亚马逊云科技
中国峰会

AIM-XXX

打通云训练到端侧 NPU 推理的交付闭环

从 SageMaker 训练到端侧设备的全自动交付链路

王维超

解决方案架构师
亚马逊云科技

议程

- 为什么端侧 AI 已成刚需
- 训练与部署之间的工程缝隙
- 云端协同链路：从训练到端侧的全景
- 实战演示：从 SageMaker 训练到端侧 NPU 部署
- Pipeline 自动判定上线，零人工干预
- 工程实践要点

亚马逊云科技
中国峰会

01 端侧 AI 推理：从可选项到必选项

三个场景，一个共同诉求

工业质检

- 产线节拍 4–6 秒一件
- PCB 检测 30 fps
- 网络可用性差，数据不能出厂
- 宽温区间长期稳定

汽车与车载

- 隧道 / 地下车库断网仍可用
- ADAS 视觉感知毫秒级响应
- 车规芯片功耗与热预算更紧

消费电子

- 30 fps 出图、单帧 33 ms 预算
- AR 目标识别 +6DoF 跟踪
- 图像不上传云端

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

亚马逊云科技
中国峰会

“图像和语音不出场、推理延迟可控、断网时业务不停——把模型跑在端侧 NPU 上是这些场景共同的选择。”

