

AI 驱动 软件研发 全面进入数字化时代

中国·北京 08.18-19

AI+
software
Development
Digital
summit



基于机器学习的自动化AI芯片编译技术

李建军 地平线

科技生态圈峰会 + 深度研习 ——1000+ 技术团队的选择



2023K+
全球软件研发行业创新峰会
上海站

会议时间 | 06.09-10



2023K+
全球软件研发行业创新峰会
北京站

会议时间 | 07.21-22



2024K+
全球软件研发行业创新峰会
深圳站

会议时间 | 05.17-18



K+峰会详情



会议时间 | 08.18-19
AiDD AI+软件研发数字峰会
北京站



会议时间 | 11.17-18
AiDD AI+软件研发数字峰会
深圳站



AiDD峰会详情



演讲嘉宾



李建军

地平线 编译器研发部负责人

2012年博士毕业于中国科学院计算技术研究所，2012年至2016年在计算所任副研究员，主要研究方向为动静态程序分析、编译优化、软件安全等。2016年底加入地平线，负责AI芯片的编译器及Runtime相关的研发工作，并直接参与AI芯片的软硬件协同设计开发。在AI芯片编译优化，基于机器学习的编译优化技术等方面做了大量的研究和实践。在CGO, ICSE, ASE, TACO, TPDS等国际会议和期刊上发表论文10余篇，AI芯片、编译器相关授权专利30余项。

目录

CONTENTS

1. AI算法和应用的演进
2. AI芯片架构
3. AI编译优化
4. AI时代的软硬件协同设计

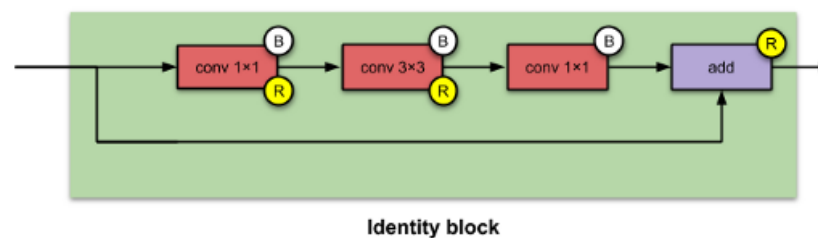
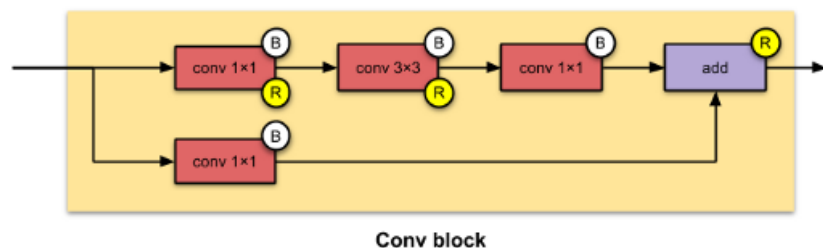
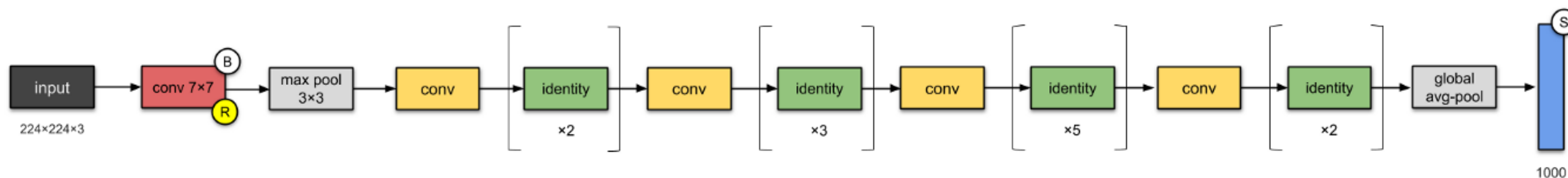
PART 01

