ISCAS 中国科学院软件研究所学术年会暨重点实验室科技活动周 2025 第十届

学术论文

QiMeng-TensorOp：一句提示词足以生成基于硬件原语的高性能张量算子

张续志，彭少辉，周其睿，文渊博，郭崎，陈睿智，
朱鑫国，熊伟强，陈海鑫，马聪颖，高科，赵琛，
武延军，陈云霁，李玲*

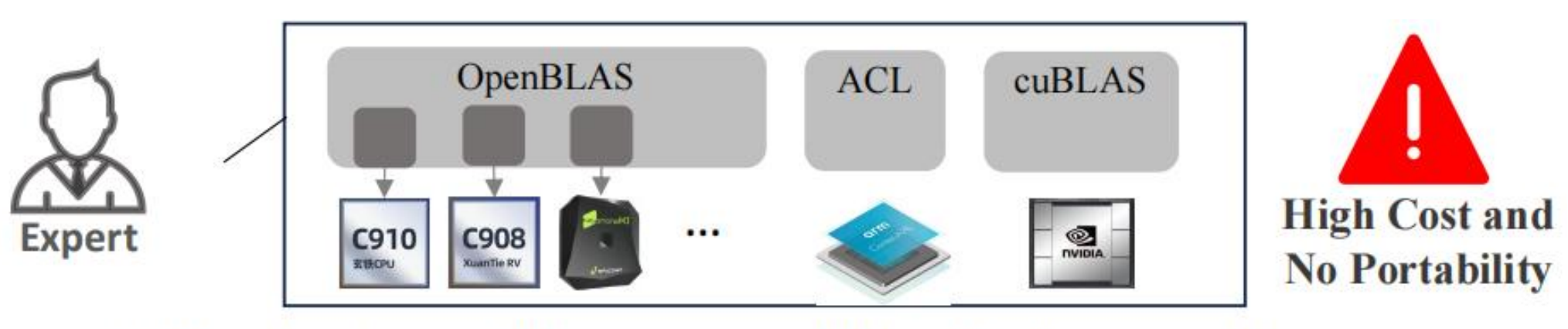
IJCAI 2025（CCF A）

联系方式：李玲，liling@iscas.ac.cn

工作介绍

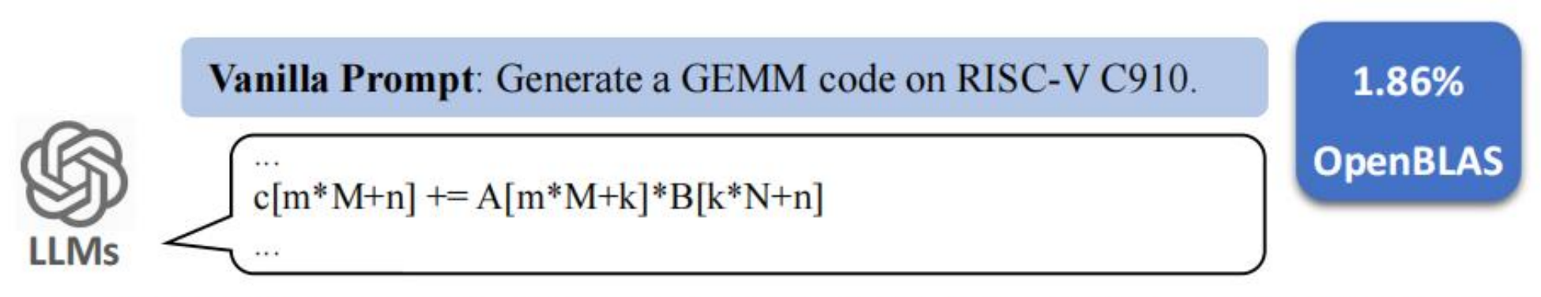
计算密集型张量算子构成了大语言模型和深度神经网络中90%的计算。由于手动优化实现不仅耗时还缺乏可移植性，高效自动地利用硬件原语生成高性能张量算子，对于 RISC-V、ARM 和 GPU 等多样化且不断演进的硬件架构至关重要。大语言模型擅长生成高级语言代码，但在充分理解硬件特性并生成高性能张量算子方面仍存在不足。

我们提出了一种**使用一句用户提示词自动生成张量算子**的框架（QiMeng-TensorOp），该框架使大语言模型自动利用硬件特性，通过硬件原语生成张量算子，并针对不同硬件平台优化参数，最终实现最佳性能。在**多种硬件平台**、**前沿大语言模型**及**典型张量算子**上的实验结果表明，QiMeng-TensorOp**能有效利用各类硬件平台的计算潜力，自动生成具有卓越性能的张量算子。**



