推理引擎 - 模型转换与优化

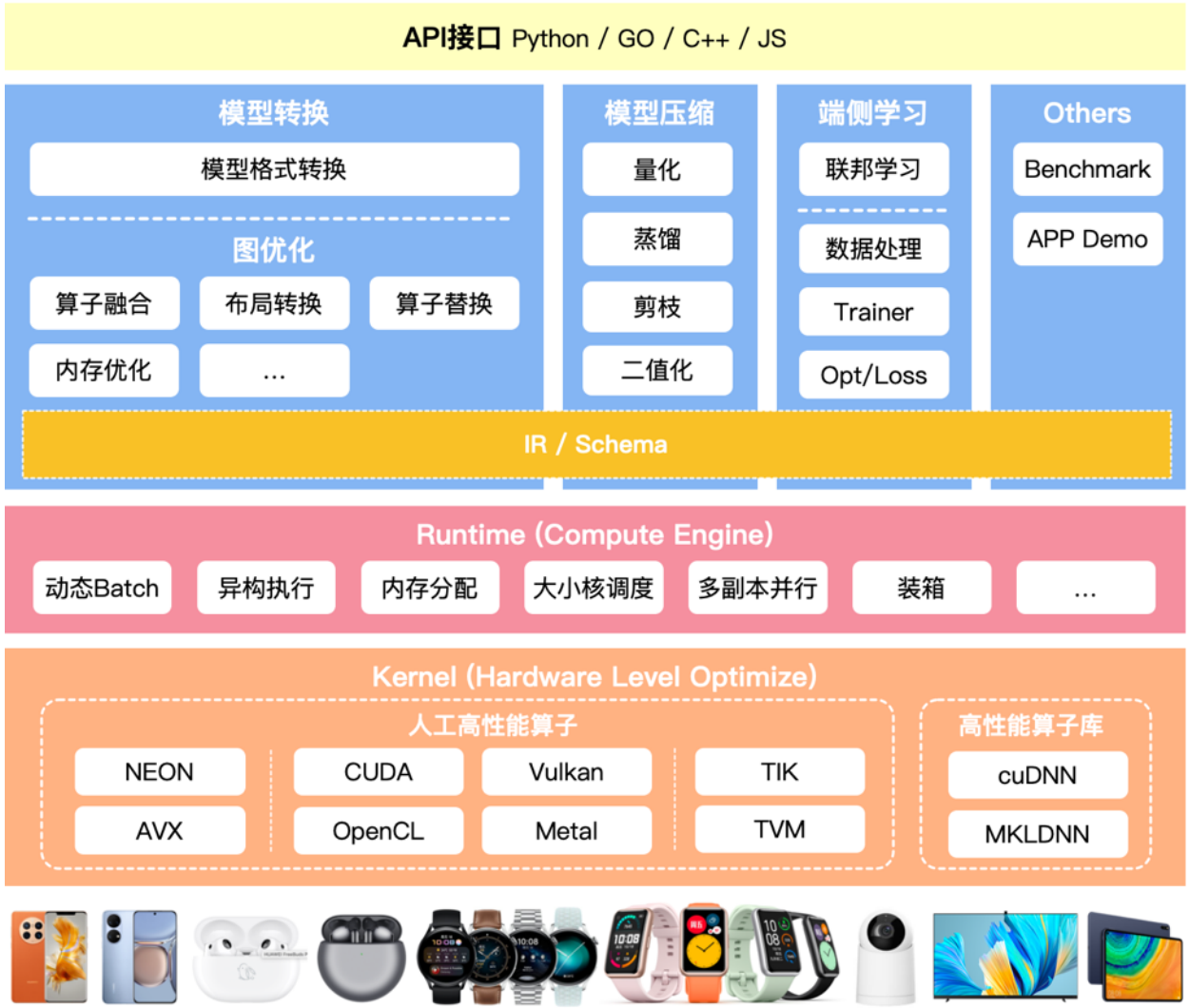
架构与流程

ZOMI

Talk Overview

1. **推理系统介绍**
   - 推理系统架构
   - 推理引擎架构
2. **模型小型化**
   - CNN小型化结构
   - Transform小型化结构
3. **离线优化压缩**
   - 低比特量化
   - 模型剪枝
   - 知识蒸馏
4. **模型转换与优化**
   - 架构与流程
   - 模型格式转换
   - 模型离线优化
5. **Runtime与在线优化**
   - 动态batch
   - bin Packing
   - 多副本并行