AI算法和应用的演进

- 从ResNet 到 Transformer
- 从CPU+BPU到复杂异构
- Software 1.0 -> 2.0

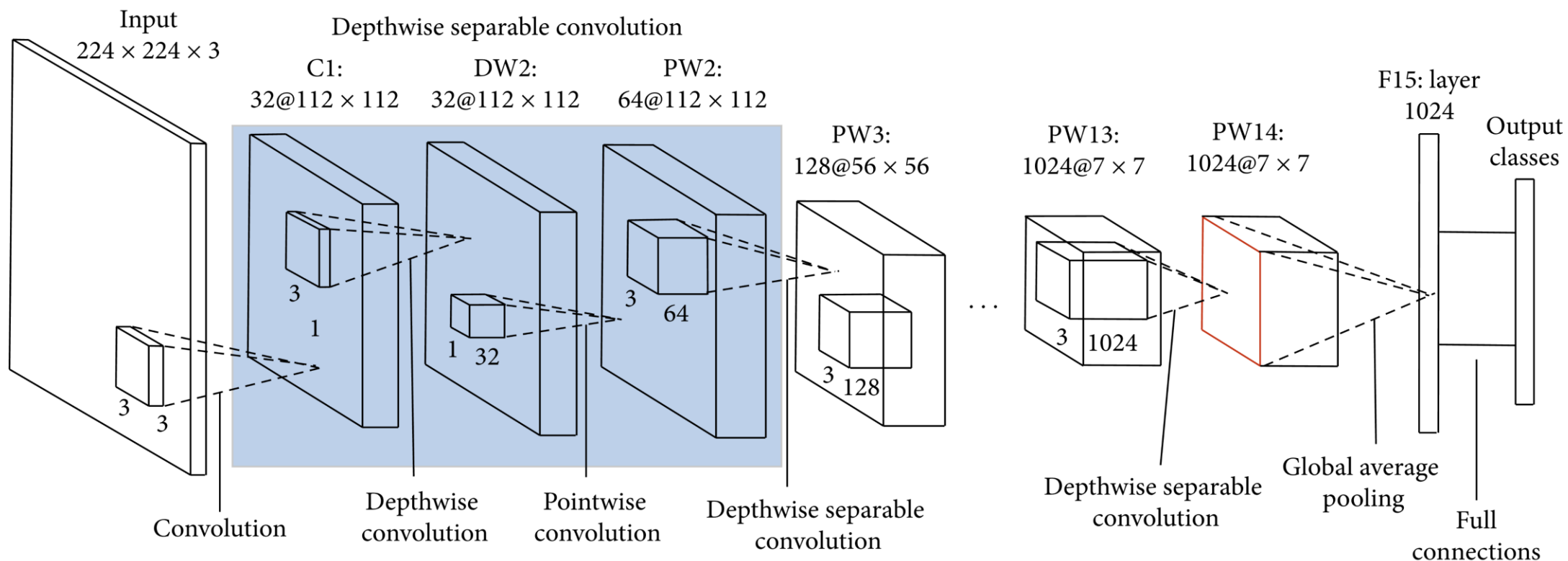
AI芯片上的神经网络算法

ResNet



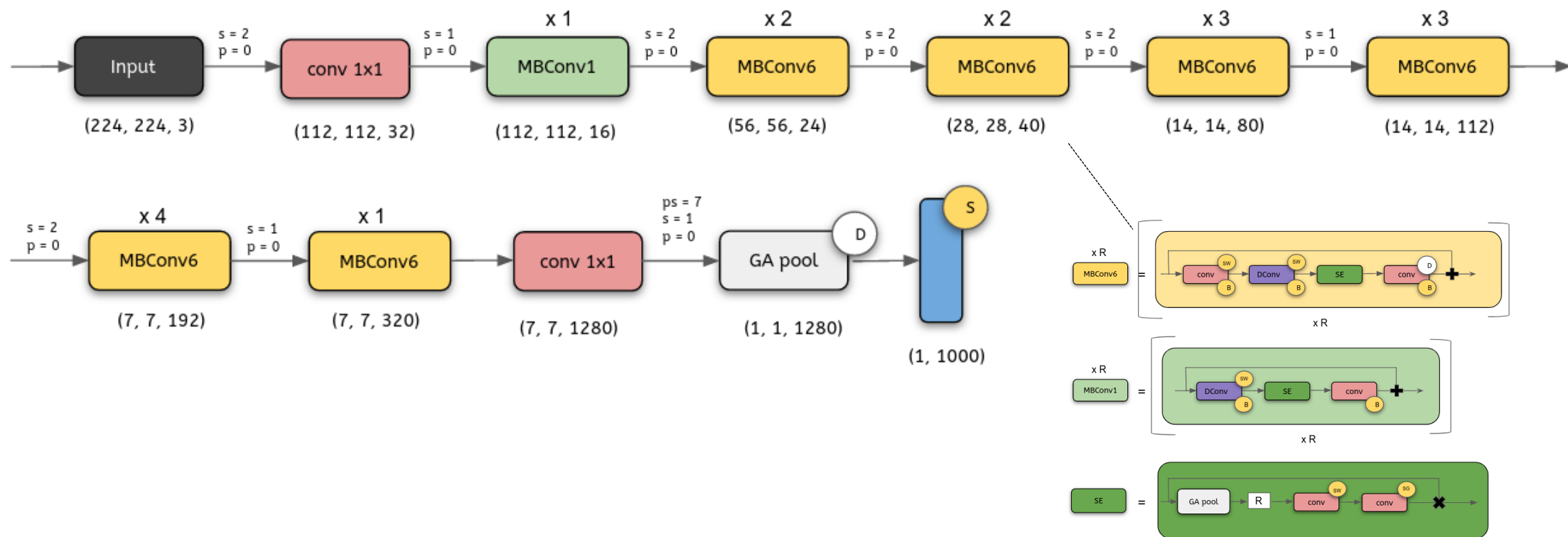
AI芯片上的神经网络算法

MobileNet



AI芯片上的神经网络算法

EfficientNet



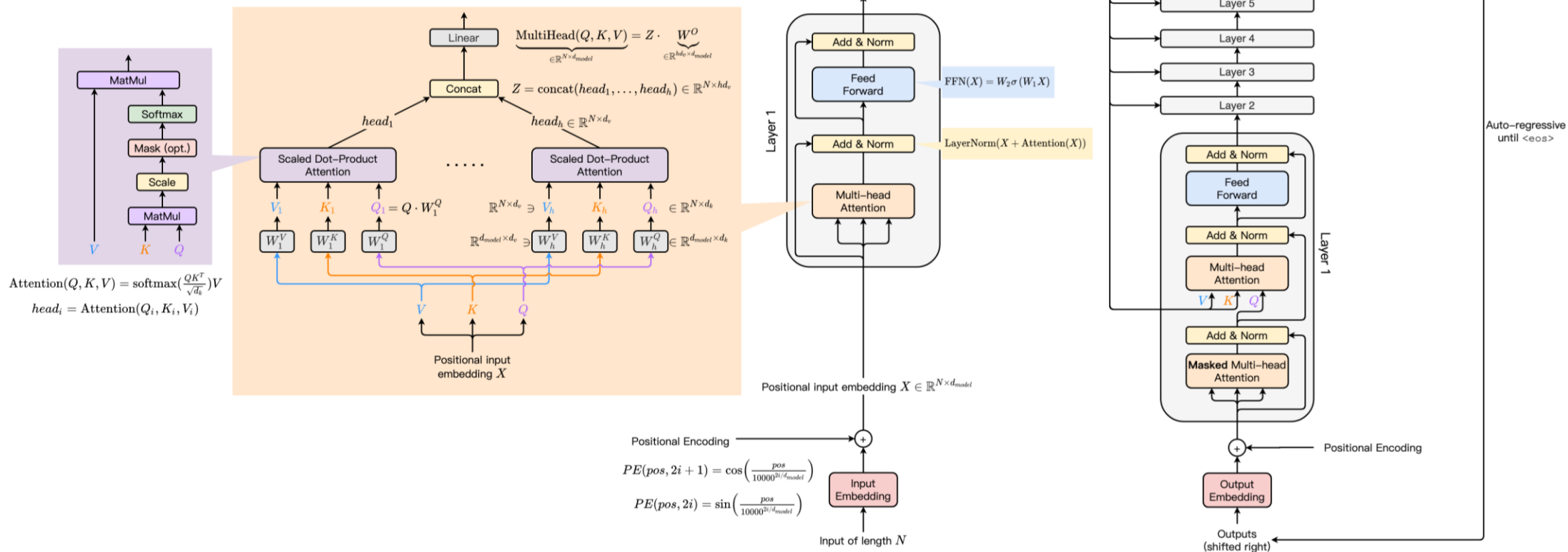
Made by 3outeille

AI芯片上的神经网络算法

Transformer

$d_{model} = 512$: length of the longest sentence in our training dataset
 $h = 8$: number of heads
 d_k : dimension of query Q and key K
 d_v : dimension of value V
 $d_k = d_v = d_{model}/h = 64$
 N : length of input

You can stack the encoder layer n times to further encode the information, where each layer has the opportunity to learn different attention representations, therefore potentially boosting the predictive power of the transformer network



▶ AI芯片上的神经网络算法

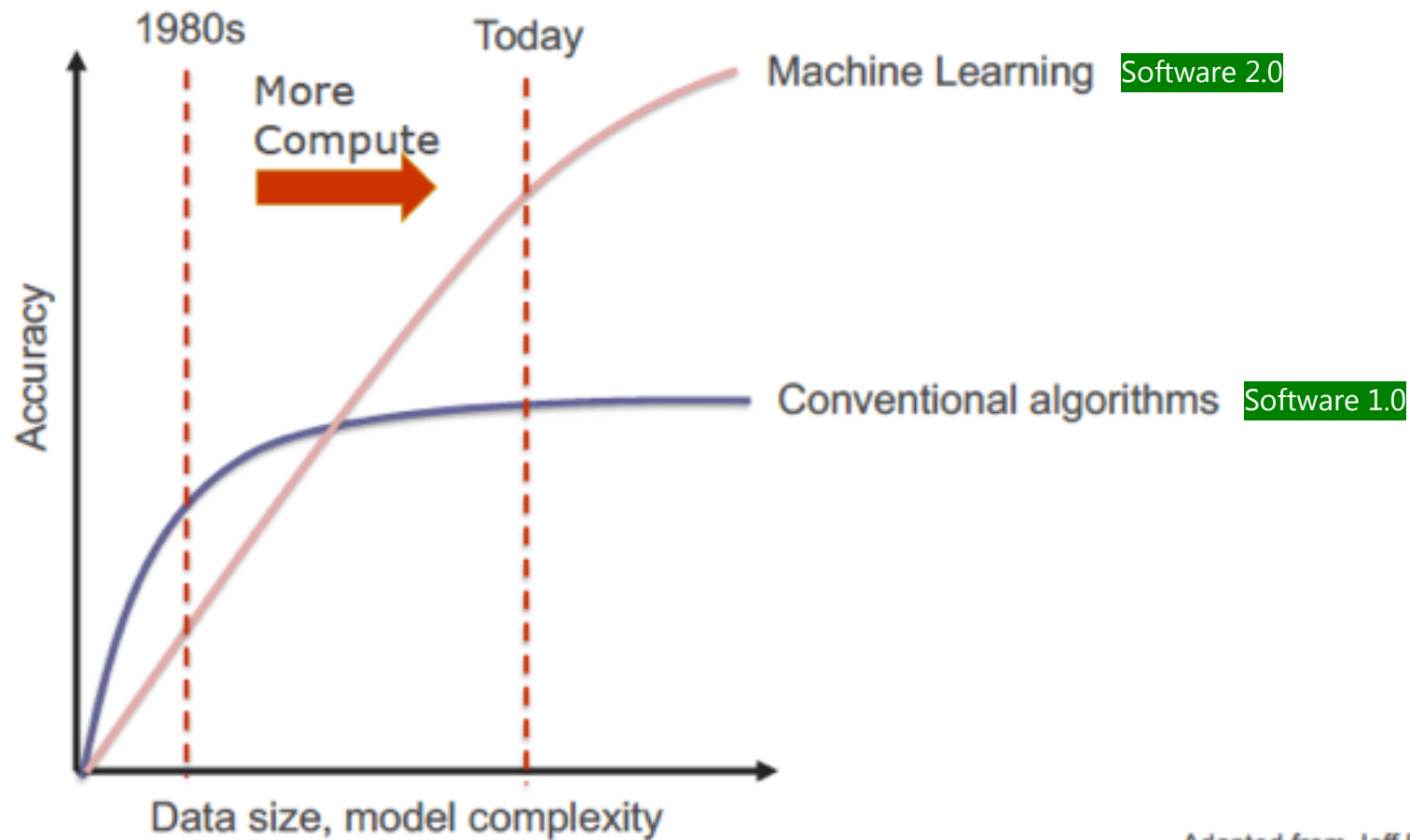
- AI 模型越来越复杂
 - Convolution -> Conv/Matmul/Reshape/Layernorm/Transpose/...
 - 模型越来越大
 - 模型结构越来越复杂
- 细粒度异构计算能力的需求
 - 异构计算部件之间的高速数据共享通路
 - 灵活的控制、调度并行

▶ Transformer模型

- 不只是Tensor计算
 - Vector: Elementwise
 - Softmax, LayerNorm
 - Reshape, Transpose
- 计算访存比
 - GPT大语言模型：一个让人崩溃的计算访存比
 - GPGPU: 将大量连续的计算（主要是vector）融合成一个大的kernel
 - NPU: 异构计算，利用SRAM等片内高速缓存
- Reshape / Transpose
 - 图优化
 - 很难完全消除