(a) Hand-optimized libraries provided by Hardware vendors

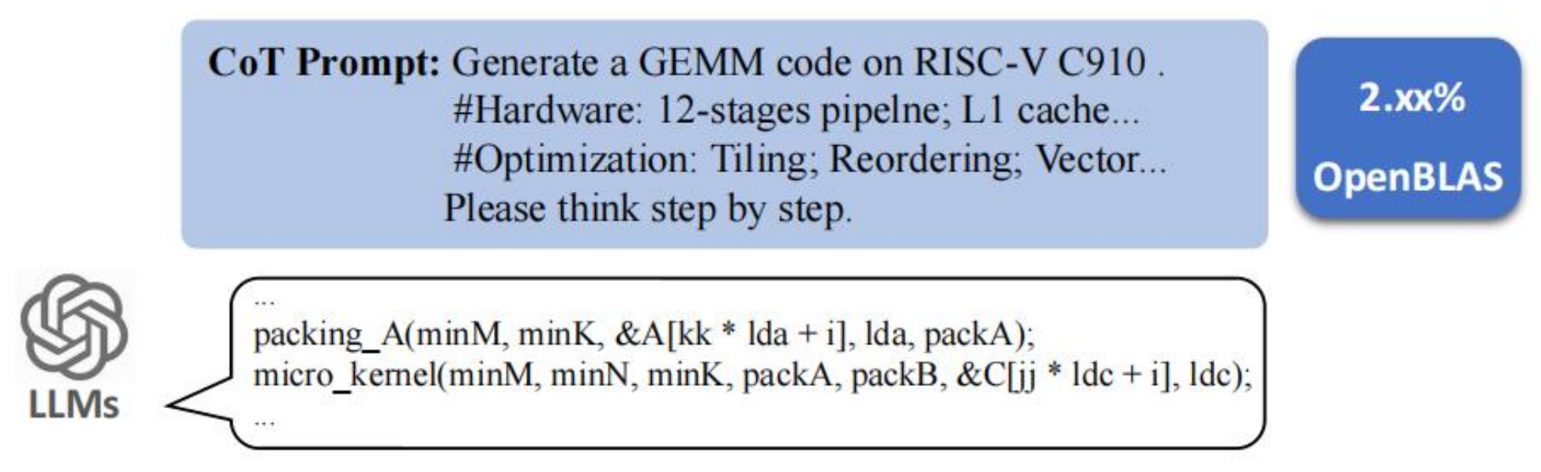
手工算子开发成本高且可移植性差



Vanilla Prompt: Generate a GEMM code on RISC-V C910.

```
...
c[m*M+n] += A[m*M+k]*B[k*N+n]
...
```