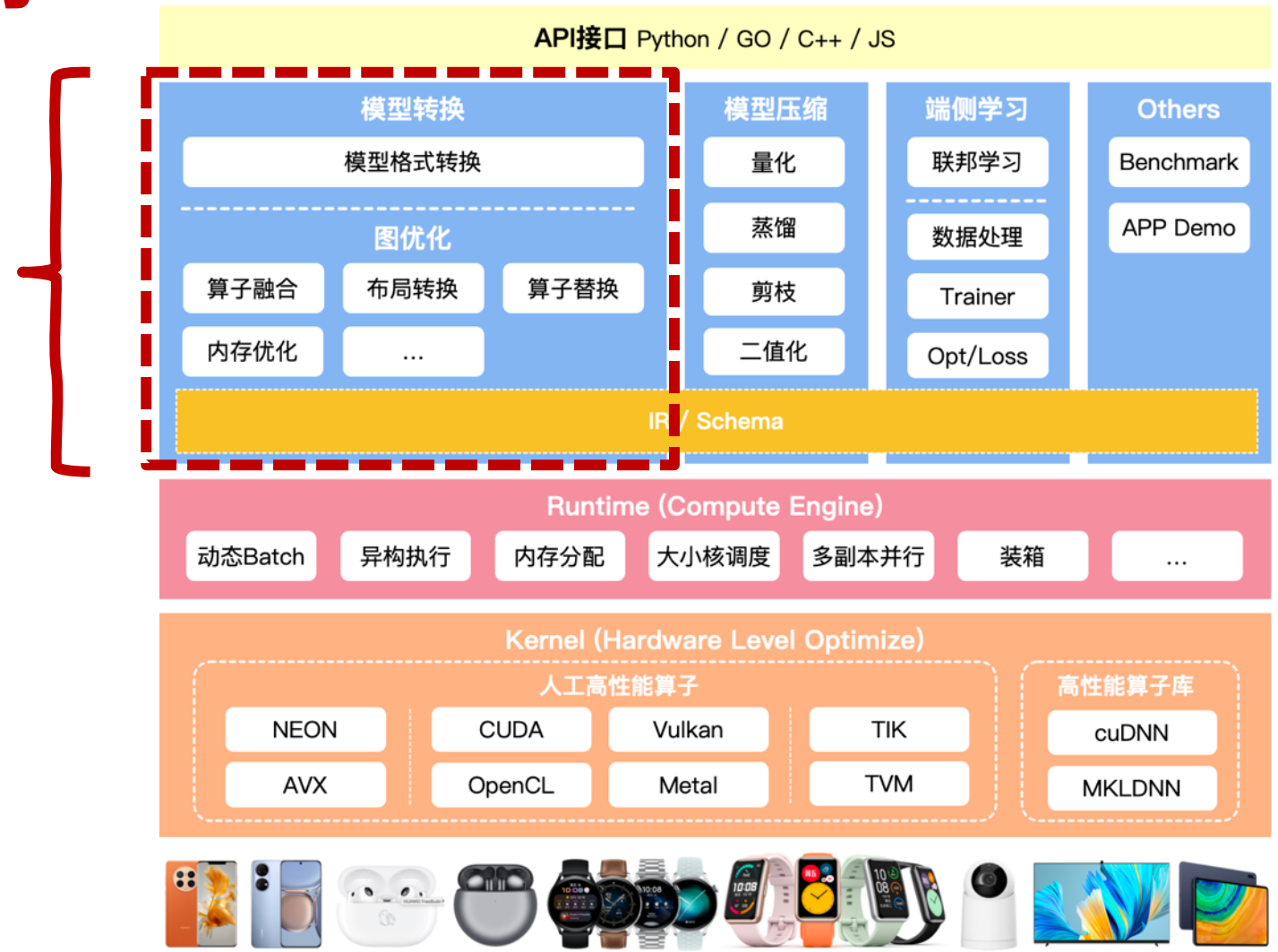
推理引擎架构



推理引擎架构

模型转换工具

- 模型格式转换
- 计算图优化



转换模块

挑战与目标

Converter Challenge 转换模块挑战

1. AI 模型本身包含众多算子，推理引擎需要用有限算子实现不同框架 AI 模型所需要的算子。
2. 支持不同框架 Tensorflow、PyTorch、MindSpore、ONNX 等主流模型文件格式。
3. 支持 CNN / RNN / GAN / Transformer 等主流网络结构。
4. 支持多输入多输出，任意维度输入输出，支持动态输入，支持带控制流的模型。