► AI芯片上的应用软件

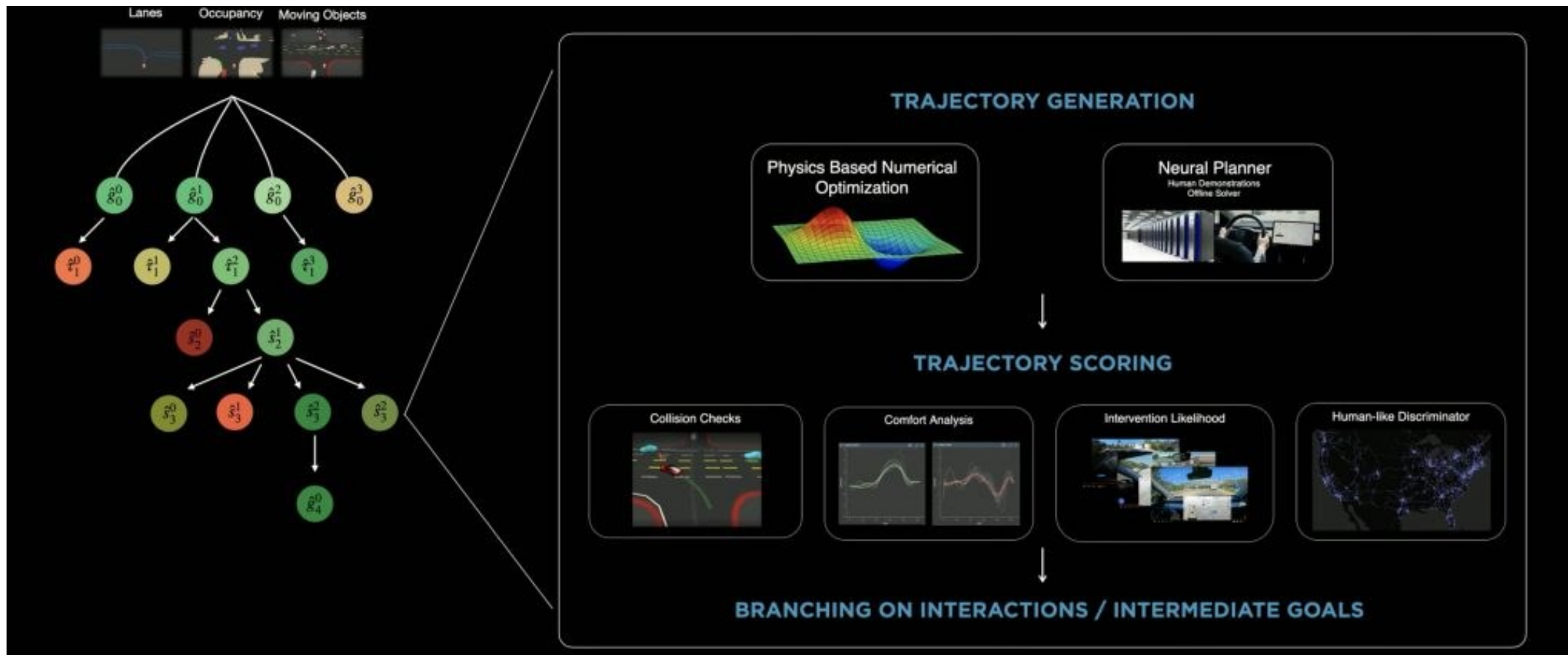
Software 1.0 -> 2.0



Adapted from Jeff Dean
HotChips 2017

AI驱动软件研发全面进入数字化时代

AI芯片上的应用软件



From Tesla AI Day

▶ AI芯片上的应用软件

- 基于规则的代码 -> 数据驱动的神经网络
 - 规划控制
 - 模型前后处理
 - 模型串联
 - ...
- 异构计算的需求
 - 高效执行整个应用流程
 - 模型LTO优化 -> Task Graph Compiler & Runtime

PART 02

AI芯片架构

Tensor -> Tensor + Vector + Scalar

地平线车载智能芯片架构

Horizon Journey3 (BPU® Bernoulli)

Instruction set with complete toolchain:

- Multi-Instruction Multi-data processor
- Compiler, Linker, Simulator

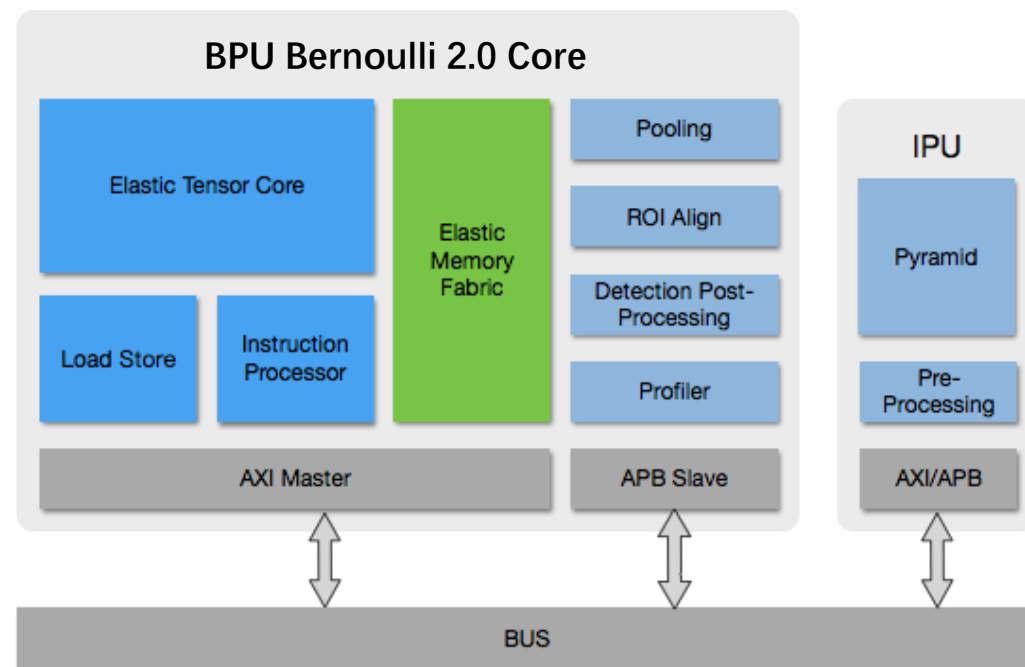
Efficient performance

- Achieving much **higher FPS** than competitors on efficient networks: MobileNet v1/v2, EfficientNet lite

Optimized DDR bandwidth

- High SRAM utilization
- Co-optimization from hardware and software
- **Saving 50~80% DDR bandwidth** relative to a competitor

Performance profiler and debug system



地平线车载智能芯片架构

Horizon Journey5 (BPU® Bayes)

High Performance

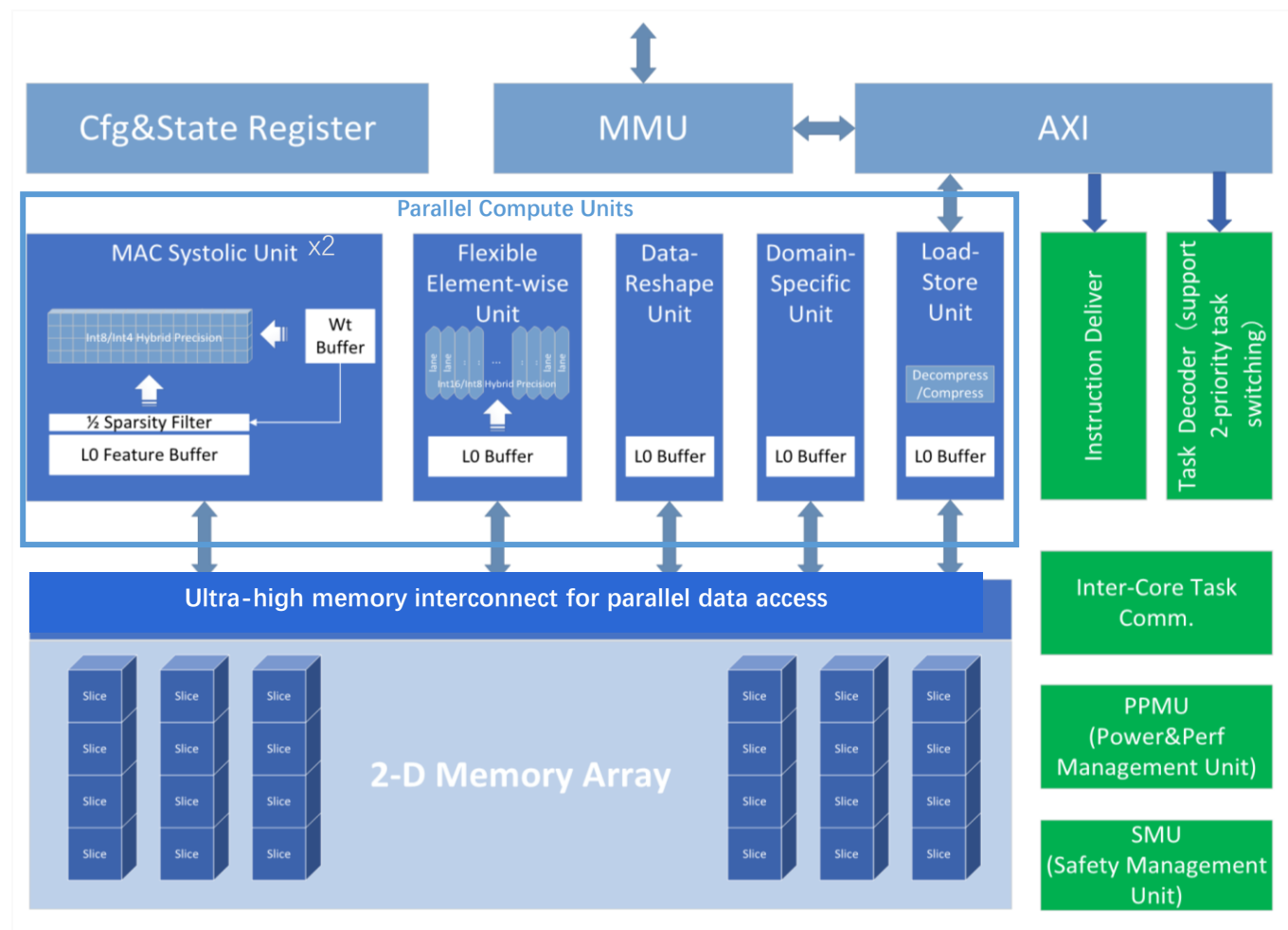
- Achieving higher FPS than a competitor (Orin-X) on object detection
- About 10 ~ 25x performance boost compare with BPU Bernoulli 2.0 on typical models

