QiMeng-Attention：SOTA注意力算法生成SOTA注意力代码
QiMeng-Attention: SOTA Attention Operator is generated by SOTA Attention Algorithm

周其睿，彭少辉，熊伟强，陈海鑫，文渊博，李昊宸，李玲*，
郭崎，赵永威，高科，陈睿智，武延军，赵琛，陈云霁

ACL 2025 (CCF-A)

主要联系人：周其睿，zhouqirui22s@ict.ac.cn

研究背景

在大语言模型中，注意力算子已成为处理长上下文任务时的关键性能瓶颈。FlashAttention是目前应用最广且最高效的注意力加速算法，其实现仍然需要依赖耗时且针对特定硬件的手动优化，限制了算法在不同GPU架构间的适应性。现有大语言模型在代码生成任务中展现巨大潜力，但仍然难以生成高性能的注意力计算代码。核心挑战在于：模型无法真正理解注意力算子复杂的数据流与计算过程，也无法有效运用底层原语来充分挖掘GPU性能。

针对上述挑战，本研究提出了一种面向大语言模型的思考语言（LLM-TL），结合两阶段推理工作流，大语言模型可自动生成适配不同GPU的FlashAttention实现。经验证，大语言模型结合本方案生成的注意力算子代码相比未优化的代码，最高可实现35.16倍加速。此外，该方法不仅在多数场景下超越人工优化库，更能扩展至未受支持的硬件与数据类型，相较专家手工开发将耗时从数月压缩至分钟级。

方案设计

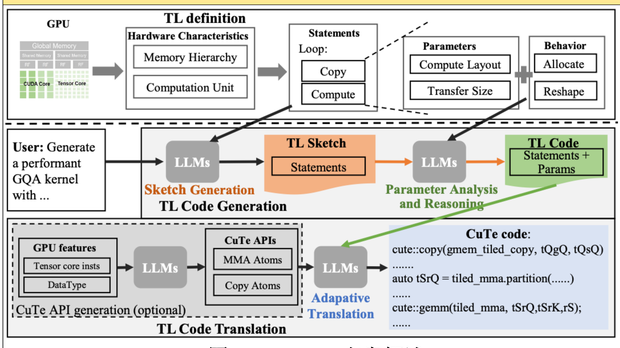


图1：LLM-TL方案概述

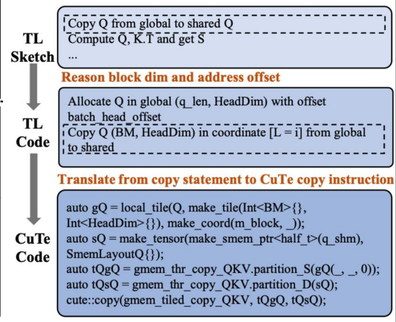


图2：两阶段推理代码变换

① 面向大语言模型的思考语言设计：提出并设计了一种面向大语言模型的思考语言（LLM-TL），将GPU执行过程抽象为两种核心原语：内存访问（Copy）和计算（Compute）。该语言将GPU硬件特性转化为语义级描述并屏蔽底层细节，从而充分发挥大语言模型的泛化能力与语义理解优势。

② 两阶段推理工作流：首先根据用户需求，大语言模型生成用于在语义层面描述算子执行流程的TL草图（TL Sketch），然后利用大语言模型进行参数推理，为TL草图中的Copy与Compute语句补充关键参数细节，生成完整TL代码；得益于TL设计的解耦特点，针对不同硬件特性，仅需在提示词中提供必要的硬件执行信息，利用大语言模型进行第二次推理，得到底层的硬件代码。

实验评估

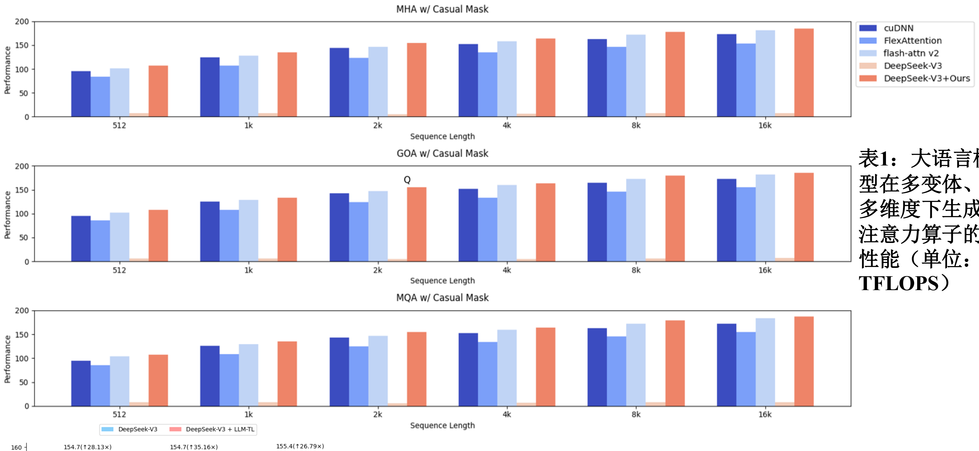
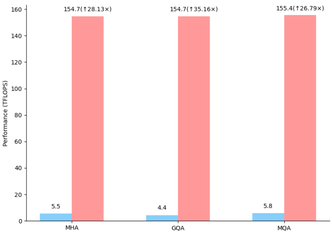


表1：大语言模型在多变体、多维度下生成注意力算子的性能（单位：TFLOPS）



	Time	TFLOPS
Human Expert	~ months	162.7
LLM-TL	10 mins	175.6

表3：本方案和人工方案的算子开发时间和性能比较

表2：使用本方案前后大语言模型生成注意力算子代码性能比较

总结：

1. 支持**多种变体**和**不同维度**的注意力算子代码生成；
2. 和大语言模型自行生成的未优化代码相比，使用本方案**最高可实现35.16倍加速**；
3. 实现周期从人工开发的数月级缩短至**分钟级**，且性能在多数场景下**超越人工优化**。