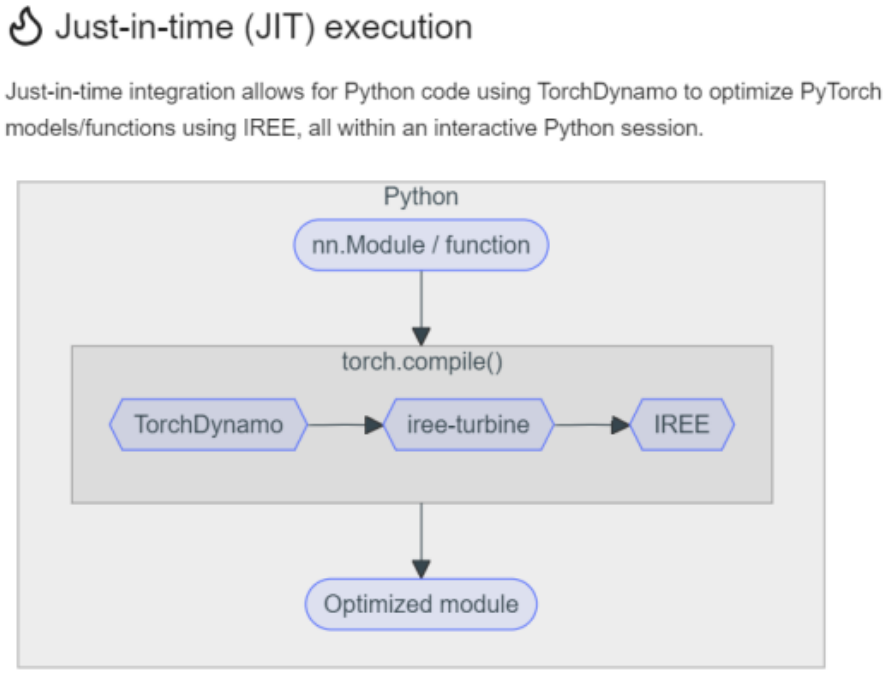
Takeaways

- SiFive AI/ML Software Stack: Highly optimized and fast time-to-market
- **Keep upstream our work to IREE**(core components)
- **A unified binary and platform**(XM Intelligence)
- Welcome collaborations to bring more AI models to RISC-V

We're happy to share our open-source work and insights

Coming Soon

- More LLM and BEV Perception models optimization & deployment
- Natively running PyTorch on RISC-V platforms



THANK YOU