(b) Vanilla prompts generate hardware-independent C code for GEMM

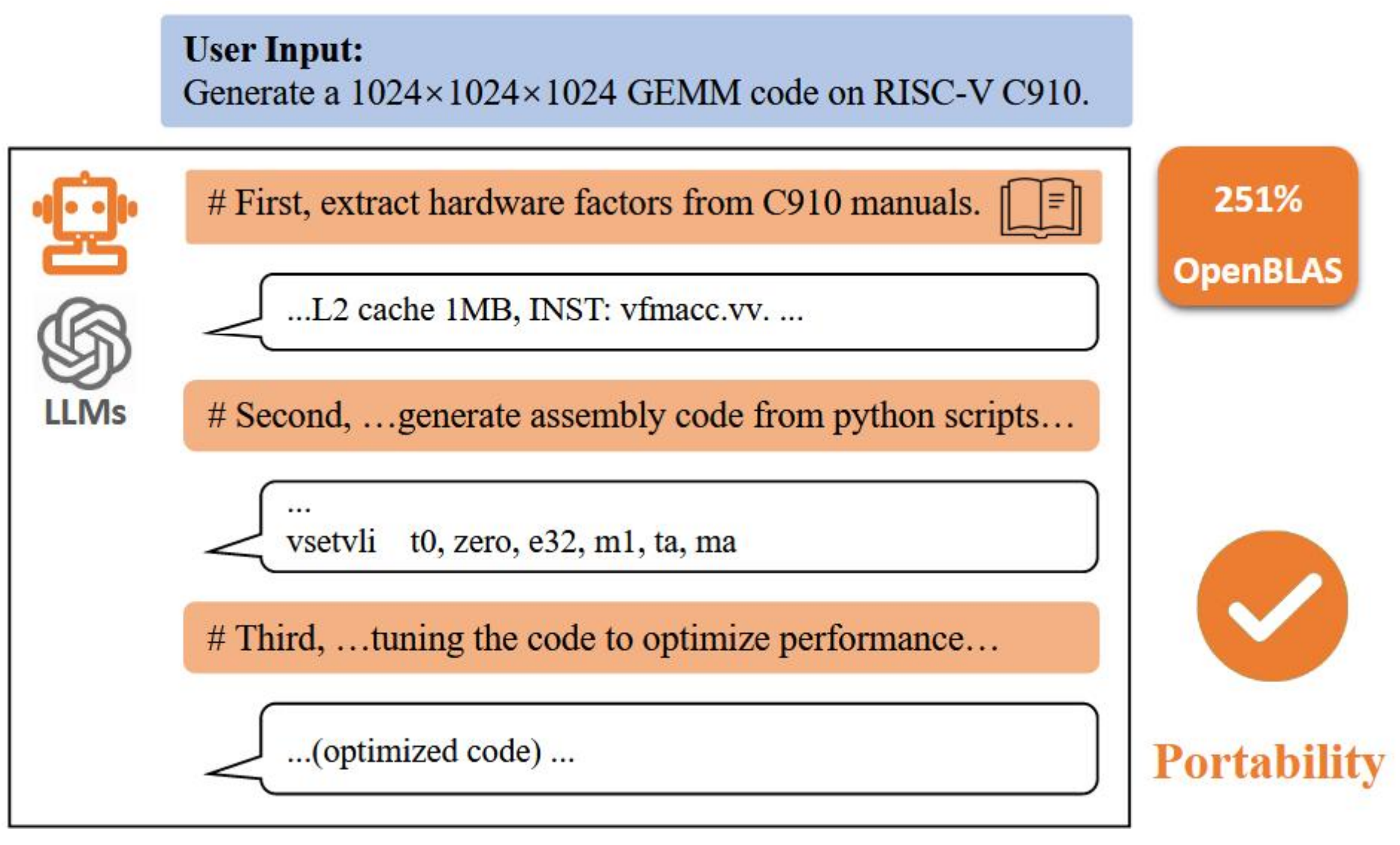


CoT Prompt: Generate a GEMM code on RISC-V C910 .
#Hardware: 12-stages pipelne; L1 cache...
#Optimization: Tiling; Reordering; Vector...
Please think step by step.

```
...
packing_A(minM, minK, &A[kk * lda + i], lda, packA);
micro_kernel(minM, minN, minK, packA, packB, &C[jj * ldc + i], ldc);
...
```

(c) CoT Prompt generate C code with simple block tiling for GEMM

大模型直接生成算子性能不足



User Input:
Generate a 1024×1024×1024 GEMM code on RISC-V C910.

\# First, extract hardware factors from C910 manuals.

...L2 cache 1MB, INST: vfmacc.vv. ...

\# Second, …generate assembly code from python scripts…

```
...
vsetvli    t0, zero, e32, m1, ta, ma
```