Converter Challenge 转换模块挑战

1. AI 模型本身包含众多算子：
   - 不同 AI 训练框架的算子重合度高，但不完全一样
   - 推理引擎需要用有限算子实现不同框架 AI 模型所需要的算子

框架	导出方式	导出成功率	算子数（不完全统计）	冗余度
Caffe	Caffe	高	52	低
Tensorflow	1.X	高	1566	高
	Tflite	中	141	低
Pytorch	Self	中	1200+	高
	Onnx	中	165	低
	TorchScripts	高	566	高

Converter Challenge 转换模块挑战

1. AI 模型本身包含众多算子，推理引擎需要用有限算子实现不同 AI 框架所需要的算子。
2. 支持不同框架 Tensorflow、PyTorch、MindSpore、ONNX 等主流模型文件格式。
   - AI 训练框架随版本变迁会有不同的导出格式
   - AI 训练框架随版本变迁有大量的算子新增与修改

Converter Challenge 转换模块挑战

1. AI 模型本身包含众多算子，推理引擎需要用有限算子实现不同 AI 框架所需要的算子。
2. 支持不同框架（TensorFlow、PyTorch、MindSpore、ONNX 等主流模型文件格式。
3. 支持 CNN、RNN、Transformer 等主流网络结构。
4. 支持多输入多输出，支持动态输入，支持带控制流的模型。

拥有自己的算子定义和格式

对接不同AI框架的算子层

Converter Challenge 转换模块挑战

1. AI 模型本身包含众多算子，推理引擎需要用有限算子实现不同 AI 框架所需要的算子。
2. 支持不同框架 Tensorflow、PyTorch、MindSpore、ONNX 等主流模型文件格式。
3. 支持 CNN / RNN / GAN / Transformer 等主流[illegible]终结构。
4. 支持多输入多输出，任意维度输入输出，[illegible]的模型。

自定义计算图IR

对接不同AI框架及其版本

Converter Challenge 转换模块挑战

1. AI 模型本身包含众多算子，推理引擎需要用有限算子实现不同 AI 框架所需要的算子。
2. 支持不同框架 Tensorflow、PyTorch、MindSpore、ONNX 等主流模型文件格式。
3. 支持 CNN / RNN / GAN / Transformer 等主流网络结构。
4. 支持多输入多[illegible]出，任意维度输入输出，支持动态输入，支持带控制流的模型。

丰富Demo和Benchmark

提供主流模型性能和功能基准

Converter Challenge 转换模块挑战

1. AI 模型本身包含众多算子，推理引擎需要用有限算子实现不同 AI 框架所需要的算子。
2. 支持不同框架 Tensorflow、PyTorch、MindSpore、ONNX 等主流模型文件格式。
3. 支持 CNN / RNN / GAN / Transformer 等主流网络结构。
4. 支持多输入多输出，任意维度输入输出，支持动态输入，支持带控制流的模型。

支持可扩展性和AI特性

对不同任务、大量集成测试验证

优化模块

挑战与目标

Optimizer Challenge 优化模块挑战

- **结构冗余**：深度学习网络模型结构中的无效计算节点、重复的计算子图、相同的结构模块，可以在保留相同计算图语义情况下无损去除的冗余类型；
- **精度冗余**：推理引擎数据单元是张量，一般为FP32浮点数，FP32表示的特征范围在某些场景存在冗余，可压缩到 FP16/INT8 甚至更低；数据中可能存大量0或者重复数据。
- **算法冗余**：算子或者Kernel层面的实现算法本身存在计算冗余，比如均值模糊的滑窗与拉普拉斯的滑窗实现方式相同。
- **读写冗余**：在一些计算场景重复读写内存，或者内存访问不连续导致不能充分利用硬件缓存，产生多余的内存传输。

Optimizer Challenge 优化模块挑战

- **结构冗余**：深度学习网络模型结构中的无效计算节点、重复的计算子图、相同的结构模块，可以在保留相同计算图语义情况下无损去除的冗余类型；
- **精度冗余**：推理引擎数据单元是张量，一般为FP32浮…场景存在冗余，可压缩到 FP16/INT8 甚至更低；数据中可能…
- **算法冗余**：算子或者Kernel层面的实现算法本身存在计算冗余，比如3D卷积模拟的滑窗与拉普拉斯的滑窗实现方式相同。
- **读写冗余**：在一些计算场景重复读写内存，或者内存访问不连续导致不能充分利用硬件缓存，产生多余的内存传输。