New design for MAC array and memory hierarchy

- Reduce the requirement of DDR bandwidth and increase efficiency

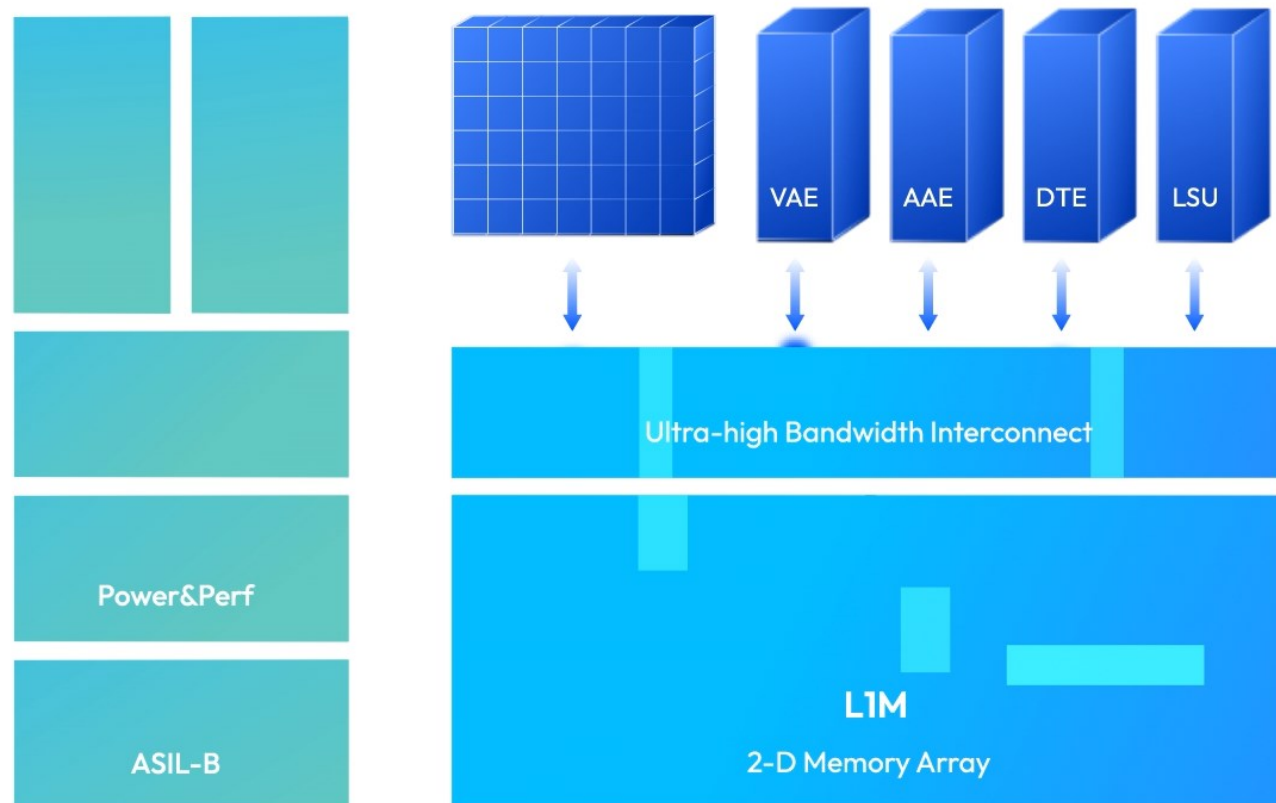
Improved flexibility at hardware level

- Flexible element-wise unit
- Data reshape unit
- Domain Specific Unit, Pooling, Warping, Resizing



地平线车载智能芯片架构

Horizon Journey5 (BPU® Bayes)



地平线车载智能芯片架构

Next Generation -- BPU® Nash



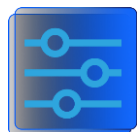
三级存储架构

极致优化大参数下的带宽瓶颈
核间高效协同



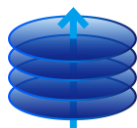
多脉动立方加速引擎

灵活的引擎间数据流动
高效能、低宽带占用



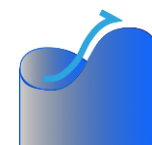
紧耦合异构计算

高效加速不同类型数据处理



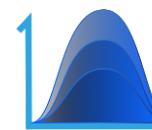
虚拟化Virtualization

透明式提升多任务并行处理能力



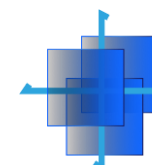
数据变换引擎

灵活支持Transformer细小算子



浮点向量加速单元

通用、灵活
关键算子精度需求



多向数据流动

核内、核间、片间高效灵活的数据流动
动态调度，灵活调优



数据驱动功耗优化

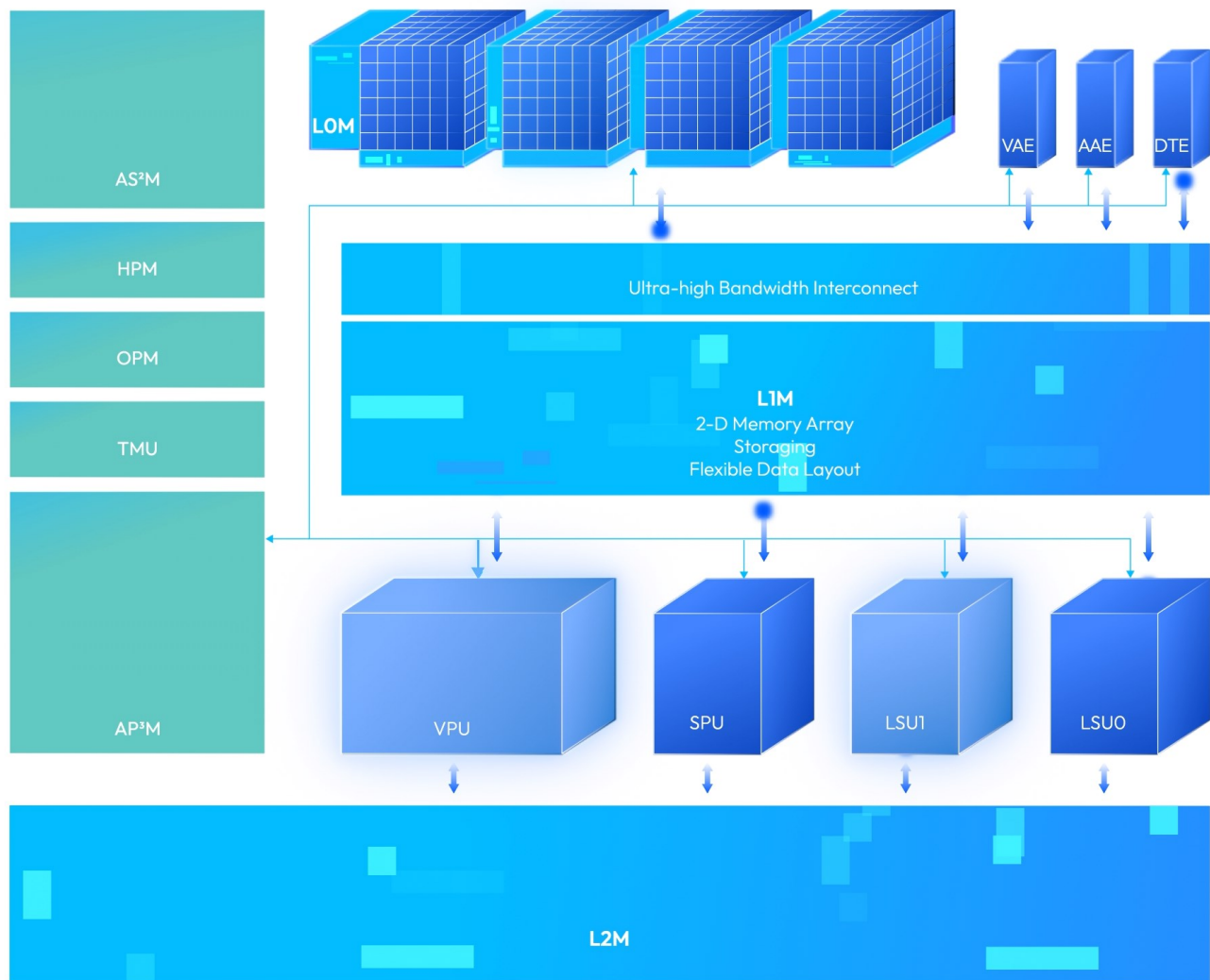
针对神经网络数据动态范围特性，
降低功耗30%

AI驱动软件研发全面进入数字化时代

AI+ 软件研发数字峰会
AI+ software Development Digital summit

地平线车载智能芯片架构

Next Generation -- BPU® Nash



AI驱动软件研发全面进入数字化时代

▶ AI芯片架构的演进

Tensor + Vector + Scalar

- 异构计算架构
 - 将绝大部分模型的计算在芯片内完成
 - 高速数据共享通路，异构调度协作
 - Tensor Core
 - Convolution, Matmul, 以及少部分其它OP
 - Vector Core
 - Elementwise, 各种激活函数, Attention结构等
 - Scalar Core
 - 控制逻辑, 非Tensor/Vector的OP, 用户自定义OP等
 - 特定计算部件
 - 高频激活函数, 影响关键性能的复杂OP
 - NoC: 高速互联

