



GFFT:A Computational Graph based FFT Library

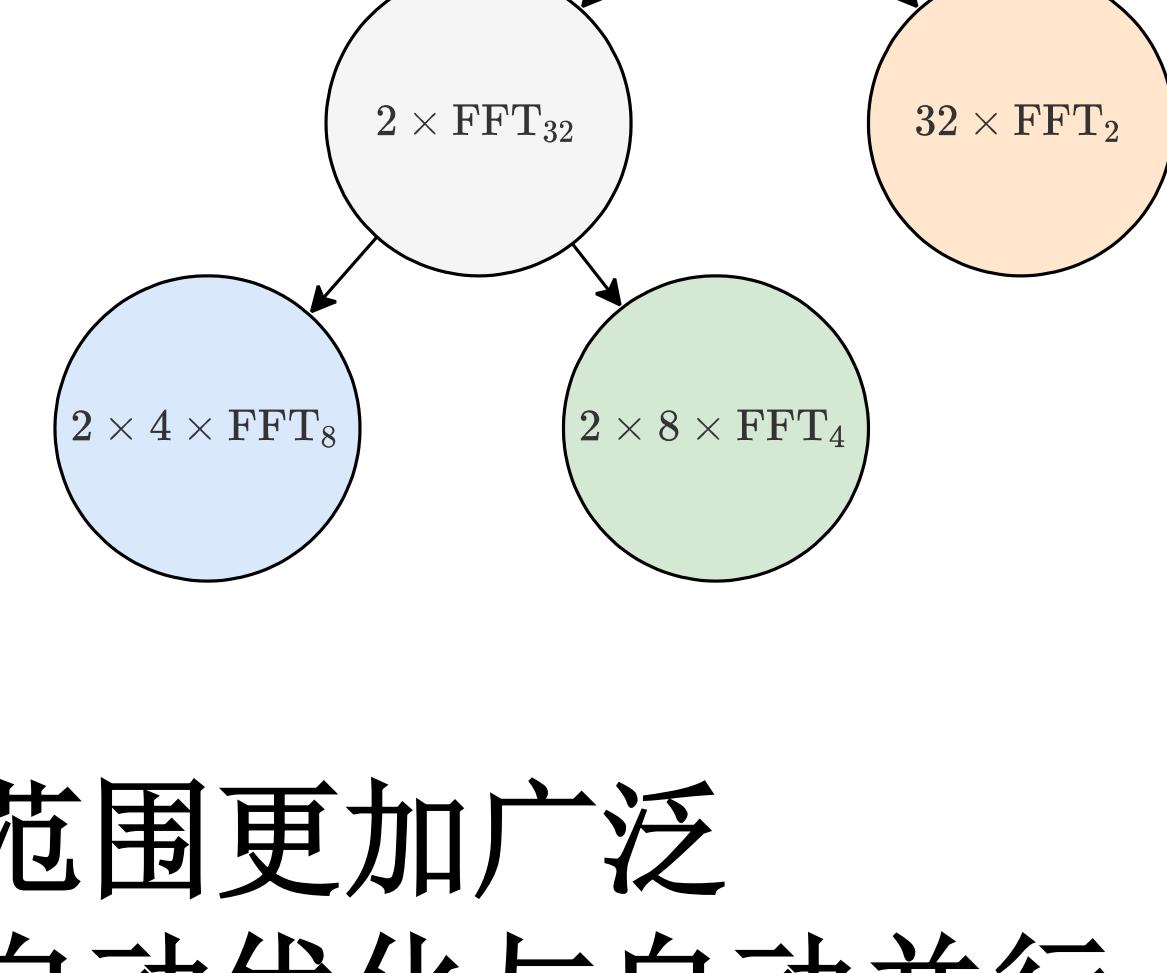