计算图优化

算子融合、算子替换、常量折叠

Optimizer Challenge 优化模块挑战

- **结构冗余**：深度学习网络模型结构中的无效计算节点、重复的计算子图、相同的结构模块，可以在保留相同计算图语义情况下无损去除的冗余类型；
- **精度冗余**：推理引擎数据单元是张量，一般为FP32浮点数，FP32表示的特征范围在某些场景存在冗余，可压缩到 FP16/INT8 甚至更低；数据中可能存大量0或者重复数据。
- **算法冗余**：算子Kernel层面的实现算法本身存在计算冗余，比如均值模糊的滑窗与拉普拉斯的滑窗
- **读写冗余**：或者内存访问不连续导致不能充分利用硬件缓存，产生多余的内存传输。

模型压缩

低比特量化、剪枝、蒸馏等

Optimizer Challenge 优化模块挑战

- **结构冗余**：深度学习网络模型结构中的无效计算节点、重复的计算子图、相同的结构模块，可以在保留相同计算图语义情况下无损去除的冗余类型；
- **精度冗余**：推理引擎数据单元是张量，一般为FP32浮点数，FP32表示的特征范围在某些场景存在冗余，可压缩到 FP16/INT8 甚至更低；数据中可能存大量0或者重复数据。
- **算法冗余**：算子或者Kernel层面的实现算法本身存在计算冗余，比如均值模糊的滑窗与拉普拉斯的滑窗实现方式相同。
- **读写冗余**：在一些计算场景重复读写内存，或者内存[illegible]缓存，产生多余的内存传输。

统一算子/计算图表达

Kernel提升泛化性

Optimizer Challenge 优化模块挑战

- **结构冗余**：深度学习网络模型结构中的无效计算节点、重复的计算子图、相同的结构模块，可以在保留相同计算图语义情况下无损去除的冗余类型；
- **精度冗余**：推理引擎数据单元是张量，一般为FP32浮点数，FP32表示的特征范围在某些场景存在冗余，可压缩到 FP16/INT8 甚至更低；数据中可能存大量0或者重复数据。
- **算法冗余**：算子或者Kernel层面的实现算法本身存在计算冗余，比如均值模糊的滑窗与拉普拉斯的滑窗实现方式相同。
- **读写冗余**：在一些计算场景重复读写内存，或者内存访问不连续导致不能充分利用硬件缓存，产生多余的内存传输。