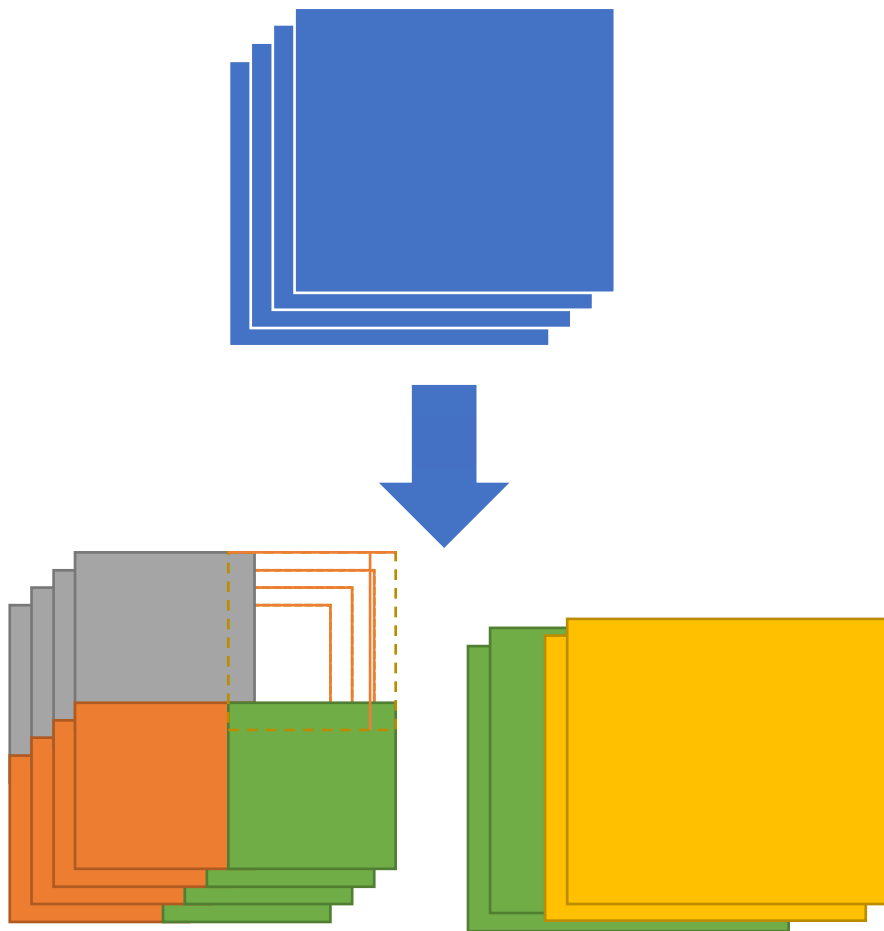
PART 03

AI编译优化

规则驱动 -> 数据驱动

▶ AI编译优化

Tiling



编译器优化

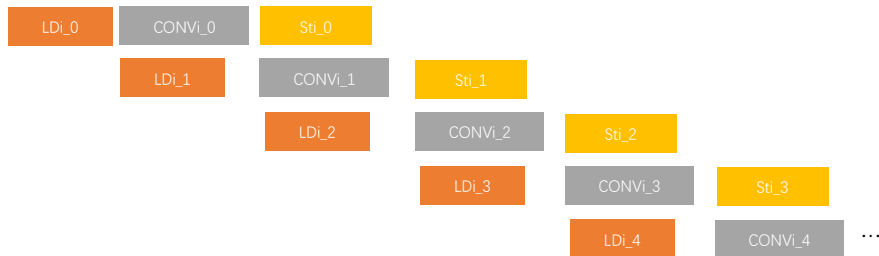
1

计算自动拆分

通过对特征图和卷积kernel的计算拆分，编译器可以用更小的粒度描述计算，避免引入不必要的依赖，提升数据并行性，创造潜在的调度机会。

AI编译优化

Scheduler

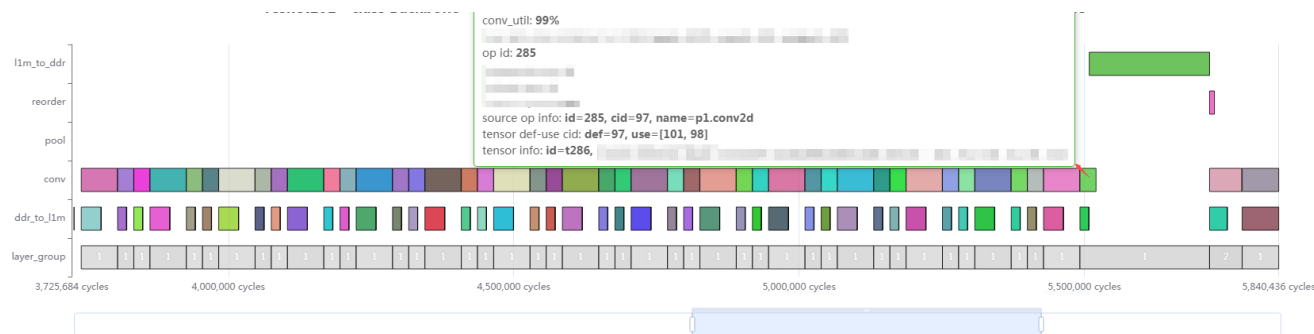


编译器优化

2

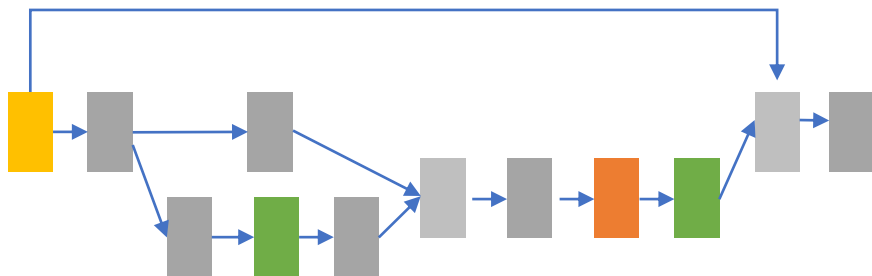
指令自动并行调度

编译器实现的针对张量的软件流水线调度优化，缓解指令间依赖，提升了指令并行度，通过将多个拆分后的动作拼接成无依赖的软件流水线步骤，辅以面向张量计算的指令调度算法，可以充分的利用芯片中的访存和计算单元，并行执行。