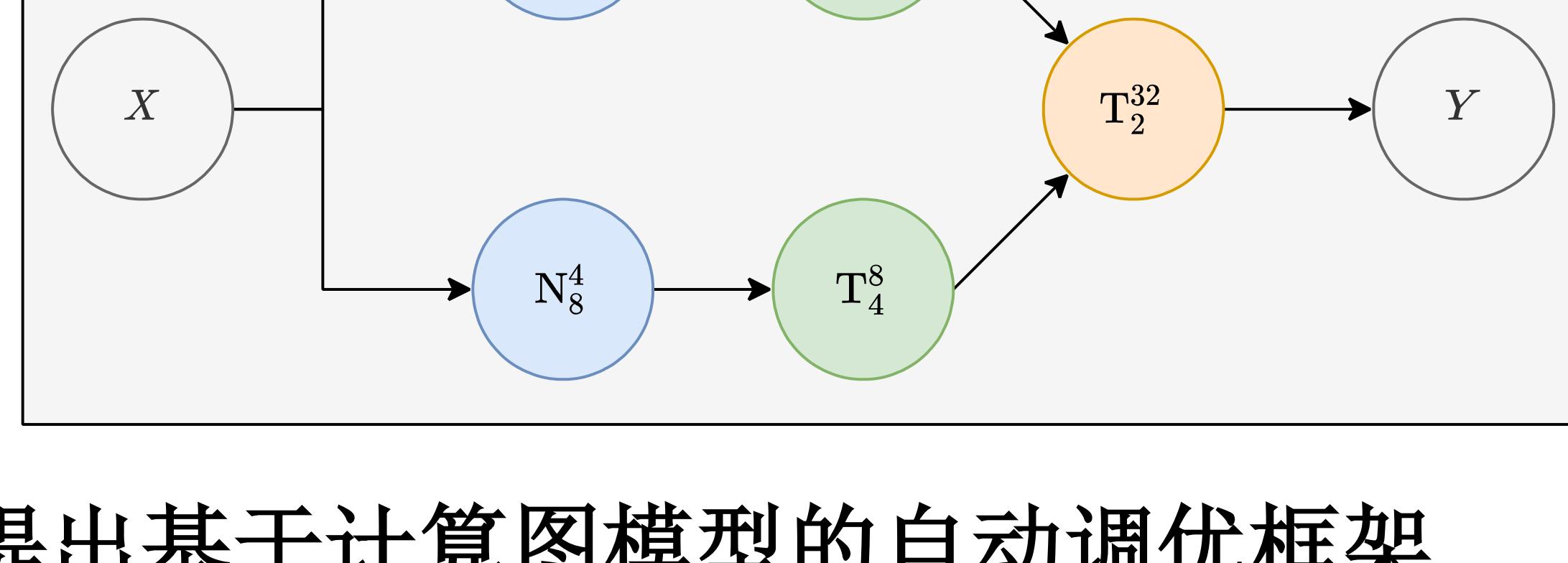
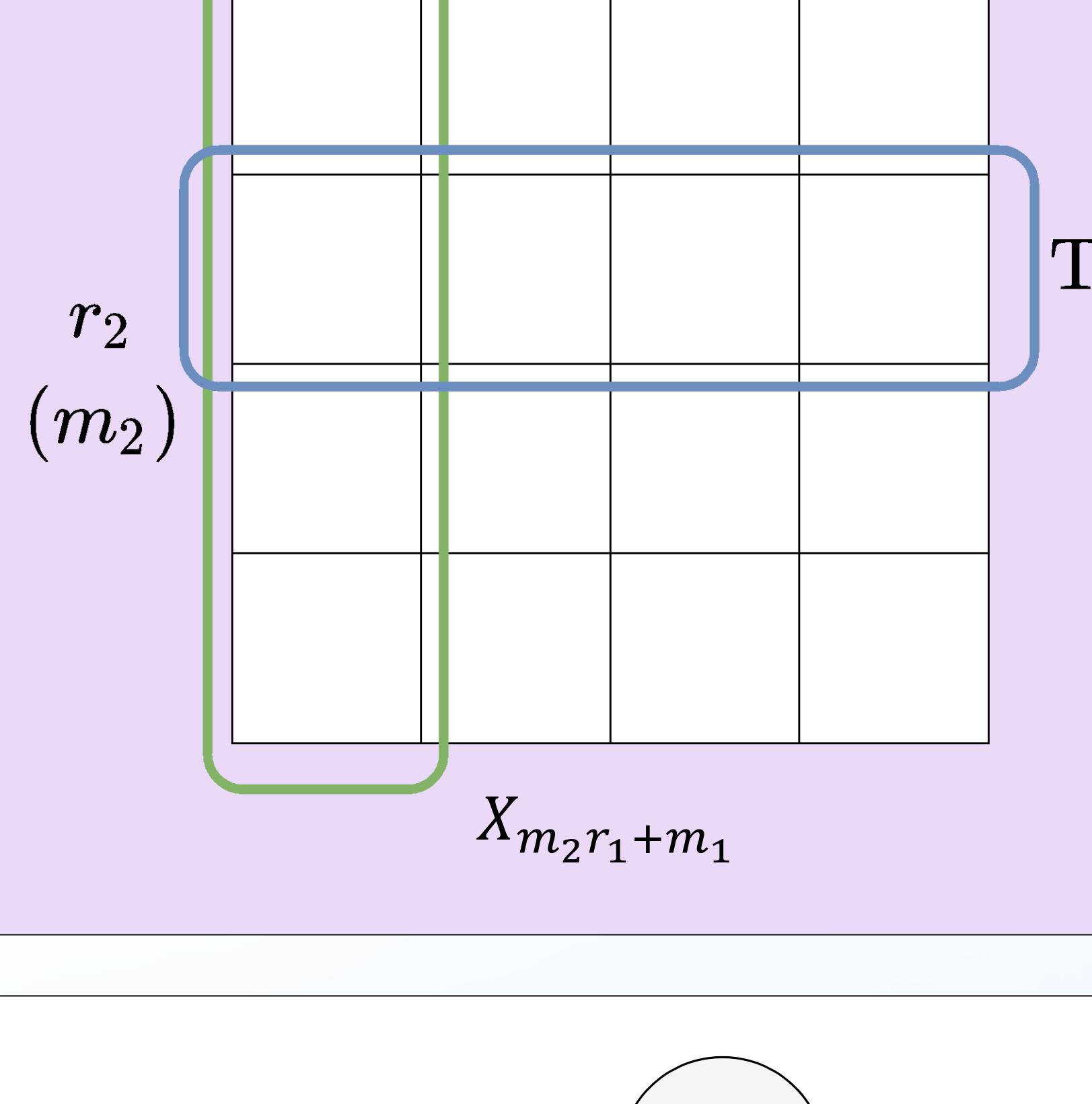
路青霖