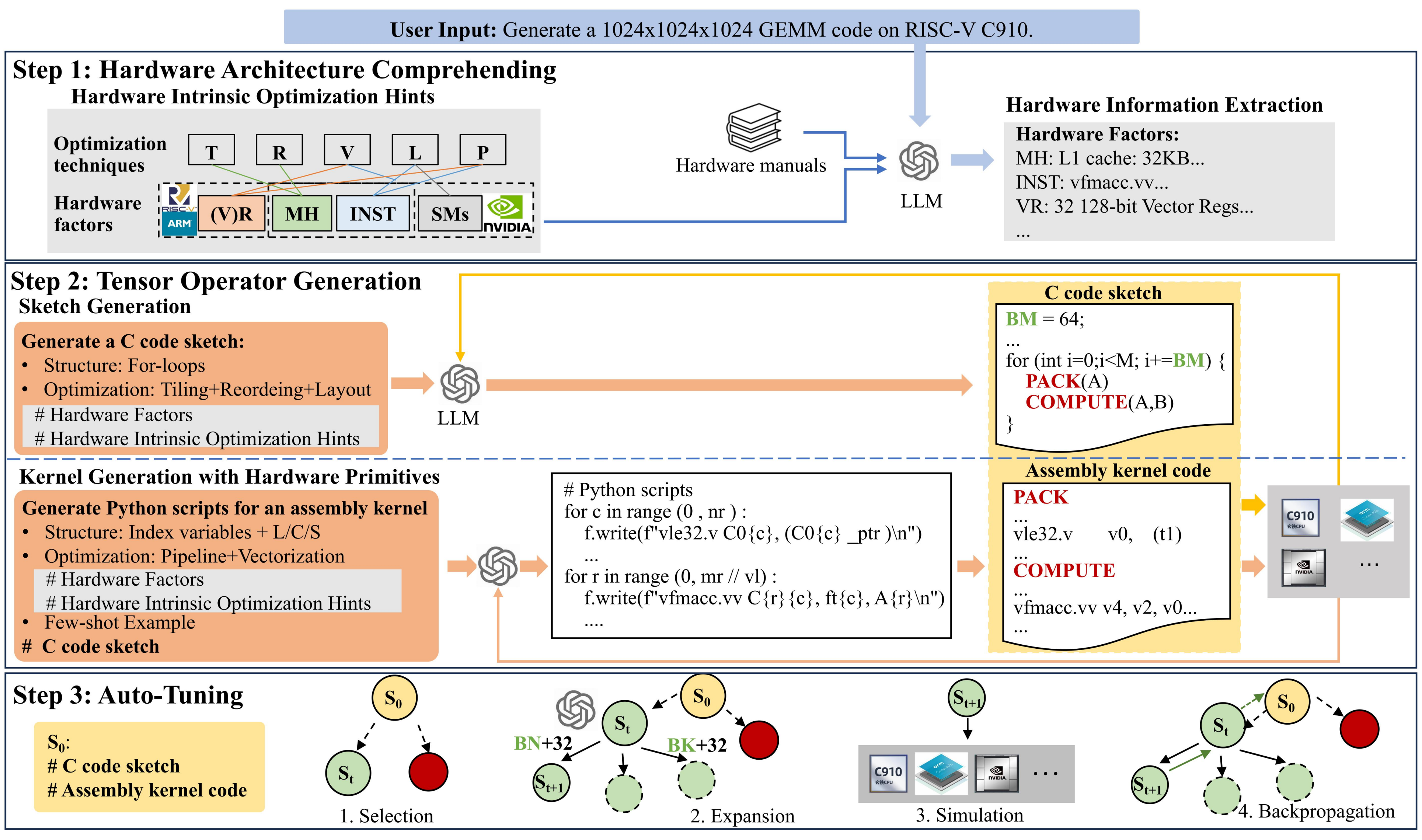
\# Third, …tuning the code to optimize performance…

...(optimized code) ...

一句用户提示词生成高性能张量算子

总体框架

- **硬件架构理解**：使用**优化元提示**和**硬件信息**激发大模型对硬件架构和张量算子优化的理解
- **张量算子生成**：通过**分层策略**分别生成张量算子的骨架和内核代码，实现**良好的代码结构**
- **自动调优**：利用**基于大模型的MCTS**发掘细微的优化机会，进一步**调优性能**



User Input: Generate a 1024x1024x1024 GEMM code on RISC-V C910.

Step 1: Hardware Architecture Comprehending

Hardware Intrinsic Optimization Hints

Hardware Information Extraction

Hardware Factors:
MH: L1 cache: 32KB...
INST: vfmacc.vv...
VR: 32 128-bit Vector Regs...
...

Step 2: Tensor Operator Generation

Sketch Generation

Generate a C code sketch:
- Structure: For-loops
- Optimization: Tiling+Reordeing+Layout

\# Hardware Factors
\# Hardware Intrinsic Optimization Hints

C code sketch

```
BM = 64;
...
for (int i=0;i<M; i+=BM) {
  PACK(A)
  COMPUTE(A,B)
}
```

Kernel Generation with Hardware Primitives

Generate Python scripts for an assembly kernel
- Structure: Index variables + L/C/S
- Optimization: Pipeline+Vectorization
  - \# Hardware Factors
  - \# Hardware Intrinsic Optimization Hints
- Few-shot Example

 # C code sketch

```
# Python scripts
for c in range (0 , nr ) :
   f.write(f"vle32.v C0{c}, (C0{c} _ptr )\n")
   ...
for r in range (0, mr // vl) :
   f.write(f"vfmacc.vv C{r}{c}, ft{c}, A{r}\n")
   ....
```

Assembly kernel code

```
PACK
...
vle32.v       v0,    (t1)
...
COMPUTE
...
vfmacc.vv v4, v2, v0...
...
```