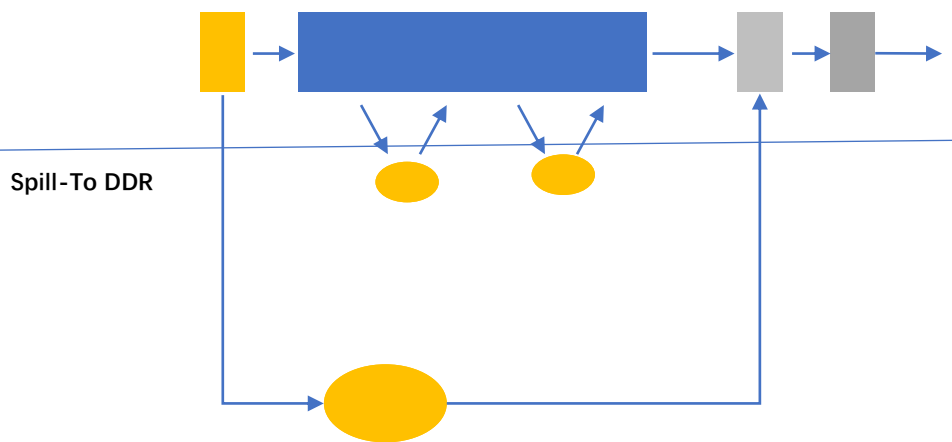


AI编译优化

Fusion



On-Chip Computation



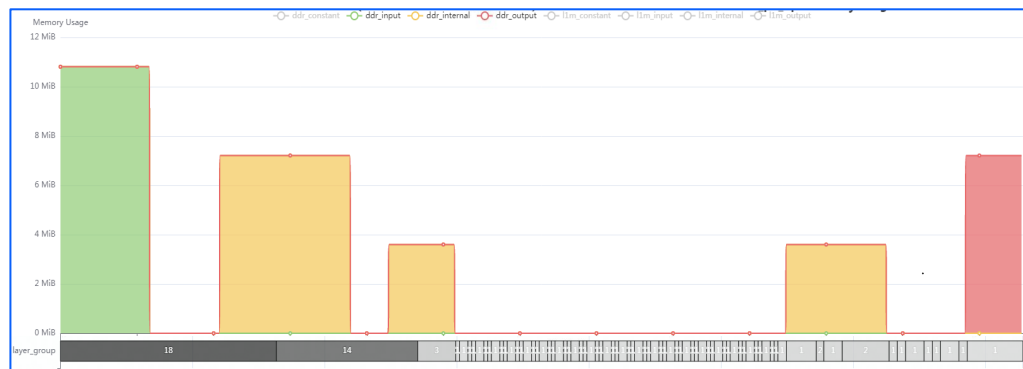
Spill-To DDR

编译器优化

3

计算融合

编译器自动地通过全局分析整个神经网络，配合大算子计算拆分和指令并行优化，进行全局的计算拆分和融合，减少访存操作。

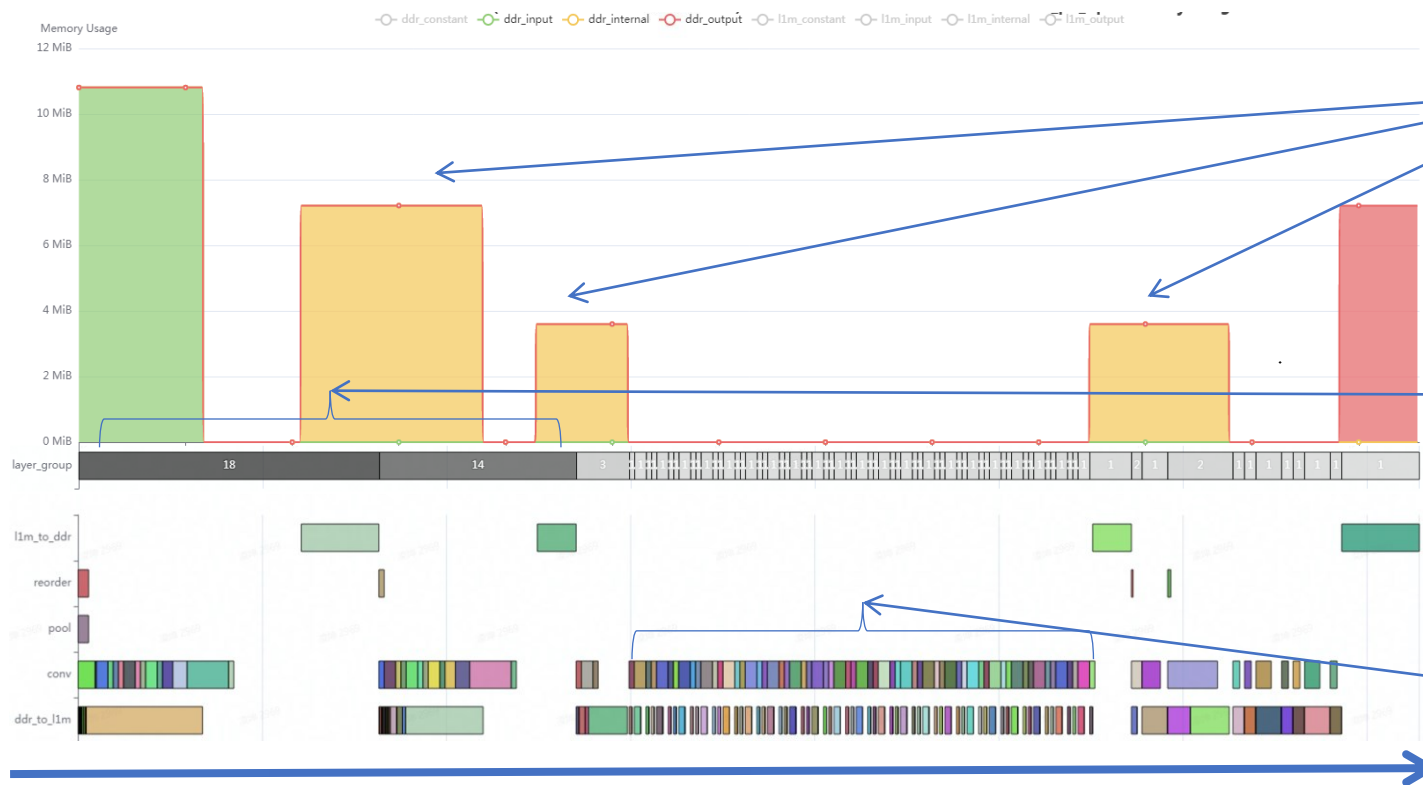


AI驱动软件研发全面进入数字化时代

AI+ 软件研发数字峰会
AI+ software Development Digital summit

AI编译优化

Tiling + Fusion + Schedule



计算融合
模型运行过程中仅3次DDR访问

计算拆分
通过Tiling将中间数据缓存在片内，省带宽

计算并行与指令调度
较高的CONV计算阵列利用率

ResNet101 with 720P Input Inference TimeLine

▶ AI模型编译

模型&芯片复杂度 -> 搜索时间变长

- 传统的AI模型编译优化: 本质上是一个基于规则的优化搜索问题。
 - DP + 启发式搜索的方法
 - 基于硬件平台的Cost Model
- 模型&芯片复杂度 -> 编译时间
 - 模型输入变大: Fusion & Tiling & Schedule的选择空间
 - 模型结构复杂: 数据对齐, 调度优化
 - 芯片异构复杂度: 调度优化的搜索空间
- 搜索空间剪枝
 - 不同的优化级别O2, O3, Ofast
 - 在探索空间和编译速度之间均衡