刘芳芳 13466713051 fangfang@iscas.ac.cn

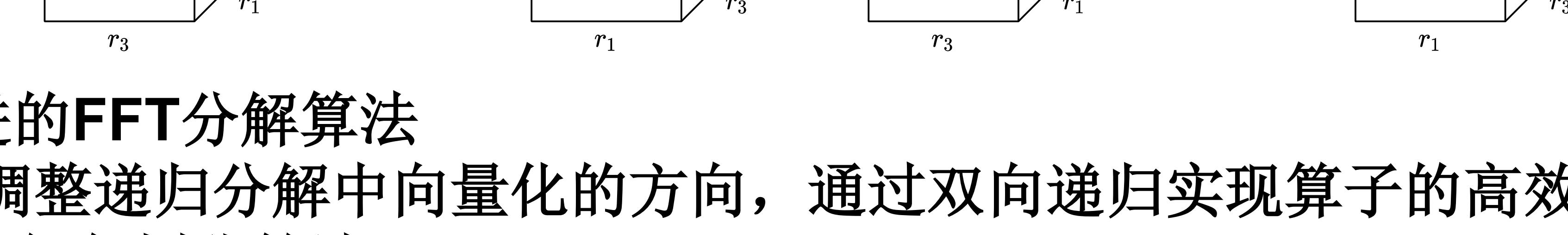
快速傅立叶变换数学库作为众多科学计算应用广泛使用的基础软件之一，提供针对硬件平台的高性能FFT计算。本工作从开源数学库软件FFTW分解方案搜索时间长、性能不稳定等问题入手，设计实现高性能FFT数学库软件GFFT，支持国产申威多核架构、X86架构等多核处理器平台。在申威多核处理器平台上单线程、32线程相比FFTW实现平均1.94倍、5.49倍加速。

$$Y_{k_2 r_2 + k_1} = \sum_{m_1=0}^{r_1-1} \left[\omega_n^{m_1 k_1} \left(\sum_{m_2=0}^{r_2-1} X_{m_2 r_1 + m_1} \omega_{r_2}^{m_2 k_1} \right) \right] \omega_{r_1}^{m_1 k_2}$$