共性诉求
—— 端侧 AI 的三条硬约束

但落地很难：开发者面对三组工程难题

模型定制

开源模型很少能直接命中类目分布、相机色彩、光照条件，必须用业务数据做迁移训练。

针对特定芯片的优化

不同 NPU 架构、Runtime（TFLite / ONNX Runtime）、量化策略各不相同，PyTorch 到目标二进制涉及大量手工劳动。

真机验证

性能数字必须来自目标设备本身，跨平台延迟差异可能成倍出现，缺少真机数据评审就靠拍脑袋。

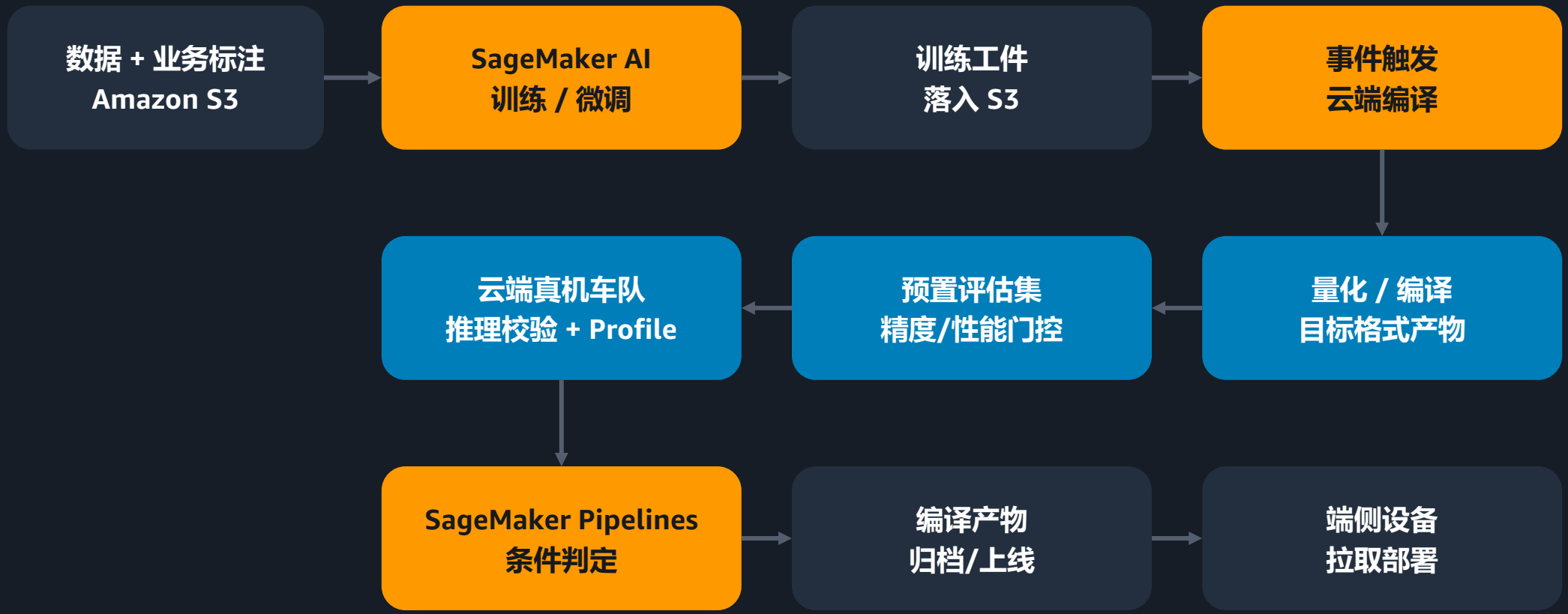
开发者真正想要的：端到端、可复制、可自动化

- 训练在云端
- 优化与编译在云端
- 真机验证在云端
- 只把编译产物下发到端侧

亚马逊云科技
中国峰会

02 协同链路：填补训练与端侧之间的工程缝隙

端到端协同链路总览



数据始终在 Amazon 生态内流转 · 全程零人工干预

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

SageMaker AI：训练与产物管理

弹性训练算力

按需获取 NVIDIA A10G、H100 等 GPU；按使用量计费，免运维训练集群。

BYOM / BYOD

自带模型与自带数据微调；分布式训练扩展到上千 GPU，集成 DeepSpeed / FSDP。

工件管理

训练完成后产物自动落入 S3 并释放算力；可注册到 Model Registry 进入生命周期管理。

端侧编译与真机验证服务：填补落地最后一公里

云端编译

PyTorch / ONNX 自动转换 → 图优化 → 量化 → 编译为目标 Runtime 产物（ TFLite 等）。

云端真机车队

真实手机、车载域控、工业模组组成的设备车队；同代 NPU 也能拿到逐设备性能数据。

API 化能力

submit_compile_job / submit_profile_job / submit_inference_job

三个 API 打通整条链路。

亚马逊云科技
中国峰会

03 闭环演示：从 SageMaker 训练到端侧部署

案例：手机端实时人像分割

场景与约束

- 视频会议背景虚化
- 30 fps 出图 · 单帧 < 33 ms
- 图像不上传云端
- 走端侧 NPU 推理

技术选型

- 网络：FPN + MobileNetV2（ NPU 友好）
- 数据：Hugging Face 公开人体解析数据集
- 训练：SageMaker AI on ml.g5.2xlarge
- 部署：INT8 量化 TFLite，端侧 NPU

STEP 1 · 在 SAGEMAKER AI 上启动训练

PyTorch estimator：一颗 A10G / 10 epoch / 260 秒

```
from sagemaker.pytorch import PyTorch
import sagemaker

estimator = PyTorch(
    entry_point='train_portrait_seg.py',
    source_dir='scripts',
    role=role,
    instance_count=1,
    instance_type='ml.g5.2xlarge',
    framework_version='2.1.0',
    py_version='py310',
    hyperparameters={'epochs': 10, 'lr': 1e-3},
    output_path=f's3://{bucket}/portrait-seg/',
)

estimator.fit({'train': train_s3, 'val': val_s3})
# best.pt + portrait_seg.onnx → model.tar.gz 落入 S3
```

STEP 2 · 训练工件落 S3 → 触发云端编译

提交编译 Job，目标 Runtime 为 TFLite + INT8

```
# 训练产出的ONNX 模型 → 提交到云端编译服务
import torch, edge_compiler as ec

traced = torch.jit.trace(model.eval(),
                         torch.randn(1, 3, 256, 256))

compile_job = ec.submit_compile_job(
    model=traced,
    input_specs={'image': ((1, 3, 256, 256), 'float32')},
    device=ec.Device('Galaxy S24'),
    options=('--target_runtime tflite '
             '--quantize_full_type int8 '
             '--quantize_io'),
    name='portrait-seg-int8',
)
compiled_model = compile_job.get_target_model()
# FP32 ONNX 16.5 MB → INT8 TFLite 4.5 MB  (≈27%)
```