► 编译器的进化

规则实现 -> 数据驱动(监督学习) -> 强化学习

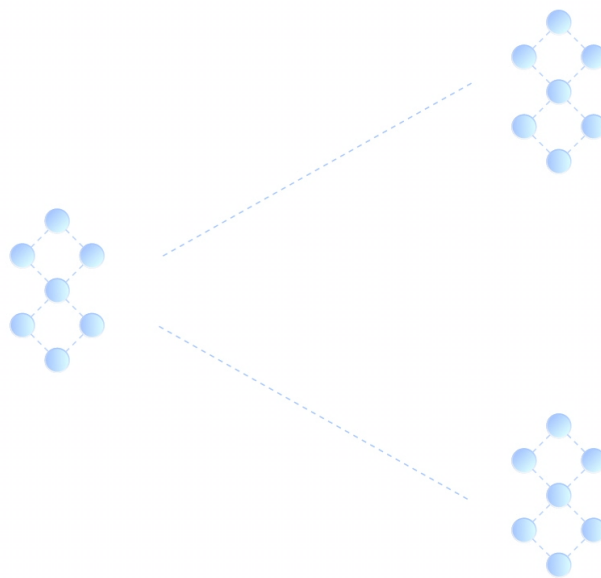
► 编译器的进化

规则实现 -> 数据驱动(监督学习) -> 强化学习

Handwritten Rules
Dynamic Programming

Policy Network
Dynamic Programming

Policy Network Value Network
Monte-Carlo Tree Search



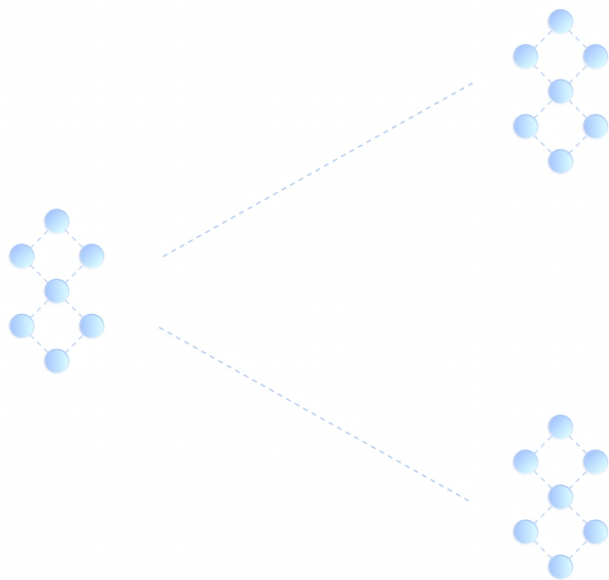
► 编译器的进化

规则实现 -> 数据驱动(监督学习) -> 强化学习

Heuristic
Dynamic Programming

Policy Network
Dynamic Programming

Policy Network Value Network
Monte-Carlo Tree Search



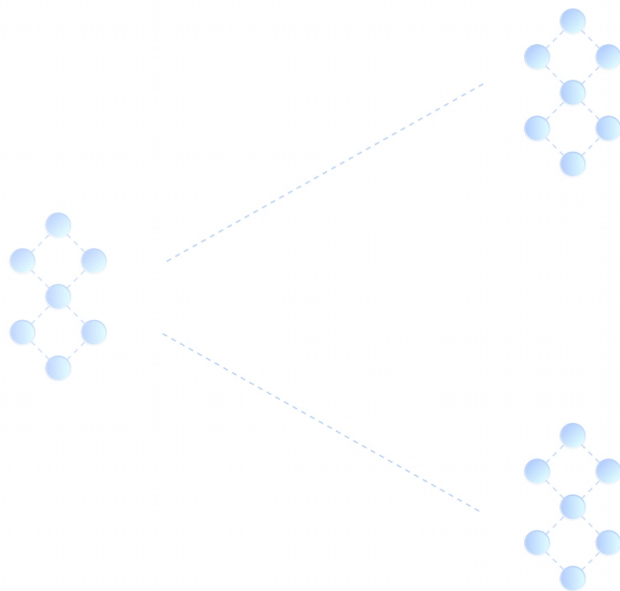
► 编译器的进化

规则实现 -> 数据驱动(监督学习) -> 强化学习

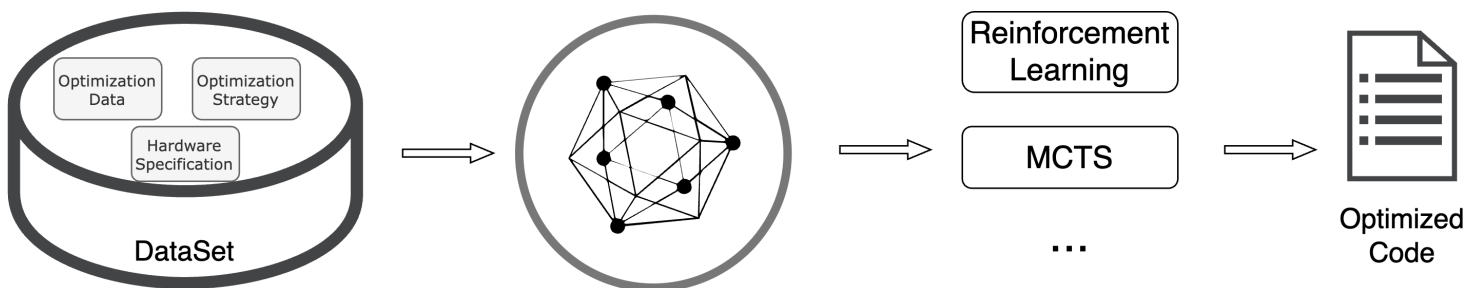
Heuristic
Dynamic Programming

Policy Network
Dynamic Programming

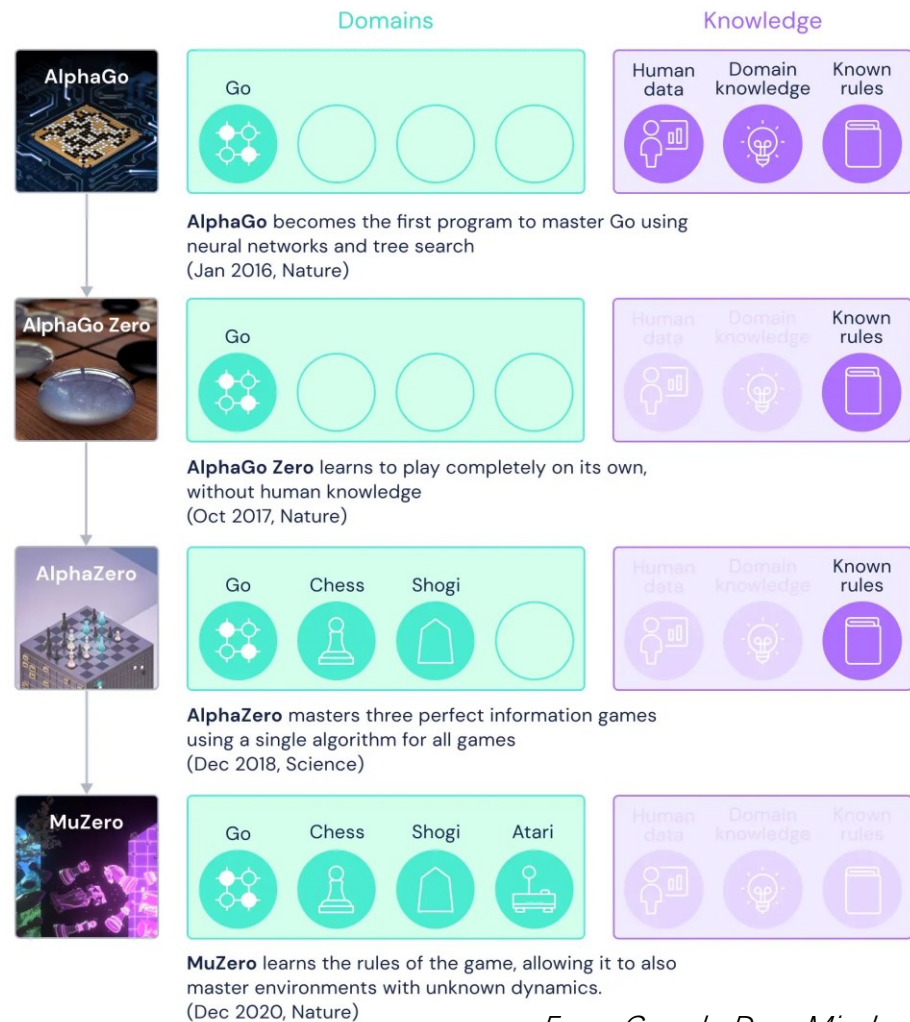
Policy Network Value Network
Monte-Carlo Tree Search



► 基于ML的编译优化

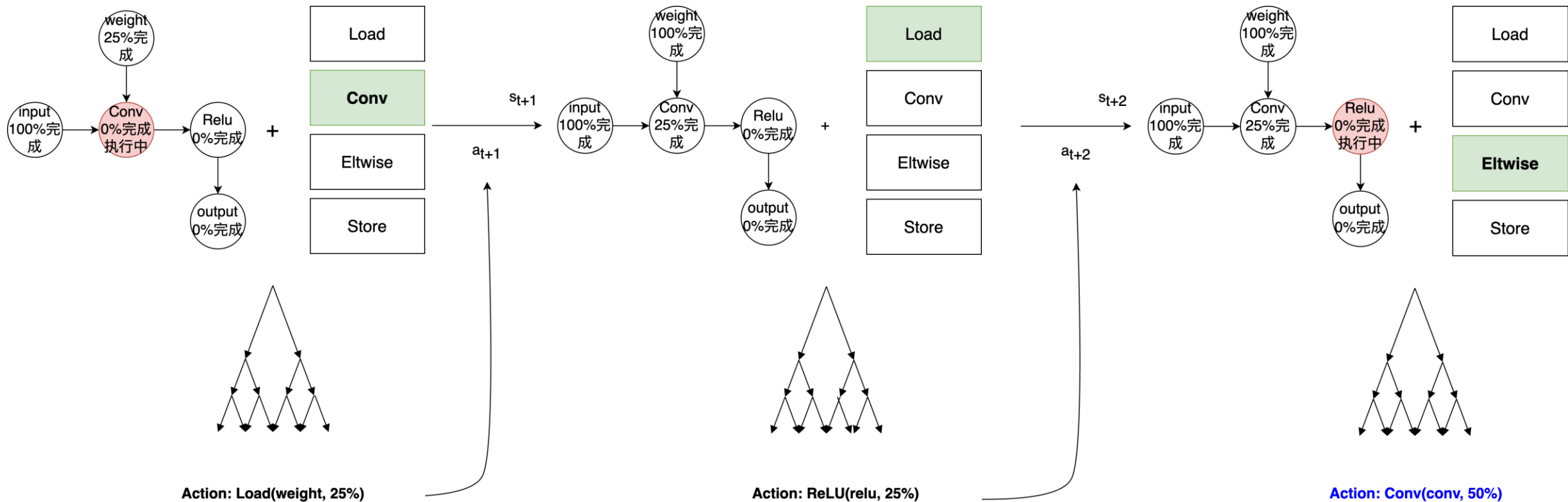


- History Optimization Data
- Compiler Optimization Strategy
- Hardware Specification



From Google DeepMind

▶ 基于RL的编译优化



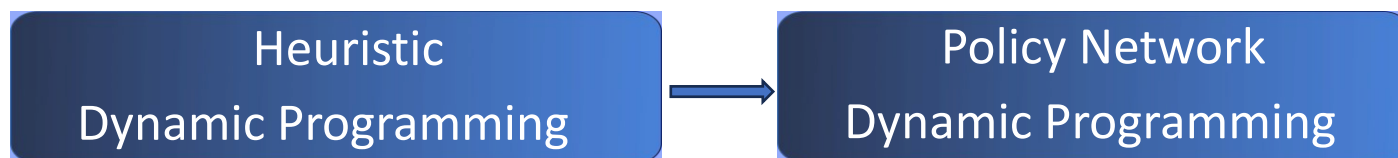
► 编译器的进化

问题规模

Name	搜索空间	动作空间
AlphaZero (Chess)	10^{120}	242
AlphaZero (围棋)	10^{700}	361
AlphaStar (星际2)	$>10^{20000}$	10^{26}
AlphaDev (排序)	10^{200}	300-1000
HBDK codegen	10^{2000}	10^3-10^5