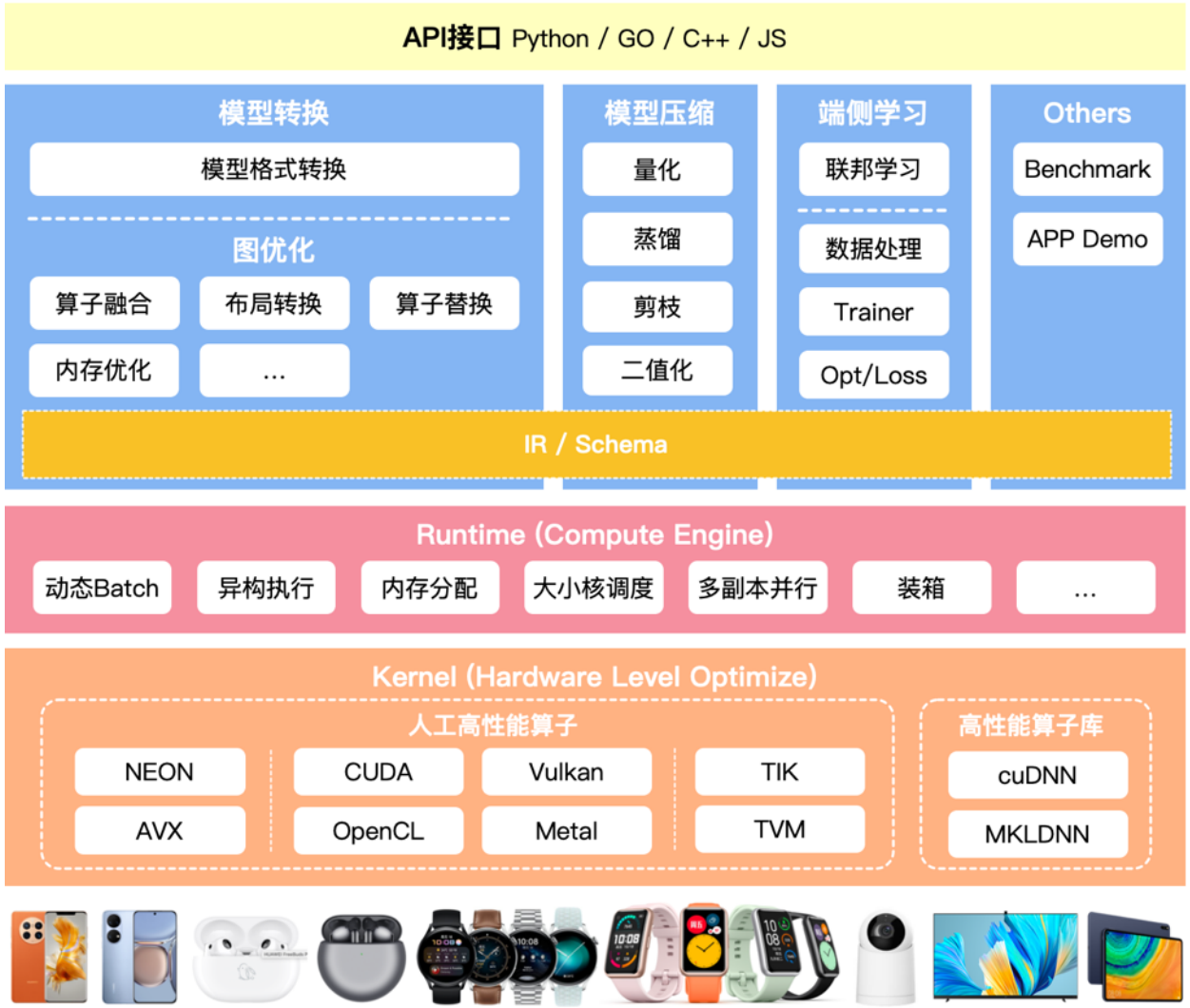
数据排布优化

内存分配优化

转换与优化

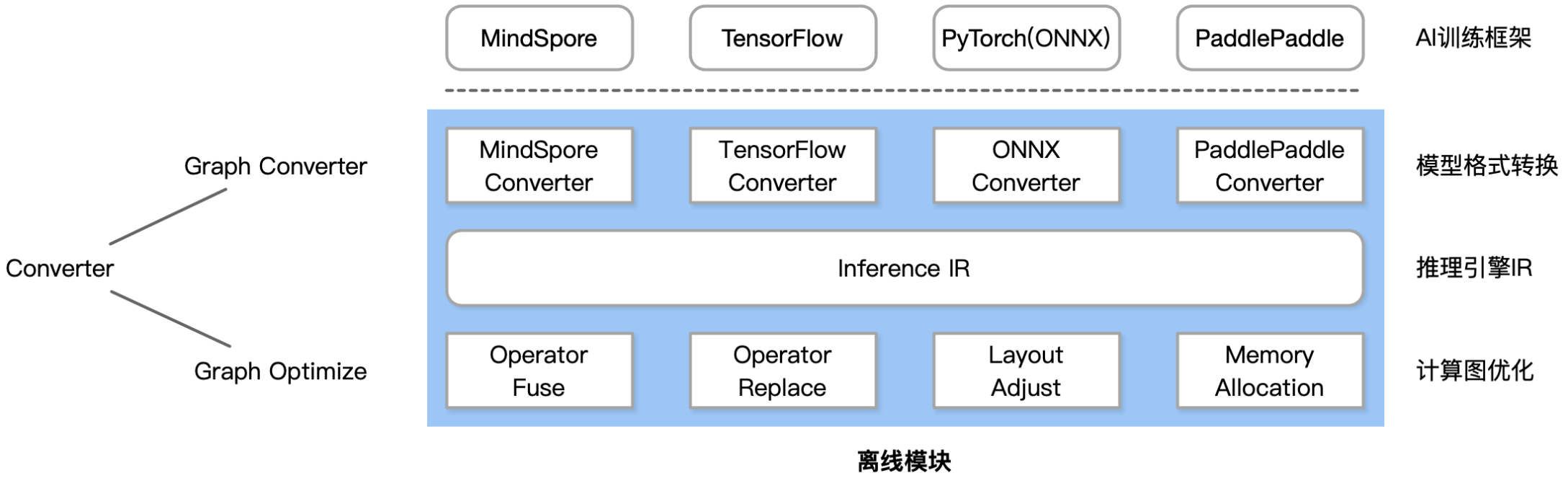
模块架构与流程

推理引擎架构



转换模块架构

- Converter由Frontends和Graph Optimize构成。前者负责支持不同的AI 训练框架；后者通过算子融合、算子替代、布局调整等方式优化计算图：



转换模块的工作流程