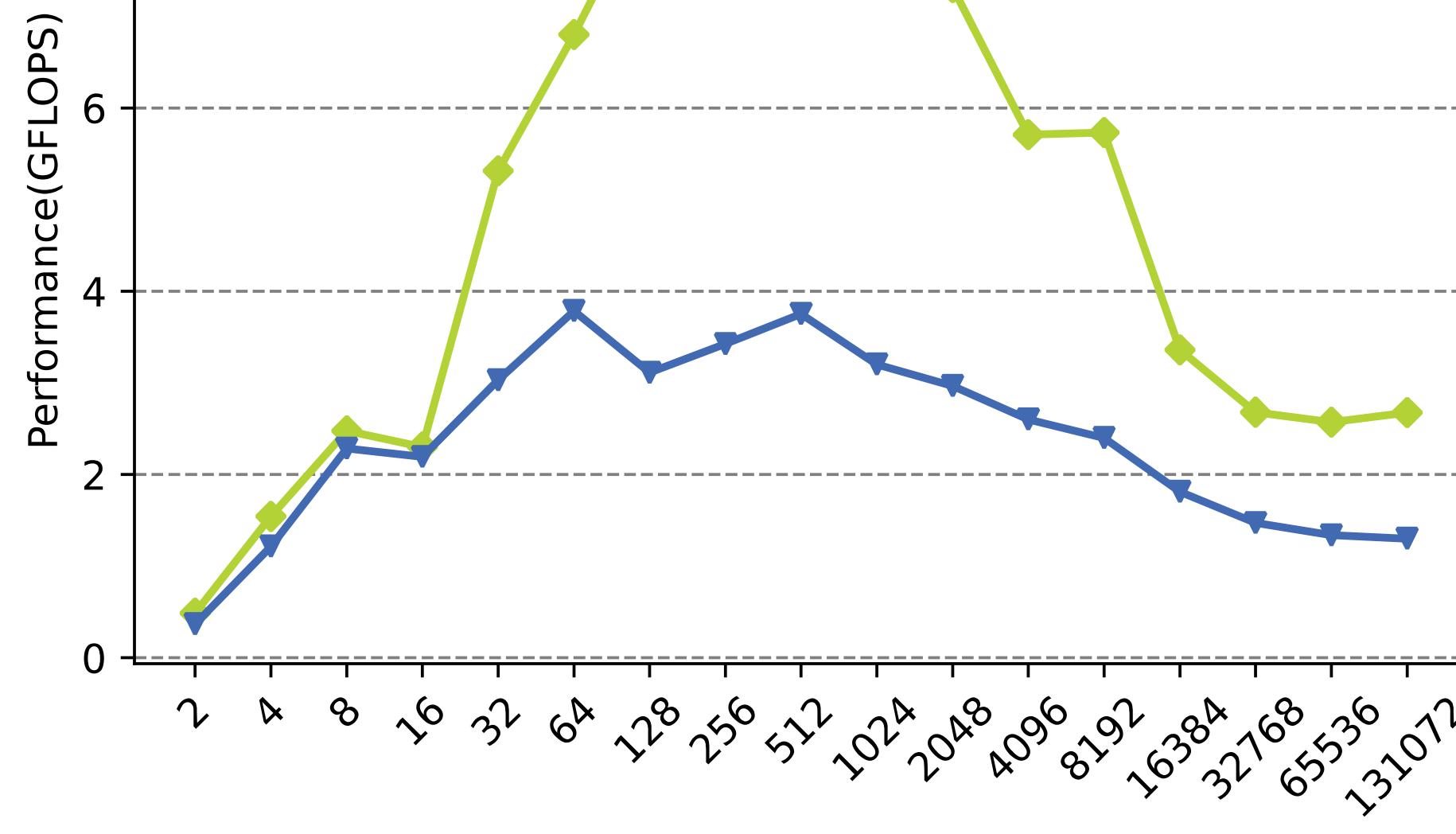
- FFT将序列作为二维矩阵，以多行、多列的形式实现计算规模的分解与计算
- 包含两种类型的计算过程：
 - 直接FFT计算（N类型计算）
 - 带旋转因子FFT计算（T类型计算）
- FFTW采用递归分解进行方案搜索，通过测量方案的执行时间选择最优方案