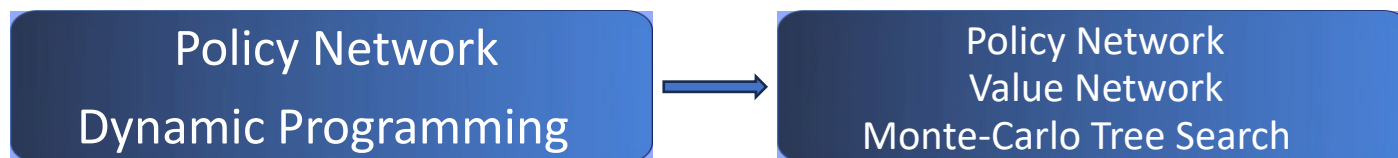
► 编译器的进化

规则实现 -> 数据驱动(监督学习) -> 强化学习



编译速度提升

10x



预期模型推理性能提升 > 20%

PART 04

AI时代的软硬件协同

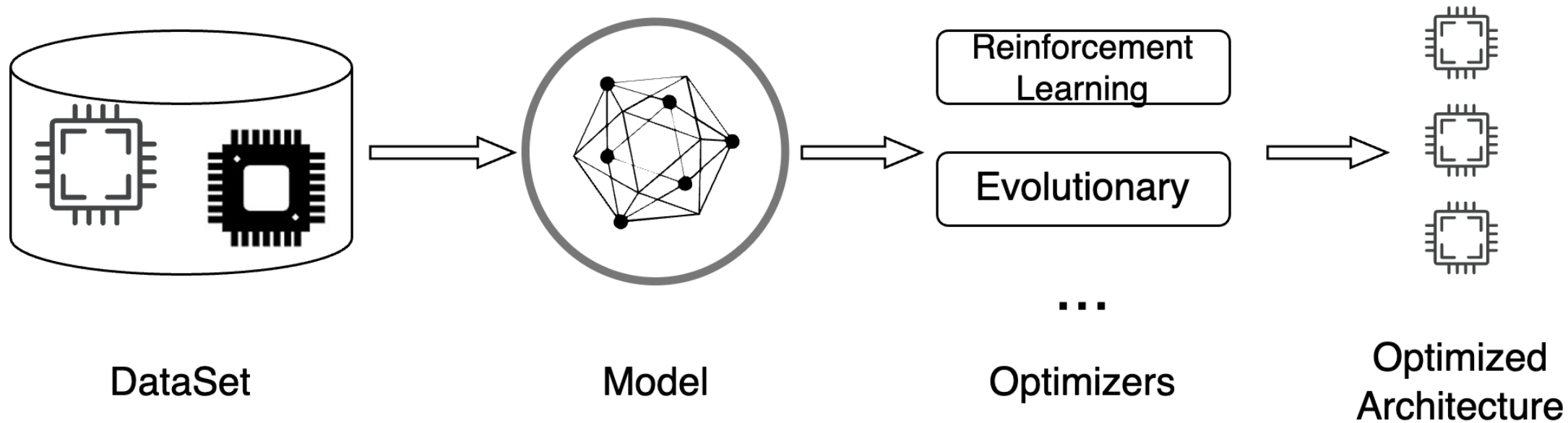
数据驱动的软硬件协同设计



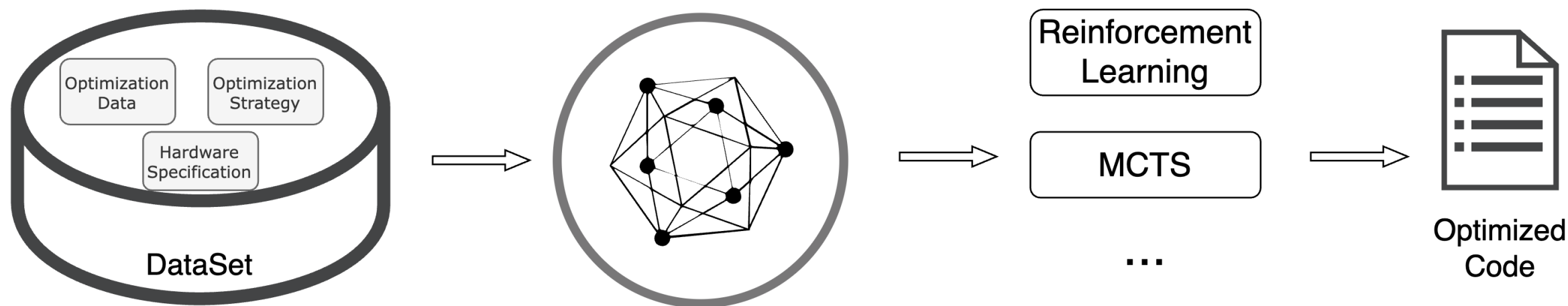
AI驱动软件研发全面进入数字化时代

AI+ 软件研发数字峰会
AI+ software Development Digital summit

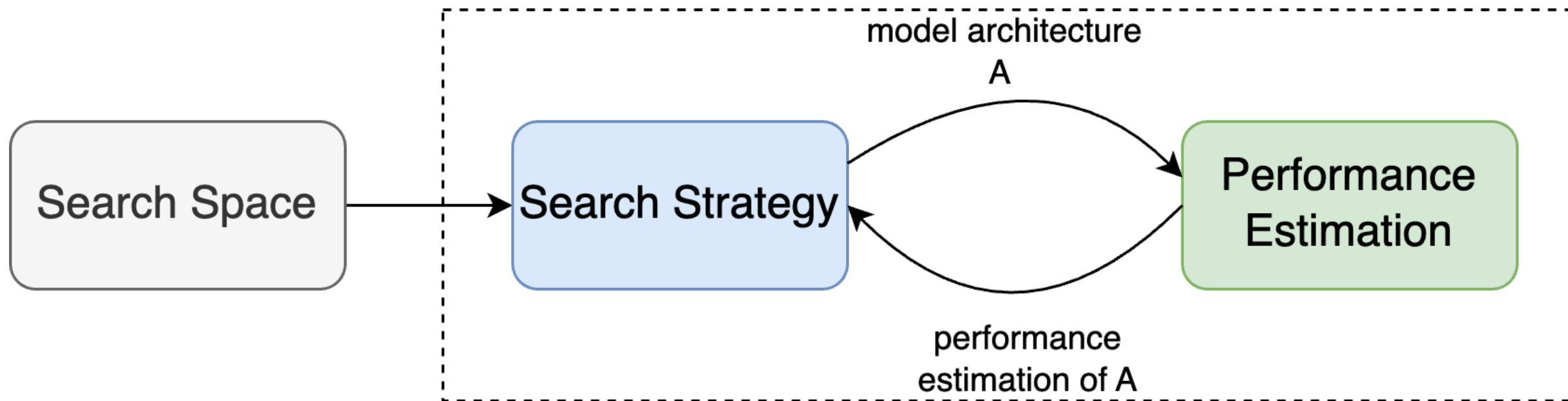
▶ 基于ML的BPU架构探索



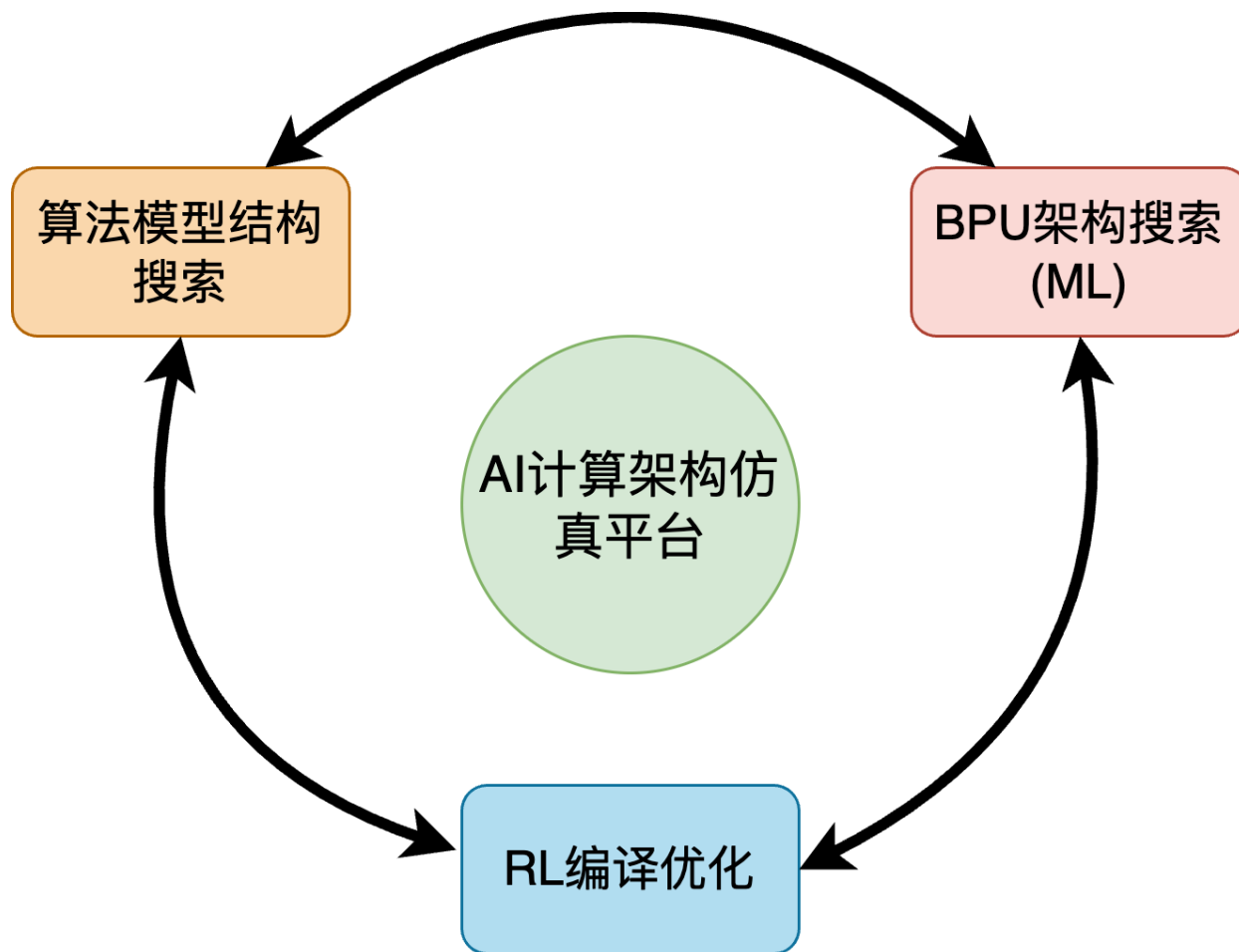
► 基于RL的编译优化



► NAS: 特定芯片上最优的模型结构



► 软硬件协同、数据驱动的AI系统设计



感谢聆听