STEP 3 · 预置评估集在量化阶段拦截不达标版本

云端真机推理 + 自动精度门控，IoU 不达标即终止

```
# 在云端真实 NPU 设备上跑推理，与 FP32 逐像素对比
inf_job = ec.submit_inference_job(
    model=compiled_model,
    inputs={'image': eval_set_inputs},  # 预置评估集
    device=ec.Device('Galaxy S24'),
)
npu_out = inf_job.download_output_data()

iou_npu  = compute_iou(npu_out, gt_masks)
iou_fp32 = compute_iou(fp32_ref, gt_masks)

GATE_IOU_DROP = 0.01    # 与 FP32 相比的最大允许下降
GATE_LATENCY  = 20.0    # 单帧最大延迟（ms）

passed = ((iou_fp32 - iou_npu) <= GATE_IOU_DROP
          and profile_job.latency_ms <= GATE_LATENCY)
# passed=False → Pipeline 终止，模型不进入注册表
```

真机结果：在 Galaxy S24 NPU 上的实测数据

13.59 ms

INT8 模型单帧推理延迟，远小于 30 fps 所需的 33 ms 预算（含渲染与后处理）。

4.5 MB

INT8 量化把模型体积从 FP32 ONNX 的 16.5 MB 压到 4.5 MB，约为原始的 27%。

IoU 0.935

端侧 NPU 推理 vs FP32 本地推理几乎一致（FP32 为 0.933），二值掩膜质量满足生产要求。

端到端验证：FP32 vs 端侧 NPU 99.6% 像素一致

- 在同一张验证集图像上分别跑本地 FP32 推理与云端真机 NPU 推理
- FP32 IoU = 0.933，NPU IoU = 0.935，差异落在人像边缘的 64 个像素
- 差异热力图显示：除边缘锯齿外，主体区域完全一致
- 结论：从 SageMaker 训练 → 云端编译 → 端侧 NPU 运行，链路语义保真
- 整个验证过程不需要采购物理设备，多机型只需更换 device 参数

STEP 4 · 接入 PIPELINE 自动判定上线

SageMaker Pipelines + ConditionStep：零人工干预

```
from sagemaker.workflow.condition_step import ConditionStep
from sagemaker.workflow.conditions import ConditionGreaterThanOrEqualTo
from sagemaker.workflow.pipeline import Pipeline

# 训练 → 云端编译 → 云端真机评估
train_step   = TrainingStep('Train',   estimator=estimator, ...)
compile_step = ProcessingStep('Compile', processor=compiler, ...)
eval_step    = ProcessingStep('EvalOnDevice', processor=evaluator, ...)

gate = ConditionGreaterThanOrEqualTo(
   left=eval_step.properties.iou_npu, right=0.92)

register_step = ModelStep('Register',
                  step_args=model.register(...))

pipeline = Pipeline(
   name='edge-npu-delivery',
   steps=[train_step, compile_step, eval_step,
        ConditionStep('QualityGate', conditions=[gate],
                  if_steps=[register_step], else_steps=[])])
# IoU 达标 → 自动注册并上线；否则终止 Pipeline
```

工程实践要点

多设备真机数据

同代 NPU 在不同平台延迟也可能成倍差异，更换 device 参数即可获取多机型真机 Profile。

量化精度单独验证

PSNR 对输出范围有界的任务有效；分割 logit 需用 sigmoid+ 阈值后的 IoU/ 像素一致率作为门控。

公开 → 产品迁移

公开数据训练得到的 backbone 作为热启动，再用产品图像做二次微调以适配相机色彩与光照。

总结：端侧 AI 交付从月级到日级

- SageMaker AI 提供按需扩展的训练能力，工件自动落入 S3
- 云端编译与真机验证服务把跨芯片量化、编译、Profile 封装为 API
- 数据始终在 Amazon 生态内流转，无跨云搬运
- Pipeline 条件步骤实现精度门控，全程零人工干预
- 下一步：拉一份业务数据，把这条链路在你自己的 Notebook 里跑一遍
- 同样的模式可复用到 OCR、关键词唤醒、AR 目标检测、低光增强、端侧蒸馏

亚马逊云科技
中国峰会

Thank you

亚马逊云科技
中国峰会

Session 6:打通云训练到端侧NPU推理的交付闭环

扫描上方二维码
填写调查问卷

亚马逊云科技
中国峰会

Session 6:打通云训练到端侧NPU推理的交付闭环

扫描上方二维码
填写调查问卷

亚马逊云科技
中国峰会

Session 6:打通云训练到端侧NPU推理的交付闭环

扫描上方二维码
填写调查问卷

亚马逊云科技
中国峰会

Session 6:打通云训练到端侧NPU推理的交付闭环

扫描上方二维码
填写调查问卷