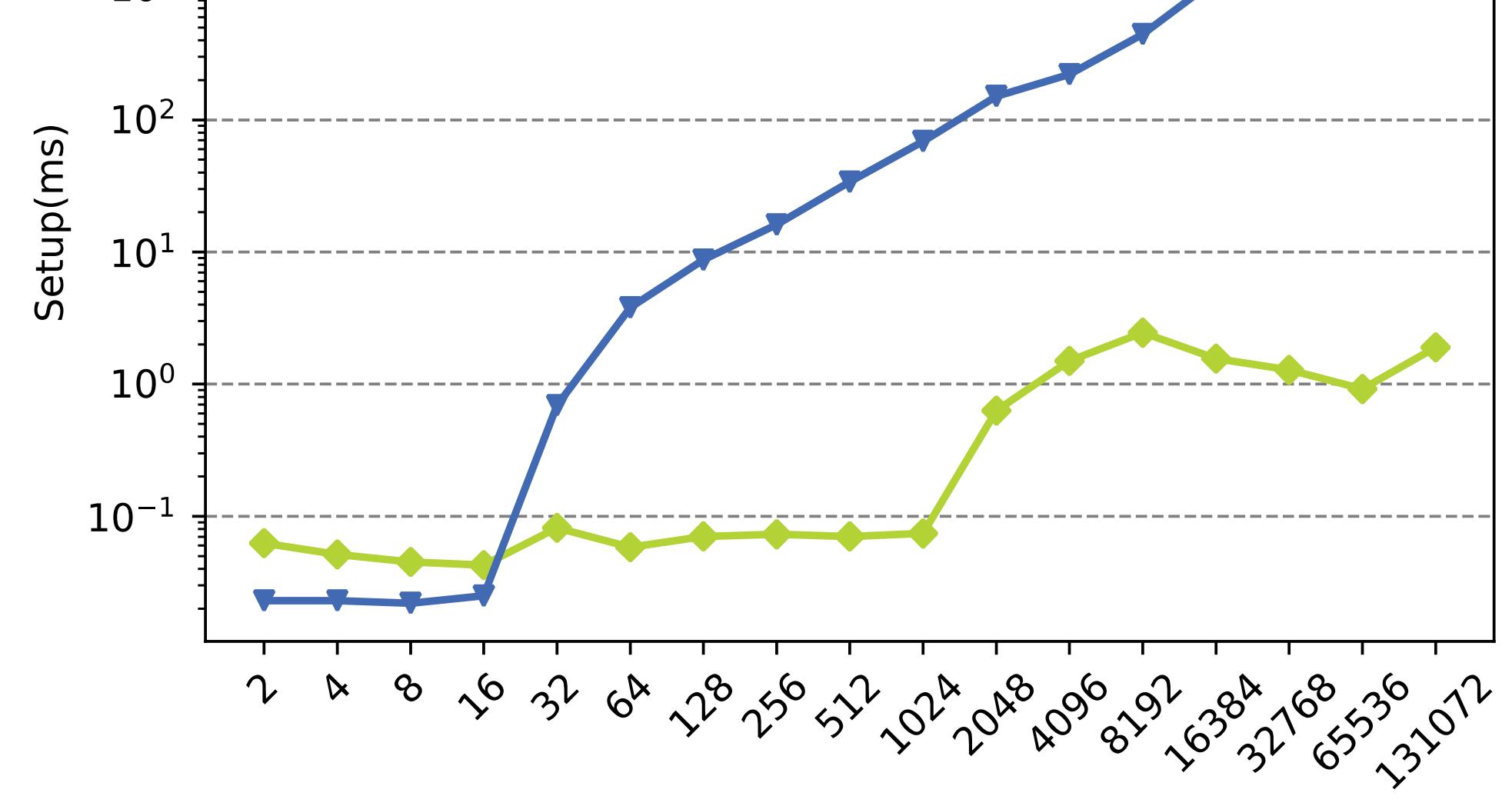
- 提出基于计算图模型的自动调优框架
 - ◆ 利用计算图表达分解过程，相比递归形式表达范围更加广泛
 - ◆ 结合拓扑排序、图染色、算子融合等技术实现自动优化与自动并行
- 基于计算规模的代价模型
 - ◆ 针对L1缓存提供静态方案
 - ◆ 利用各算子计算时间对分解过程、机器建模
 - ◆ 利用整数线性规划求解器求解分解方案



- 改进的FFT分解算法
 - ◆ 调整递归分解中向量化的方向，通过双向递归实现算子的高效向量化
- 并行任务划分算法
 - ◆ 按线程数量进行规模分解
 - ◆ 对计算图进行拓扑排序实现并行任务的自动划分与负载均衡



➤ 在申威处理器平台的性能表现



➤ 在申威处理器平台的搜索时间