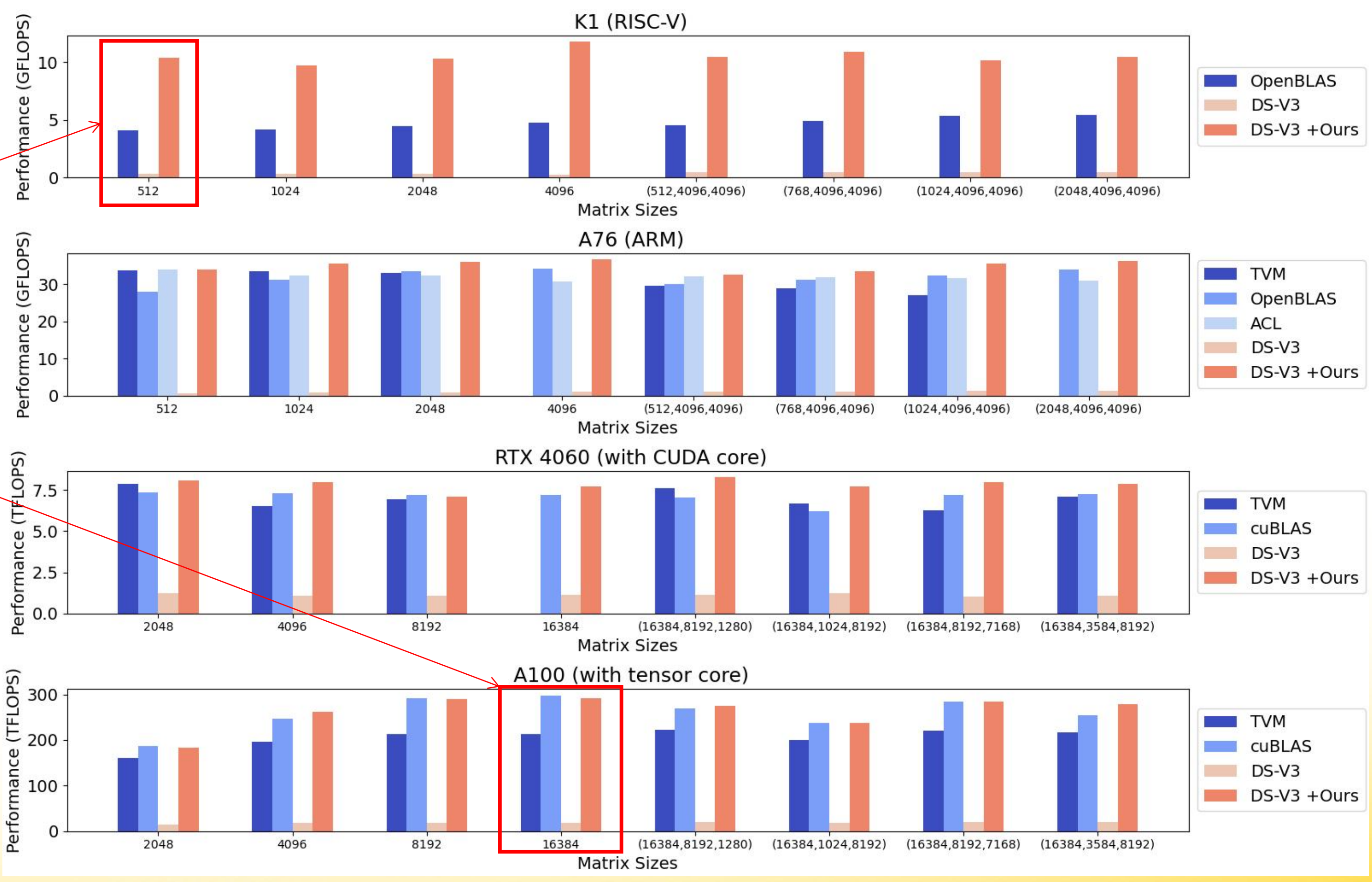
Step 3: Auto-Tuning

S_0:
 # C code sketch
 # Assembly kernel code

实验评估

高性能且低开发代价

- **在RISC-V平台上最高达到了OpenBLAS库的2.51倍性能**
- **在GPU平台上相比于TVM实现了显著的加速，矩阵乘算子上最高达到1.38倍的性能**
- **相比于高级程序员，QiMeng-TensorOp将开发代价从几天缩短到了二十分钟**



不同平台上的矩阵乘算子生成结果